Symbolic Model Checking Quantum Circuits in Maude

Canh Minh Do and Kazuhiro Ogata
School of Information Science
Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
Email: {*canhdo,ogata*}*@jaist.ac.jp*

***Abstract*—This paper presents a symbolic approach to model checking quantum circuits by using a set of laws from quantum mechanics and basic matrix operations with Dirac notation. We use Maude, a high-level specification/programming language based on rewriting logic, to implement our symbolic approach. As a case study, we use the approach to formally specify and verify the correctness of the quantum teleportation protocol, which is an important quantum communication protocol in the early work of quantum communications. Moreover, our implementation can be used as a general framework to formally specify and verify quantum circuits in Maude in an effortless way, where only an initial quantum state and a sequence of actions describing how a quantum circuit works in a simple way are required.**

***Keywords*-quantum circuits; Dirac notation; symbolic model checking; Maude**

I. Introduction

Quantum circuits are a model of quantum computation used in quantum computing. They are composed of a sequence of quantum gates, measurements, initializations of qubits, and possibly other actions. Quantum gates operate on quantum bits (qubits), the quantum counterpart of classical bits, and manipulate the state of a quantum system to perform quantum computations. The outputs of quantum circuits are quantum states, which can be measured to obtain classical outcomes with probabilities from which other actions can take place. Quantum circuits play a crucial role in quantum algorithms as they are used to design and implement quantum algorithms before actually running on quantum computers. Because quantum computing is counter-intuitive and radically different from classical computing, the likelihood of errors in quantum algorithms and circuits is much higher than in classical algorithms. Therefore, it is critical to verify that quantum circuits (or algorithms) enjoy desired properties. There is a symbolic approach [1] to (semi-)automatically reasoning about quantum circuits in Coq[1], an interactive theorem prover, but it often requires human users to provide necessary lemmas to complete its proofs. Meanwhile, model checking is a formal verification technique widely used in both academia and industry to automatically verify that a system satisfies some desired properties. Although there are some model checkers dedicated to quantum programs [2], [3], there is still a gap between model checking quantum programs and quantum circuits, which should be filled in.

In this present paper, we propose a symbolic approach to model checking quantum circuits by using a set of laws from quantum mechanics and basic matrix operations with Dirac notation [4]. Concretely, quantum states, quantum gates, and measurements are described in Dirac notation instead of using explicitly complex vectors and matrices as Paykin et al. proposed in [5], making our representations more compact. Using the set of laws, we can automatically reason about quantum operations on quantum data, such as qubits. We use Maude [6], a high-level specification/programming language based on rewriting logic, to formalize quantum states, some basic gates (e.g., Hadamard and controlled-NOT gates), and measurements on a standard basis with Dirac notation. As a case study, we use our approach to analyze the quantum teleportation protocol, which is an important quantum communication protocol in the early work of quantum communications. Moreover, our formalization takes the probabilities into account and so we are able to analyze both the quantitative and qualitative properties of the quantum teleportation protocol with a built-in LTL model checker in Maude. Although we only use the quantum teleportation protocol as a case study, our implementation can be used as a general framework to formally specify and verify quantum circuits in an effortless way, where only an initial quantum state and a sequence of actions describing how a quantum circuit works in a simple way are required. Our implementation is publicly available at https://github.com/canhminhdo/QTC-Maude.

The rest of the paper is organized as follows: § II Preliminaries, § III Symbolic Reasoning, § IV Quantum Teleportation Protocol, § V Formal Specification, § VI Symbolic Model Checking, § VII Related Work, and § VIII Conclusion.

II. Preliminaries

This section briefly describes some basic notations from quantum mechanics based on linear algebra and Kripke structures.

This work was supported by JST SICORP Grant Number JPMJSC20C2, Japan and JSPS KAKENHI under Grant JP24H03370.
DOI reference number: 10.18293/SEKE2023-014

[1] https://coq.inria.fr/

A. Basic Quantum Mechanics

In classical computing, the fundamental unit of information is a bit whose value is either 0 or 1. In quantum computing, the counterpart is a *quantum bit* or *qubit*, which has two basis states, conventionally written in Dirac notation [4] as $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$, corresponding to one-bit classical values, whose values are two column vectors $\begin{pmatrix}1\\0\end{pmatrix}$ and $\begin{pmatrix}0\\1\end{pmatrix}$, respectively. In quantum theory, a general state of a quantum system is a superposition or linear combination of basis states. A single qubit has state $|\psi\rangle = \alpha\,|\mathbf{0}\rangle + \beta\,|\mathbf{1}\rangle$, where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. States can be represented by column complex vectors as follows: $|\psi\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix} = \alpha\,|\mathbf{0}\rangle + \beta\,|\mathbf{1}\rangle$, where $\{|\mathbf{0}\rangle, |\mathbf{1}\rangle\}$ forms an orthonormal basis of the 2D complex vector space. The basis $\{|\mathbf{0}\rangle, |\mathbf{1}\rangle\}$ is also called as the *standard* basis. Formally, a quantum state is a unit vector in a Hilbert space $\mathcal{H}$, which is equipped with an inner product satisfying some axioms.

The evolution of a closed quantum system can be performed by a unitary transformation. If the state of a qubit is represented by a column vector then a unitary transformation $\boldsymbol{U}$ can be represented by a complex-value matrix such that $\boldsymbol{U}\boldsymbol{U}^\dagger = \boldsymbol{U}^\dagger\boldsymbol{U} = \boldsymbol{I}$ or $\boldsymbol{U}^\dagger = \boldsymbol{U}^{-1}$, where $\boldsymbol{U}^\dagger$ is the conjugate transpose of $\boldsymbol{U}$. $\boldsymbol{U}$ acts on the Hilbert space $\mathcal{H}$ transforming a state $|\psi\rangle$ to a state $|\psi'\rangle$ by a matrix multiplication such that $|\psi'\rangle = \boldsymbol{U}\,|\psi\rangle$. There are some common quantum gates: the Hadamard gate $\boldsymbol{H}$, the identity gate $\boldsymbol{I}$, the Pauli gates $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$, and the controlled-NOT gate $\boldsymbol{CX}$. Note that the $\boldsymbol{CX}$ gate performs on two qubits, while the remaining gates perform on a single qubit. Their matrix representations are as follows:

$$\boldsymbol{I}_2 = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}, \qquad \boldsymbol{X} = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix},$$

$$\boldsymbol{Y} = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}, \qquad \boldsymbol{Z} = \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix},$$

$$\boldsymbol{H} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}, \quad \boldsymbol{CX} = \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}.$$

For example, the Hadamard gate on a single qubit performs the mapping $|\mathbf{0}\rangle \mapsto \frac{1}{\sqrt{2}}(|\mathbf{0}\rangle + |\mathbf{1}\rangle)$ and $|\mathbf{1}\rangle \mapsto \frac{1}{\sqrt{2}}(|\mathbf{0}\rangle - |\mathbf{1}\rangle)$. The controlled-NOT gate on pairs of qubits performs the mapping $|\mathbf{00}\rangle \mapsto |\mathbf{00}\rangle, |\mathbf{01}\rangle \mapsto |\mathbf{01}\rangle, |\mathbf{10}\rangle \mapsto |\mathbf{11}\rangle, |\mathbf{11}\rangle \mapsto |\mathbf{10}\rangle$, which can be understood as inverting the second qubit (referred to as the *target*) if and only if the first qubit (referred to as the *control*) is one. For the sake of simplicity, we do not take the Pauli gate $\boldsymbol{Y}$ into account in this present paper because it is not used in our case study.

A quantum measurement is described as a collection $\{\boldsymbol{M}_m\}$ of measurement operators, where the indices m refer to the measurement outcomes. It is required that the measurement operators satisfy $\sum_m \boldsymbol{M}_m^\dagger \boldsymbol{M}_m = \boldsymbol{I}_{\mathcal{H}}$. If the state of a quantum system is $|\psi\rangle$ before the measurement, then the probability for the result m is as follows:

$$p(m) = \langle\psi|\,\boldsymbol{M}_m^\dagger \boldsymbol{M}_m\,|\psi\rangle,$$

the state of the quantum system after the measurement is $\frac{\boldsymbol{M}_m|\psi\rangle}{\sqrt{p(m)}}$ provided that $p(m) > 0$. For example, if a qubit is in state $\alpha\,|\mathbf{0}\rangle + \beta\,|\mathbf{1}\rangle$ and measuring with $\{\boldsymbol{M}_0, \boldsymbol{M}_1\}$ operators, we have the result 0 with probability $|\alpha|^2$ at the post-measurement state $|\mathbf{0}\rangle$ and the result 1 with probability $|\beta|^2$ at the post-measurement state $|\mathbf{1}\rangle$, where $\boldsymbol{M}_0 = |\mathbf{0}\rangle \times \langle\mathbf{0}|$ and $\boldsymbol{M}_1 = |\mathbf{1}\rangle \times \langle\mathbf{1}|$.

For multiple qubits, we use the tensor product of Hilbert spaces. Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two Hilbert spaces. Their tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ is defined as a vector space consisting of linear combinations of the vectors $|\psi_1\psi_2\rangle = |\psi_1\rangle\,|\psi_2\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$, where $|\psi_1\rangle \in \mathcal{H}_1$ and $|\psi_2\rangle \in \mathcal{H}_2$. Systems of two or more qubits may be in *entangled* states, meaning that states of qubits are correlated and inseparable. For example, we consider a measurement of the first qubit of the entangled state $\frac{1}{\sqrt{2}}(|\mathbf{00}\rangle + |\mathbf{11}\rangle)$. The result is 0 with probability $\frac{1}{2}$ leaving its state $|\mathbf{00}\rangle$ or 1 with probability $\frac{1}{2}$ leaving its state $|\mathbf{11}\rangle$. In either case, a subsequent measurement of the second qubit gives a non-probabilistic result, which is immediate to the result of the first measurement before. Entanglement shows that an entangled state of two qubits cannot be expressed as a tensor product of single-qubit states. We can use $\boldsymbol{H}$ and $\boldsymbol{CX}$ gates to create entangled states as follows: $\boldsymbol{CX}((\boldsymbol{H} \otimes \boldsymbol{I})\,|\mathbf{00}\rangle) = \frac{1}{\sqrt{2}}(|\mathbf{00}\rangle + |\mathbf{11}\rangle)$.

B. Kripke Structures

A Kripke structure K is $\langle S, I, T, A, L\rangle$, where S is a set of states, $I \subseteq S$ is the set of initial states, $T \subseteq S \times S$ is a left-total binary relation over S, A is a set of atomic propositions and L is a labeling function whose type is $S \to 2^A$. Each element $(s, s') \in T$ is called a state transition from s to s' and T may be called the state transitions (with respect to K). For a state $s \in S$, $L(s)$ is the set of atomic propositions that hold in s. A path π is an infinite sequence $s_0, \ldots, s_i, s_{i+1}, \ldots$ such that $s_i \in S$ and $(s_i, s_{i+1}) \in T$ for each i. We use the following notation for paths: $\pi^i \triangleq s_i, s_{i+1}, \ldots$, where $\triangleq$ is used as "be defined as." π^i is obtained by deleting the first i states $s_0, s_1, \ldots, s_{i-1}$ from π. Let $\mathcal{P}$ be the set of all paths. π is called a computation if $\pi(0) \in I$. Let $\mathcal{C}$ be the set of all computations.

The syntax of a formula φ in LTL for K is as follows:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\,\varphi \mid \varphi\,\mathcal{U}\,\varphi$$

where $p \in A$. Let $\mathcal{F}$ be the set of all formulas in LTL for K. An arbitrary path $\pi \in \mathcal{P}$ of K and an arbitrary LTL formula $\varphi \in \mathcal{F}$ of K, $K, \pi \models \varphi$ is inductively defined as follows:

- $K, \pi \models \top$

- $K, \pi \models p$ iff $p \in L(\pi(0))$
- $K, \pi \models \neg\varphi_1$ iff $K, \pi \not\models \varphi_1$
- $K, \pi \models \varphi_1 \wedge \varphi_2$ iff $K, \pi \models \varphi_1$ and $K, \pi \models \varphi_2$
- $K, \pi \models \bigcirc \varphi_1$ iff $K, \pi^1 \models \varphi_1$
- $K, \pi \models \varphi_1 \,\mathcal{U}\, \varphi_2$ iff there exists a natural number i such that $K, \pi^i \models \varphi_2$ and for all natural numbers $j < i$, $K, \pi^j \models \varphi_1$

where φ_1 and φ_2 are LTL formulas. Then, $K \models \varphi$ iff $K, \pi \models \varphi$ for each computation $\pi \in \mathcal{C}$ of K. $\bigcirc$ and $\mathcal{U}$ are called the next temporal connective and the until temporal connective, respectively.

In this paper, a state is expressed as a braced associative-commutative (AC) collection of name-value pairs. The order of elements is not relevant in AC collections, such as sets. AC collections are called soups, and name-value pairs are called observable components. That is, a state is expressed as a braced soup of observable components. The juxtaposition operator is used as the constructor of soups. Let oc_1, oc_2, oc_3 be observable components, and then $oc_1\ oc_2\ oc_3$ is the soup of those three observable components. Since the order is irrelevant because of AC, $oc_1\ oc_2\ oc_3$ is the same as some others, such as $oc_3\ oc_2\ oc_1$. A state is expressed as $\{oc_1\ oc_2\ oc_3\}$. In this paper, rewrite rules are used to specify state transitions. Concretely, we use Maude [6], a programming/specification language based on rewriting logic. Maude makes it possible to specify complex systems flexibly and is also equipped with an LTL model checker.

III. Symbolic reasoning

This section introduces some terms used in our symbolic reasoning and a set of laws used to reduce terms.

A. Terms

Terms are built from scalars and basic vectors with some constructors.

- Scalars are complex numbers. We extend rational numbers supported in Maude to deal with complex numbers. Some constructors for scalars, such as multiplication, fraction, addition, conjugation, absolute, power, and square root are formalized, but we do not mention them here to make the paper concise.
- Basic vectors are the standard basis written in Dirac notation as $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$.
- Constructors for matrices consist of scalar multiplication of matrices $\cdot$, matrix product $\times$, matrix addition $+$, tensor product $\otimes$, and the conjugate transpose $\boldsymbol{A}^\dagger$ of a matrix $\boldsymbol{A}$.

In Dirac notation, $\langle\mathbf{0}|$ is the dual of $|\mathbf{0}\rangle$ such that $\langle\mathbf{0}|^\dagger = |\mathbf{0}\rangle$ and $|\mathbf{0}\rangle^\dagger = \langle\mathbf{0}|$; similarly for $\langle\mathbf{1}|$. The terms $|j\rangle \times \langle k|$ and $\langle j| \times |k\rangle$ may be written shortly as $|j\rangle\langle k|$ and $\langle j|k\rangle$ for any $j, k \in \{0, 1\}$. By using these notations, we can intuitively explain how quantum operations work. For example, the $\boldsymbol{X}$ gate performs mapping $|\mathbf{0}\rangle \mapsto |\mathbf{1}\rangle$ and $|\mathbf{1}\rangle \mapsto |\mathbf{0}\rangle$. Therefore, we formalize the $\boldsymbol{X}$ gate as $|\mathbf{0}\rangle\langle\mathbf{1}| + |\mathbf{1}\rangle\langle\mathbf{0}|$ in Maude instead of using explicitly the matrix representation $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. It is immediate that $\boldsymbol{X}|\mathbf{0}\rangle = |\mathbf{1}\rangle\langle\mathbf{0}|\mathbf{0}\rangle + |\mathbf{0}\rangle\langle\mathbf{1}|\mathbf{0}\rangle = |\mathbf{1}\rangle$ because of the use of some laws in Table I and similarly for $\boldsymbol{X}|\mathbf{1}\rangle$.

We conventionally formalize some basic matrices $\boldsymbol{B}_i$ for $i \in [0..3]$ as follows:

$$\begin{array}{ll} \boldsymbol{B}_0 = |\mathbf{0}\rangle \times \langle\mathbf{0}|, & \boldsymbol{B}_1 = |\mathbf{0}\rangle \times \langle\mathbf{1}|, \\ \boldsymbol{B}_2 = |\mathbf{1}\rangle \times \langle\mathbf{0}|, & \boldsymbol{B}_3 = |\mathbf{1}\rangle \times \langle\mathbf{1}|. \end{array}$$

The $\boldsymbol{CX}$, $\boldsymbol{X}$, $\boldsymbol{Z}$, and $\boldsymbol{H}$ gates are then a linear combination of the matrices $\boldsymbol{B}_i$ as follows:

$$\boldsymbol{CX} = \boldsymbol{B}_0 \otimes \boldsymbol{I}_2 + \boldsymbol{B}_3 \otimes \boldsymbol{X},$$
$$\boldsymbol{X} = \boldsymbol{B}_1 + \boldsymbol{B}_2,\ \boldsymbol{Z} = \boldsymbol{B}_1 + (-1) \cdot \boldsymbol{B}_3,$$
$$\boldsymbol{H} = \tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_0 + \tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_1 + \tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_2 + (-\tfrac{1}{\sqrt{2}}) \cdot \boldsymbol{B}_3.$$

B. Laws

We use a set of laws in Table I derived from the properties of quantum mechanics and basic matrix operations and thus they are immediately sound (see their proofs in Coq in [1]). Because $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$ can be viewed as 2×1 matrices, then the laws actually describe matrix calculations with Dirac notation, zero and identity matrices, and scalars. These laws are described by equations in Maude and are used to automatically reduce terms until no more matrix operation is applicable. Some laws dedicated to simplifying the expressions about complex numbers are also formalized in Maude by means of equations, but we do not describe them here to make the paper concise.

For example, we would like to reduce the term $\boldsymbol{CX} \times ((\boldsymbol{H} \otimes \boldsymbol{I}) \times |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle)$ to check whether its result is $\frac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \frac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle \otimes |\mathbf{1}\rangle$. The term says that the $\boldsymbol{H}$ gate acts on the first qubit followed by the $\boldsymbol{CX}$ gate where the control and target bits are the first and second qubits, respectively. The simplification of the term goes as follows:

$$\begin{aligned}
&\boldsymbol{H} \times |\mathbf{0}\rangle \\
&= (\tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_0 + \tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_1 + \tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_2 + (-\tfrac{1}{\sqrt{2}}) \cdot \boldsymbol{B}_3) \times |\mathbf{0}\rangle \\
&= \tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_0 \times |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_1 \times |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot \boldsymbol{B}_2 \times |\mathbf{0}\rangle + (-\tfrac{1}{\sqrt{2}}) \cdot \boldsymbol{B}_3 \times |\mathbf{0}\rangle \\
&= \tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle \times \langle\mathbf{0}| \times |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle \times \langle\mathbf{1}| \times |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle \times \langle\mathbf{0}| \times \\
&|\mathbf{0}\rangle + (-\tfrac{1}{\sqrt{2}}) \cdot |\mathbf{1}\rangle \times \langle\mathbf{1}| \times |\mathbf{0}\rangle \\
&= \tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle
\end{aligned}$$

$$\begin{aligned}
&(\boldsymbol{H} \otimes \boldsymbol{I}) \times (|\mathbf{0}\rangle \otimes |\mathbf{0}\rangle) \\
&= (\boldsymbol{H} \times |\mathbf{0}\rangle) \otimes (\boldsymbol{I} \times |\mathbf{0}\rangle) \\
&= (\tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle) \otimes |\mathbf{0}\rangle \\
&= \tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle \otimes |\mathbf{0}\rangle
\end{aligned}$$

$$\begin{aligned}
&\boldsymbol{CX} \times ((\boldsymbol{H} \otimes \boldsymbol{I}) \times (|\mathbf{0}\rangle \otimes |\mathbf{0}\rangle)) \\
&= (\boldsymbol{B}_0 \otimes \boldsymbol{I} + \boldsymbol{B}_3 \otimes \boldsymbol{X}) \times (\tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle \otimes |\mathbf{0}\rangle) \\
&= (\boldsymbol{B}_0 \otimes \boldsymbol{I}) \times (\tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle) + (\boldsymbol{B}_0 \otimes \boldsymbol{I}) \times (\tfrac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle \otimes |\mathbf{0}\rangle) + \\
&(\boldsymbol{B}_3 \otimes \boldsymbol{X}) \times (\tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle) + (\boldsymbol{B}_3 \otimes \boldsymbol{X}) \times (\tfrac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle \otimes |\mathbf{0}\rangle) \\
&= \tfrac{1}{\sqrt{2}} \cdot (\boldsymbol{B}_0 \times |\mathbf{0}\rangle) \otimes (\boldsymbol{I} \times |\mathbf{0}\rangle) + \tfrac{1}{\sqrt{2}} \cdot (\boldsymbol{B}_0 \times |\mathbf{1}\rangle) \otimes (\boldsymbol{I} \times |\mathbf{0}\rangle) + \\
&\tfrac{1}{\sqrt{2}} \cdot (\boldsymbol{B}_3 \times |\mathbf{0}\rangle) \otimes (\boldsymbol{X} \times |\mathbf{0}\rangle) + \tfrac{1}{\sqrt{2}} \cdot (\boldsymbol{B}_3 \times |\mathbf{1}\rangle) \otimes (\boldsymbol{X} \times |\mathbf{0}\rangle) \\
&= \tfrac{1}{\sqrt{2}} \cdot |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \tfrac{1}{\sqrt{2}} \cdot |\mathbf{1}\rangle \otimes |\mathbf{1}\rangle
\end{aligned}$$

Using the laws, the term is reduced to a normal form that is a linear combination of the tensor product of the standard

Table I
A SET OF LAWS USED FOR SYMBOLIC REASONING

No.	Law
L1	$\langle \mathbf{0} \vert \mathbf{0} \rangle = \langle \mathbf{1} \vert \mathbf{1} \rangle = 1, \langle \mathbf{1} \vert \mathbf{1} \rangle = \langle \mathbf{0} \vert \mathbf{1} \rangle = 0$
L2	Associativity of $\times, +, \otimes$ and Commutativity of $+$
L3	$0 \cdot \boldsymbol{A}_{m\times n} = \mathbf{O}_{m\times n},\ c \cdot \mathbf{O} = \mathbf{O},\ 1 \cdot \boldsymbol{A} = \boldsymbol{A}$
L4	$c \cdot (\boldsymbol{A} + \boldsymbol{B}) = c \cdot \boldsymbol{A} + c \cdot \boldsymbol{B}$
L5	$c_1 \cdot \boldsymbol{A} + c_2 \cdot \boldsymbol{A} = (c_1 + c_2) \cdot \boldsymbol{A}$
L6	$c_1 \cdot (c_2 \cdot \boldsymbol{A}) = (c_1 \cdot c_2) \cdot \boldsymbol{A}$
L7	$(c_1 \cdot \boldsymbol{A}) \times (c_2 \cdot \boldsymbol{B}) = (c_1 \cdot c_2) \cdot (\boldsymbol{A} \times \boldsymbol{B})$
L8	$\boldsymbol{A} \times (c \cdot \boldsymbol{B}) = (c \cdot \boldsymbol{A}) \times \boldsymbol{B} = c \cdot (\boldsymbol{A} \times \boldsymbol{B})$
L9	$\boldsymbol{A} \otimes (c \cdot \boldsymbol{B}) = (c \cdot \boldsymbol{A}) \otimes \boldsymbol{B} = c \cdot (\boldsymbol{A} \otimes \boldsymbol{B})$
L10	$\mathbf{O}_{m\times n} \times \boldsymbol{A}_{n\times p} = \boldsymbol{A}_{m\times n} \times \mathbf{O}_{n\times p} = \mathbf{O}_{m\times p}$
L11	$\boldsymbol{I}_m \times \boldsymbol{A}_{m\times n} = \boldsymbol{A}_{m\times n} \times \boldsymbol{I}_n = \boldsymbol{A}_{m\times n}$
L12	$\boldsymbol{A} + \mathbf{O} = \mathbf{O} + \boldsymbol{A} = \mathbf{O}$
L13	$\mathbf{O}_{m\times n} \otimes \boldsymbol{A}_{p\times q} = \boldsymbol{A}_{p\times q} \otimes \mathbf{O}_{m\times n} = \mathbf{O}_{mp\times nq}$
L14	$\boldsymbol{A} \times (\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{A} \times \boldsymbol{B} + \boldsymbol{A} \times \boldsymbol{C}$
L15	$(\boldsymbol{A} + \boldsymbol{B}) \times \boldsymbol{C} = \boldsymbol{A} \times \boldsymbol{C} + \boldsymbol{B} \times \boldsymbol{C}$
L16	$(\boldsymbol{A} \otimes \boldsymbol{B}) \times (\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{A} \times \boldsymbol{C}) \otimes (\boldsymbol{B} \times \boldsymbol{D})$
L17	$\boldsymbol{A} \otimes (\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{A} \otimes \boldsymbol{B} + \boldsymbol{A} \otimes \boldsymbol{C}$
L18	$(\boldsymbol{A} + \boldsymbol{B}) \otimes \boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{C} + \boldsymbol{B} \otimes \boldsymbol{C}$
L19	$(c \cdot \boldsymbol{A})^\dagger = c^* \cdot \boldsymbol{A}^\dagger,\ (\boldsymbol{A} \times \boldsymbol{B})^\dagger = \boldsymbol{B}^\dagger \times \boldsymbol{A}^\dagger$
L20	$(\boldsymbol{A} + \boldsymbol{B})^\dagger = \boldsymbol{A}^\dagger + \boldsymbol{B}^\dagger,\ (\boldsymbol{A} \otimes \boldsymbol{B})^\dagger = \boldsymbol{A}^\dagger \otimes \boldsymbol{B}^\dagger$
L21	$\boldsymbol{I}_m{}^\dagger = \boldsymbol{I}_m, \boldsymbol{O}^\dagger_{m\times n} = \boldsymbol{O}_{n\times m}, (\boldsymbol{A}^\dagger)^\dagger = \boldsymbol{A}$
L22	$\vert\mathbf{0}\rangle^\dagger = \langle\mathbf{0}\vert,\ \langle\mathbf{0}\vert^\dagger = \vert\mathbf{0}\rangle,\ \vert\mathbf{1}\rangle^\dagger = \langle\mathbf{1}\vert,\ \langle\mathbf{1}\vert^\dagger = \vert\mathbf{1}\rangle$

basis with scalars. The whole process is conducted automatically in Maude and the result is the same as expected. The key idea is to reduce the matrix multiplication in the form of $\langle i|j\rangle$ into a scalar and simplify the matrix representation by absorbing ones and eliminating zeros (see the law with label L3). In this manner, our symbolic reasoning about matrices can be conducted automatically by rewriting in Maude instead of explicitly calculating matrices.

IV. Quantum Teleportation Protocol

We use quantum teleportation protocol [7] as a case study to demonstrate how our symbolic reasoning can be used to model check quantum circuits in Maude. The protocol takes advantage of entanglement in quantum mechanics to send an unknown quantum state $|\psi\rangle$ from a sender to a receiver by using only three qubits and two classical bits. The circuit depicted in Fig. 1 shows how the protocol works. The single wires denote qubits referred to as q_i, while the double wires denote classical bits referred to as c_i. The sender acts on q_0 and q_1, and the receiver acts on q_2 as follows:

- First, we prepare an unknown state $|\psi\rangle = \alpha\,|\mathbf{0}\rangle + \beta\,|\mathbf{1}\rangle$ at q_0, where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. Initially, q_1 and q_2 are in the state $|\mathbf{0}\rangle$.
- Second, we apply a sequence of quantum gates to manipulate three qubits. We first apply the $\boldsymbol{H}$ gate on q_1 followed by the $\boldsymbol{CX}$ gate on q_1 and q_2 in order to make an entangled state shared between the sender and the receiver. The sender then applies the $\boldsymbol{CX}$ gate on q_0 and q_1 followed by the $\boldsymbol{H}$ gate on q_0.
- Third, we measure the qubits q_0 and q_1 and immediately obtain two classical outcomes (0 or 1) stored in c_0 and c_1, respectively.
- Fourth, we conditionally apply single-qubit $\boldsymbol{Z}$ and $\boldsymbol{X}$ gates on q_2 depending on the two classical bits in c_0 and c_1. Concretely, we use the $\boldsymbol{X}$ gate if c_1 equals 1 and follow by the $\boldsymbol{Z}$ gate if c_0 equals 1.

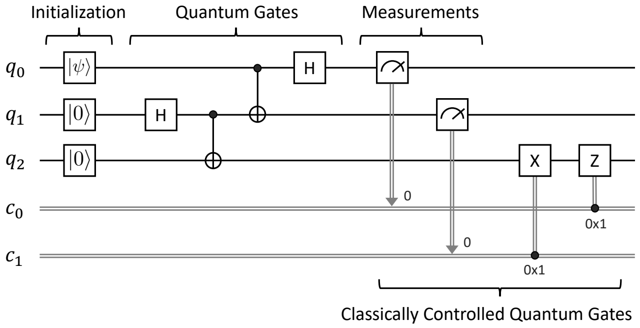

Figure 1. Quantum teleportation protocol

At the end, the receiver will have $|\psi\rangle$ and the sender will not have anymore. We would verify whether the sender can send correctly an arbitrary unknown quantum state to the receiver at the end by using our symbolic model checking.

V. Formal specification

A. Formalization of Qubits, Gates, and Measurements

Qubits are formalized as the linear combination of tensor product of the standard basis in Dirac notation with scalars and similarly for quantum gates. Because $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$ can be viewed as 2×1 matrices, then qubits and quantum gates are basically matrices. Quantum gates act on qubits (a quantum state) formalized as a matrix multiplication with a deterministic transition in Maude. In this paper, we only consider projective measurements on the standard basis, and thus the measurement operators are $\{\boldsymbol{M}_0, \boldsymbol{M}_1\}$. A measurement of a single qubit in a quantum state is formalized by two state transitions with probabilities $p(m)$ for $m \in \{0, 1\}$, making a non-deterministic probabilistic transition. Each of the two transitions shows how its measurement operator acts on the single qubit in a state and is formalized similarly as quantum gates, however, with respect to the probabilities.

B. A Generic Formalization of Quantum Circuits

A whole quantum state is formalized as a collection of qubits associated with indices in circuits, where each element is one of the forms as follows:

- `(q[`i`] = `$|\psi\rangle$`)` denote a single qubit in state $|\psi\rangle$ at q_i,
- `(q[`$i, \ldots, j$`] = `$|\psi\rangle$`)` denote an entangled state in state $|\psi\rangle$ at $q_i, \ldots, q_j$ where the order of $i, \ldots, j$ is relevant.

Classical bits are formalized as a map from indices in circuits to Boolean values, where each entry is in the form of $(i \mapsto b)$, meaning that the value of the classical bit stored at c_i is b whose value is either 0 or 1.

A sequence of quantum gates, measurements, and conditional gates in a quantum circuit is formalized as a list of actions in which each action is one of the forms as follows:

- I(i) applies the $\boldsymbol{I}$ gate on q_i,
- X(i) applies the $\boldsymbol{X}$ gate on q_i,
- Z(i) applies the $\boldsymbol{Z}$ gate on q_i,
- H(i) applies the $\boldsymbol{H}$ gate on q_i,
- CX(i,j) applies the $\boldsymbol{CX}$ gate on q_i and q_j,
- M(i) measures q_i with the standard basis,
- c[i] == b ? AL checks if the classical bit at c_i equals b, then a list AL of actions is executed.

Based on the actions formalized above, we can describe the circuit for quantum teleportation protocol as follows:

```
H(1) CX(1, 2) CX(0, 1) H(0) M(0) M(1)
(c[1] == 1 ? X(2)) (c[0] == 1 ? Z(2))
```

Let K_C be the Kripke structure formalizing a quantum circuit. There are five kinds of observable components in our formalization as follows:

- (qstate: qs) denotes the whole quantum state qs.
- (bits: bm) denotes the classical bits obtained from measurements and stored in a bit map bm.
- (prob: p) denotes the probability p at the current quantum state.
- (actions: al) denotes the action list al, guiding us on how the circuit works.
- (isEnd: b) denotes termination with Boolean flag b.

Each state in S_C is expressed as $\{obs\}$, where obs is a soup of one qstate observable component, one prob observable component, one bits observable component, one actions observable component, and one isEnd observable component.

T_C consists of 10 rewrite rules in our formalization. Let OCs be a Maude variable of observable component soups, Q and Q′ be Maude variables of whole quantum states, BM be a Maude variable of bit maps, Prob and Prob′ be Maude variables of scalars, AL and AL′ be Maude variables of action lists, B be a Maude variable of Boolean values, and N, N1, and N2 are Maude variables of natural numbers.

The first five rewrite rules are as follows:

```
rl [I] : {(qstate: Q) (actions: (I(N) AL))
OCs} => {(qstate: Q) (actions: AL) OCs} .
crl [X] : {(qstate: Q) (actions: (X(N) AL))
OCs} => {(qstate: Q′) (actions: AL) OCs}
if Q′ := (Q).X(N) .
crl [Z] : {(qstate: Q) (actions: (Z(N) AL))
OCs} => {(qstate: Q′) (actions: AL) OCs}
if Q′ := (Q).Z(N) .
crl [H] : {(qstate: Q) (actions: (H(N) AL))
OCs} => {(qstate: Q′) (actions: AL) OCs}
if Q′ := (Q).H(N) .
crl [CX] : {(qstate: Q) (actions: (CX(N1, N2)
AL)) OCs} => {(qstate: Q′) (actions: AL) OCs}
if Q′ := (Q).CX(N1, N2).
```

The rules I, X, Z, H, and CX simulate how the $\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{H}$, and $\boldsymbol{CX}$ gates act on the whole quantum state in qstate observable component if its action appears in actions observable component, respectively.

The next two rewrite rules are as follows:

```
crl [M0] : {(qstate: Q) (actions: (M(N) AL))
    (prob: Prob) (bits: BM) OCs}
=> {(qstate: Q′) (actions: AL) (prob: (Prob
    .* Prob′)) (bits: insert(N, 0, BM)) OCs}
if {qstate: Q′, prob: Prob′} := (Q).M(P0, N).
crl [M1] : {(qstate: Q) (actions: (M(N) AL))
    (prob: Prob) (bits: BM) OCs}
=> {(qstate: Q′) (actions: AL) (prob: (Prob
    .* Prob′)) (bits: insert(N, 1, BM)) OCs}
if {qstate: Q′, prob: Prob′} := (Q).M(P1, N).
```

The rules M0 and M1 say that we measure the qubit at index N with the measurement operators $\boldsymbol{M}_0$ and $\boldsymbol{M}_1$, respectively; the classical outcomes are stored accordingly into the bit map in bits observable component; the probabilities and the post-measurement states are also updated in prob and qstate observable components, respectively. These two rules make a non-deterministic probabilistic transition when measuring a single qubit.

The next rewrite rule describes how to conditionally perform the next actions based on classical bits obtained from measurements if applicable.

```
rl [cif] : {(qstate: Q) (bits: ((N |-> N1),
BM)) (actions: ((c[N] == N2 ? AL′) AL)) OCs}
=> {(qstate: Q) (bits: ((N |-> N1), BM))
(actions: ((if (N1 == N2) then AL′ else nil
    fi) AL)) OCs} .
```

This rule says that if c[N] == N2 ? AL′ is in the action list and the classical bit N1 at index N equals the conditional value N2, then the action list AL′ is prepended to the action list AL in actions observable component to be executed next; otherwise, it is ignored.

The last two rules are as follows:

```
rl [end]: {(actions: nil) (isEnd: false) OCs}
=> {(actions: nil) (isEnd: true) OCs} .
rl [stutter]: {(isEnd: true) OCs}
=> {(isEnd: true) OCs} .
```

The rule end marks the termination if the action list is nil, meaning no more action. Meanwhile, the rule stutter is necessary to make T_C total when isEnd observable component is true.

C. Formalization of Quantum Teleportation Protocol

To formalize quantum teleportation protocol, let I_C consist of only one initial state as follows:

```
{(isEnd: false) (prob: 1) (bits: empty)
(qstate: (q[0]: a . |0> + b . |1>)
         (q[1]: |0>) (q[2]: |0>))
(actions: H(1) CX(1,2) CX(0,1) H(0) M(0) M(1)
      (c[1] == 1 ? X(2)) (c[0] == 1 ? Z(2)))}
```

where a and b are Maude constants of scalars denoting arbitrary scalars such that $|\text{a}|^2 + |\text{b}|^2 = 1$. Initially, isEnd observable component is false, prob observable component is one, qstate is a symbolic state as the input state of the protocol, actions observable component contains the action list describing how the protocol works. For other protocols, we only need to formalize the initial quantum state and the action list in the initial state of I_C, while we can definitely reuse S_C and T_C in K_C, making our formalization as a general framework to formally specify quantum circuits.

VI. Symbolic Model Checking

Let `TELEPORT` be the specification of the quantum teleportation protocol, `init` be the initial state for `TELEPORT`, `qstate` and `qubitAt` be functions to get the whole quantum state from the initial state and to get a single qubit at some index, respectively. To model check that K_C satisfies some desired properties, we specify A_C and L_C. A_C has one atomic proposition `isSuccess`. L_C is specified as follows:

```
eq {(isEnd: true) (qstate: Q) (prob: Prob)
   OCs} |= isSuccess
= Prob > 0 implies
  qubitAt(Q, 2) == qubitAt(qstate(init), 0) .
eq {OCs} |= PROP = false [owise] .
```

The two equations say that `isSuccess` holds at a state if the state contains `(isEnd: true)`, `(qstate: Q)`, and`(prob: Prob)` such that the condition `qubitAt( Q, 2)== qubitAt(qstate(init), 0)` holds whenever `Prob > 0`, meaning that the qubit received by the receiver at the end is equal to the qubit sent by the sender at the beginning with a non-zero probability. Let a LTL formula `teleProp` be defined as `True U isSuccess`, where `U` is the temporal until operator.

We model check that K_C satisfies `teleProp` from the initial state `init` in Maude as follows:

```
red modelCheck(init, teleProp) .
```

No counterexample is found in just a few moments; thus, K_C satisfies `teleProp`. In other words, we successfully verify the correctness of the quantum teleportation protocol by using our symbolic model checking approach. Because our formalization considers the probabilities at each state, then we are able to check not only qualitative properties but also quantitative properties with Maude LTL model checker.

VII. Related work

There are several studies [8], [9] in the early work of formal specification and verification of quantum protocols. For example, Gay, et al. provide a way to use classical model checkers (e.g., PRISM - a probabilistic model checker) to analyze quantum protocols. They give each quantum state a unique number and the transition from a unique number to another unique number models the action of quantum gates and measurements. Their approach needs to enumerate states and calculate the state transitions in advance and then encode them into a PRISM specification. Although they develop a so-called PRISMGEN tool to automate this, their approach is impractical in reality and only supports two or three qubits because of the exponential growth of the number of states. Our approach does not need to enumerate such states in advance because a quantum state is directly formalized in Dirac notation with scalars. Moreover, rewrite rules are used to formalize the action of quantum gates and measurements, making our approach feasible to deal with more qubits.

Our symbolic approach to model checking quantum circuits is inspired by the work [1]. However, their approach is oriented to theorem proving, not model checking. They also use Dirac notation with a small set of laws to specify quantum states, quantum gates, measurements, and reasoning about quantum circuits in Coq, an interactive theorem prover. However, they usually require human users to provide necessary lemmas to complete their proofs, which are not easy tasks in general. Meanwhile, our approach is fully automatic and does not need any intervention from human users. Moreover, our implementation can be used as a general framework to formally specify and verify quantum circuits in a symbolic way in Maude.

VIII. Conclusion

We have proposed a symbolic approach to model checking quantum circuits by using a set of laws from quantum mechanics and basic matrix operations with Dirac notation. We have analyzed the quantum teleportation protocol as a case study to demonstrate the usefulness of our approach. Moreover, our implementation developed in Maude can be used as a general framework to formally specify and verify quantum circuits using our symbolic model checking approach. Our formalization takes the probabilities into account, and then we can tackle both qualitative and quantitative properties with the built-in LTL model checker in Maude. As one piece of our future work, we would like to conduct more case studies to demonstrate the usefulness of our approach.

References

[1] W. Shi, Q. Cao, Y. Deng, H. Jiang, and Y. Feng, "Symbolic reasoning about quantum circuits in coq," *J. Comput. Sci. Technol.*, vol. 36, no. 6, pp. 1291–1306, 2021.

[2] S. Gay, R. Nagarajan, and N. Papanikolaou, "Qmc: A model checker for quantum systems," 2007.

[3] Y. Feng, E. M. Hahn, A. Turrini, and L. Zhang, "Qpmc: A model checker for quantum programs and protocols," in *FM 2015: Formal Methods*, 2015, pp. 265–272.

[4] P. A. M. Dirac, "A new notation for quantum mechanics," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 35, no. 3, p. 416–418, 1939.

[5] J. Paykin, R. Rand, and S. Zdancewic, "Qwire: A core language for quantum circuits," *SIGPLAN Not.*, vol. 52, no. 1, p. 846–858, 2017.

[6] M. Clavel, et al., *All About Maude*, ser. Lecture Notes in Computer Science. Springer, 2007, vol. 4350.

[7] C. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. Wootters, "Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels," *Physical review letters*, vol. 70, pp. 1895–1899, 1993.

[8] R. Nagarajan and S. Gay, "Formal verification of quantum protocols," 2002.

[9] S. Gay, R. Nagarajan, and N. Papanikolaou, "Probabilistic model–checking of quantum protocols," 2005.