A Coalgebraic Semantics Framework for Quantum Systems

Ai Liu and Meng Sun

School of Mathematical Sciences, Peking University, China
{shaoai,sunm}@pku.edu.cn

Abstract. As a quantum counterpart of labeled transition system (LTS), quantum labeled transition system (QLTS) is a powerful formalism for modeling quantum programs or protocols, and gives a categorical understanding for quantum computation. With the help of quantum branching monad, QLTS provides a framework extending some ideas in nondeterministic or probabilistic systems to quantum systems. In this paper, we propose the notion of reactive quantum system (RQS), a variant of QLTS, and develop a coalgebraic semantics for both QLTS and RQS by an endofunctor on the category of convex sets, which has a final coalgebra. Such a coalgebraic semantics provides a unifying abstract interpretation for both QLTS and RQS. The notions of bisimulation and simulation can be employed to compare the behavior of different types of quantum systems and judge whether a coalgebra can be behaviorally simulated by another.

Keywords: Quantum labeled transition system, Reactive quantum system, Final coalgebra, Bisimulation, Simulation

1 Introduction

Quantum computation has been widely believed to bypass the end of Moore's Law and have an advantage over classical algorithms for certain problems. For instance, Grover's search algorithm can search an unordered array of size n in $O(\sqrt{n})$ time as opposed to the usual $O(n)$ time and Shor's factoring algorithm can factor numbers in polynomial time while no known classical algorithms can solve this problem in such time complexity [25]. There are not only theoretical curiosities for quantum computation, but also commercially available applications in quantum cryptography, which has a striking advantage over the classical cryptography in an "*unconditionally secure*" way [18]. Such advantages benefit from some prominent features of quantum mechanics, such as *superposition* and *entanglement*.

In order to develop quantum algorithms and protocols into concrete systems, foundations of quantum programming were investigated by Ying in [26], and a quantum programming language was introduced by Selinger in [22]. Beside implementing existing quantum algorithms and protocols, it is also important

to verify their correctness, which can be carried out by describing and analyzing theoretical models of quantum computation. *Quantum Turing machines* and *quantum circuits* are two fundamental models, which are computationally equivalent [16]. There have also been several high-level formalisms for quantum computation, such as *quantum functional programming* [9], *picturing quantum processes* [2] and qCCS, which is a variant of *Calculus of Communicating Systems* (CCS) with quantum flavor [28]. Moreover, quantum extensions of some classical computational models have been investigated recently, such as *quantum automata* [10], *quantum Markov chains* [15] and *communicating quantum processes* (CQP) which extends the pi-calculus with primitives for measurement and transformation of quantum states [8].

Equivalence checking has been a particularly relevant topic in quantum system verification. A technique using equivalence checking for verification of quantum protocols has been proposed in [1], which can go beyond stabilizer states and be used to verify protocols efficiently on all input states. It has been demonstrated that applicative bisimulation can be instantiated on the linear λ-calculi with quantum data in [14]. A probabilistic branching bisimulation for CQP has been proposed and shown to be a congruence in [3]. For qCCS, several kinds of bisimulation have been defined, such as open bisimulation [4], symbolic bisimulation [5], (approximate) strong bisimulation and weak bisimulation [6]. A software tool has been implemented to decide bisimilarity of qCCS configurations in [13]. Such bisimulations are defined concretely according to the labeled transition systems induced by the corresponding operational semantics of different quantum programming languages. In this paper, we propose a coalgebraic model of quantum systems and investigate a general notion of bisimulation naturally induced by it. This notion of bisimulation is defined in a highly abstract level, since only the acceptance probability of inputs is considered.

Coalgebra has emerged as a general framework for modeling *state-based transition systems* and covering different transition types: *non-deterministic*, *probabilistic* and so on [11]. There have been some coalgebraic models for quantum computation. For example, *Quantum labeled transition system* (QLTS) is defined in [20] to model quantum systems by using the *quantum branching monad* $\mathcal{Q}$ and following the principle of "*quantum data, classical control*". The abstract characterization of QLTS by coalgebras allows for applying the general trace and simulation theory [24] to quantum systems. A coalgebraic semantics for closed quantum systems (such as measure-once quantum finite automata [17]) is proposed in [21], which helps to relate the Schrödinger picture and Heisenberg picture of quantum mechanics with the dual concepts between algebras and coalgebras.

QLTS is a quantum extension of LTS whose transition structure is given by the coalgebra $X \to \mathcal{P}(\Sigma \times X)$. As well known, there is another type of LTS $X \to \mathcal{P}(X)^{\Sigma}$ capturing the behavior of reactive systems. This motivates us to develop the corresponding quantum extension of such reactive LTS types. In this paper, we propose the notion of *reactive quantum system* (RQS) based on the quantum branching monad [20]. Comparing with QLTS which is suitable for

describing simple quantum programs and protocols, RQS provides an appropriate formalism for quantum systems with reactive behavior, like measure-many quantum automata [12]. We then employ the endofunctor F on the category of convex sets and convex maps [21], and show how both QLTS and RQS can be specified as F-coalgebras. Different from qCCS configurations which involve density matrices with the same dimension, a product of different dimensional density matrices is used to represent a configuration of a QLTS or a RQS, and the set of configurations is taken as the carrier set of the corresponding coalgebraic model. The dynamics of the coalgebra specifies the evolution of configurations and the acceptance probability of the current configuration. One advantage of using the functor F is the existence of final coalgebra, which makes it easy to verify whether two configurations in an F-coalgebra are behaviorally equivalent. We prove that two configurations are F-bisimilar if and only if they are behaviorally equivalent, and show how the forward/backward morphism can be used to explore whether one F-coalgebra can be behaviorally simulated by another.

The rest of this paper is organized as follows. Section 2 recalls the definition of QLTS. The notion of quantum reactive systems is proposed in Section 3. In Section 4, we investigate the endofunctor F and specify both QLTS and RQS as F-coalgebras. In Section 5, we prove that the bisimulation relationship is equivalent to behavioral equivalence on the final F-coalgebra. The notion of *(weak) simulation* for RQS is studied in Section 6. Finally, Section 7 concludes and discusses possible future work.

2 Quantum Labeled Transition System

In this section, we recall some concepts and notations in quantum computation [18, 25, 26]. In quantum computation, pure states are often represented by unit vector states while mixed states are often represented by density matrices.

Definition 1 (vector state). *An* n-dimensional vector state s *is a column vector in a Hilbert space* $\mathbb{C}^n$*, denoted by* $|s\rangle = (c_1, \cdots, c_n)^T$*. We denote the conjugate transpose of* $|s\rangle$ *by* $\langle s|$*, which is a row vector* $(c_1^*, \cdots, c_n^*)$*. A vector state* $|s\rangle$ *is called unit if* $\langle s|\,|s\rangle = \sum_{i=1}^{n} c_i^* c_i = 1$*.*

Definition 2 (density matrix). *An* n-dimensional density matrix *is a positive semi-definite matrix* $\rho \in \mathbb{C}^{n\times n}$ *with* $\mathbf{tr}(\rho) \in [0, 1]$*, where* $\mathbf{tr}(\rho)$ *is the trace of* ρ*.*

The set of all n-dimensional density matrices is denoted by $\mathcal{DM}_n$. Note that density matrices are allowed to have a trace less than 1 in this paper, which represents the case when "some probability is missing". For any unit vector state $|s\rangle$, there is a corresponding density matrix $|s\rangle\langle s|$. Therefore, when it comes to quantum state transformations, we only consider the quantum operations acting on density matrices.

Definition 3 (quantum operation). *A* quantum operation *(QO) from a Hilbert space* $\mathbb{C}^m$ *to another Hilbert space* $\mathbb{C}^n$ *is a linear function* $\Phi : \mathcal{DM}_m \to \mathcal{DM}_n$ *satisfying the following conditions:*

- *(Trace non-increasing)* $\forall \rho \in \mathcal{DM}_m$, $\mathbf{tr}(\Phi(\rho)) \leq \mathbf{tr}(\rho)$.
- *(Completely positive) For any* I_k*, which is the identity map on* $(k \times k)$*-dimensional matrices, the form* $I_k \otimes \Phi$ *maps a positive semi-definite matrix to a positive semi-definite one, where* $\otimes$ *is the tensor product.*

The set of quantum operations from $\mathbb{C}^m$ to $\mathbb{C}^n$ is denoted by $\mathcal{QO}_{m,n}$. Kraus' theorem ensures that any quantum operation $\Phi \in \mathcal{QO}_{m,n}$ on a density matrix $\rho \in \mathcal{DM}_m$ can always be written as

$$\Phi(\rho) = \sum_k B_k \rho B_k^\dagger$$

for some set of $n \times m$-dimensional matrices $\{B_k\}$ satisfying $\sum_k B_k^\dagger B_k \leq I$, where $B_K^\dagger$ is the conjugate transpose of B_k. Thus, we can also denote a quantum operation Φ by $\{B_k\}$.

Now we recall the definition of quantum branching monad in [9], with which we can define QLTS and RQS.

Definition 4 (quantum branching monad [9]). *The quantum branching monad* $\mathcal{Q}$ *on the category of sets and functions is defined as follows:*

$$\mathcal{Q}(X) := \{c : X \to \prod_{m,n \in \mathbb{N}} \mathcal{QO}_{m,n} | \textit{the trace condition}\}$$
$$(\mathcal{Q}(f)(c)(y))_{m,n} := \sum_{x \in f^{-1}(y)} (c(x))_{m,n}$$

where $\prod_{m,n\in\mathbb{N}}$ *denotes a Cartesian product,* $(c(x))_{m,n} \in \mathcal{QO}_{m,n}$ *is the* (m,n)*-component of* $c(x) \in \prod_{i,j} \mathcal{QO}_{i,j}$ *and the trace condition is*

$$\sum_{x \in X} \sum_{n \in \mathbb{N}} \mathbf{tr}((c(x))_{m,n}(\rho)) \leq 1, \forall m \in \mathbb{N}, \forall \rho \in \mathcal{DM}_m.$$

The unit $\eta_X : X \to \mathcal{Q}(X)$ *and the multiplication* $\mu_X : \mathcal{Q}(\mathcal{Q}(X)) \to \mathcal{Q}(X)$ *are:*

$$(\eta_X(x)(x'))_{m,n} := \begin{cases} \{I_m\} & if\ x = x'\ and\ m = n \\ 0 & otherwise \end{cases}$$
$$(\mu_X(h)(x'))_{m,n} := \sum_{c \in \mathcal{Q}(X)} \sum_{k \in \mathbb{N}} ((c(x))_{k,n} \circ (h(c))_{m,k})$$

Then we have the following definition for QLTS:

Definition 5 (QLTS [20]). *A quantum labeled transition system* (X, s, c) *consists of a set* X *and a pair of functions* $s : \mathbf{1} \to \mathcal{Q}(X)$ *and* $c : X \to \mathcal{Q}(\Sigma \times X + \mathbf{1})$*, where* Σ *is an alphabet and* $\mathbf{1} = \{\surd\}$*, which is a singleton.*

One possible execution of a QLTS is as follows. Given any initial density matrix $\rho \in \mathcal{DM}_m$, ρ is taken into some state $x \in X$ of the system and evolves into

some density matrix $\rho' = (s(x))_{m,n}(\rho) \in \mathcal{DM}_n$. Then, in a transition between x and $x' \in X$, some action $a \in \Sigma$ occurs and ρ' evolves into $(c(x)(a,x')_{n,l})(\rho') \in \mathcal{DM}_l$. After finite iterations of transitions, if the current state is x' and the current density matrix is ρ', the system terminates to $c(x')(\surd)_{l,k}(\rho') \in \mathcal{DM}_k$.

The trace semantics of a QLTS (X, s, c) is defined as an arrow $\mathbf{1} \to \mathcal{Q}(\Sigma^*)$, which can be calculated recursively as follows. First define the unique function $h_c : X \to \mathcal{Q}(\Sigma^*)$ as

$$(h_c(x)(\langle\rangle))_{m,n} = (c(x)(\surd))_{m,n}$$
$$(h_c(x)(a \cdot \sigma))_{m,n} = \sum_{x' \in X} \sum_{k \in \mathbb{N}} (h_c(x')(\sigma))_{k,n} \circ (c(x)(a,x'))_{m,k}$$

where $a \in \Sigma, \sigma \in \Sigma^*$. Then, the trace semantics is obtained:

$$(trace_{s,c}(\sigma))_{m,n} = \sum_{x \in X} \sum_{k \in \mathbb{N}} (h_c(x)(\sigma))_{k,n} \circ (s(x))_{m,k}.$$

This operator $(trace_{s,c}(\sigma))_{m,n}$ can be seen as an accumulated quantum operation along paths that leads to a sequence of observations σ through the system. The probability of observing σ at an initial state $\rho \in \mathcal{DM}_m$ is represented as

$$P_{s,c}(\sigma, \rho) = \sum_{n \in \mathbb{N}} \mathbf{tr}((trace_{s,c}(\sigma))_{m,n}(\rho)) \in [0,1].$$

Example 1. Here we show an example of QLTS for describing quantum programs with output taken from [27], where a discrete coined quantum walk on an n-cycle with an absorbing boundary at position 1 is depicted as a quantum. Let H_C be a 2-dimensional coin space with orthonormal basis states $|0\rangle$ and $|1\rangle$, and H_V be the n-dimensional principle space spanned by the position vectors $|i\rangle : i = 0, \cdots, n-1$. We can formulate a quantum walk as a quantum loop:

$$\mathbf{while}(Mq \neq 1) \qquad \{\mathbf{output} \quad Mq; \quad q := Uq\}$$

where

$$M = \sum_{i=0}^{n-1} i \, |i\rangle \langle i| \otimes I_2, \quad U = S(I_n \otimes H),$$

and q is a quantum register in $H_V \otimes H_C$. The program can be interpreted by the following three steps:

1. A 'coin-operator' $H = |+\rangle \langle 0| + |-\rangle \langle 1|$ is applied to the coin, where
$$|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}, |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}.$$
2. A shift operator
$$S = \sum_{i=0}^{n-1} |i \ominus 1\rangle \langle i| \otimes |0\rangle \langle 0| + \sum_{i=0}^{n-1} |i \oplus 1\rangle \langle i| \otimes |1\rangle \langle 1|$$
is performed on the space $H_V \otimes H_C$, which makes the quantum walk one step left or right according to the coin state. Here $\ominus$ and $\oplus$ denote subtraction and addition modulo n, respectively.

3. Measure the principle system to see if the current position of the walk is 1. If the result is 'yes' then the walk terminates, otherwise output the result and the walk continues.

The QLTS describing this program is constructed as $(X, s : 1 \to X, c : X \to \mathcal{Q}(\Sigma \times X + 1))$, where $X = \{x\}$, $\Sigma = \{0, 2, \cdots, n-1\}$ and

$$\begin{aligned}(s(x))_{2n,2n} &= \{I_n \otimes I_2\}\\ (c(x)(k,x))_{2n,2n} &= \{U(P_{|k\rangle} \otimes I_2)\}\\ (c(x)(\surd))_{2n,2n} &= \{P_{|1\rangle} \otimes I_2\}\end{aligned}$$

in which $P_{|k\rangle} = |k\rangle \langle k|$ is a projection matrix of $|k\rangle$. A trace $k_1 \cdots k_m$ records the path of the quantum walk and the trace semantics is

$$\begin{aligned}&((trace_{s,c})(k_1 \cdots k_m))_{2n,2n}\\ =&\{P_{|1\rangle} \otimes I_2\} \circ \{U(P_{|k_m\rangle} \otimes I_2)\} \cdots \{U(P_{|k_1\rangle} \otimes I_2)\} \circ \{I_n \otimes I_2\}.\end{aligned}$$

If the initial position k is 1, there will be no valid trace (output); otherwise, the trace will start at $k_1 = k$ and end at $k_m = 0$ or $k_m = 2$. It is easy to prove that the probability of observing the trace $k_1 \cdots k_m$ with the initial position k_1 is $\frac{1}{2^m}$. Note that the initial state of the coin has no effect on the probability.

3 Reactive Quantum Systems

It is well known that LTS can be usually defined by one of the two transition structure types $\alpha : X \to \mathcal{P}(X)^\Sigma$ and $\beta : X \to \mathcal{P}(\Sigma \times X)$, where $\mathcal{P}$ is the powerset monad and Σ is an alphabet. Replacing the powerset monad $\mathcal{P}$ with the coproduct of distribution monad $\mathcal{D}$ and the termination possibility, $\mathcal{D} + \mathbf{1}$, α and β are changed to $\alpha' : X \to (\mathcal{D}(X) + \mathbf{1})^\Sigma$ and $\beta' : X \to \mathcal{D}(\Sigma \times X) + \mathbf{1}$, which capture the behavior of reactive and generative probabilistic systems, respectively [23]. Similarly, for the quantum case, if we replace $\mathcal{P}$ with $\mathcal{Q}(\_ + \mathbf{1})$, β changes to $c : X \to \mathcal{Q}(\Sigma \times X + \mathbf{1})$ constituting a QLTS, and we have a new transition structure $d : X \to \mathcal{Q}(X + \mathbf{1})^\Sigma$ corresponding to α, which motivates us to investigate on the notion of reactive quantum systems.

Definition 6. *A reactive quantum system (RQS) (X, s, d) is comprised of a set X and a pair of functions $s : \mathbf{1} \to \mathcal{Q}(X)$ and $d : X \to \mathcal{Q}(X + \mathbf{1})^\Sigma$, where Σ is an alphabet.*

The notion of RQS is similar to quantum Markov chains [15] but can involve different dimensional density matrices. Different from the trace semantics of QLTS, the trace semantics of RQS (X, s, d) can no longer be an arrow $\mathbf{1} :\to \mathcal{Q}(\Sigma)$, but be a function

$$track_{s,d} : \Sigma^* \to \prod_{m,n \in \mathbb{N}} \mathcal{QO}_{m,n}$$

which is defined as follows:

$$(track_{s,d}(\sigma))_{m,n} = \sum_{x\in X}\sum_{k\in\mathbb{N}}(h_d(x)(\sigma))_{k,n}\circ(s(x))_{m,k}.$$

where for all $a \in \Sigma$ and $\sigma \in \Sigma^*$:

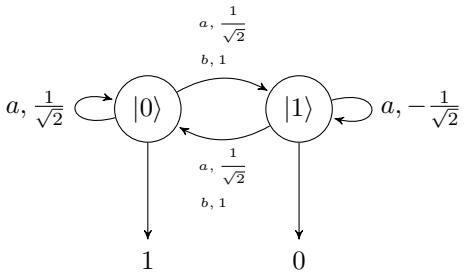
$$(h_d(x)(\langle a\rangle))_{m,n} = (d(x)(a)(\surd))_{m,n}$$
$$(h_d(x)(a\cdot\sigma))_{m,n} = \sum_{x'\in X}\sum_{k\in\mathbb{N}}(h_d(x')(\sigma))_{k,n}\circ(d(x)(a)(x'))_{m,k},$$

Given an input sequence $\sigma \in \Sigma^*$, a RQS may terminate after receiving a finite prefix of σ which means the rest of the input sequence is invalid. The acceptance probability of a nonempty input sequence σa with an initial state $\rho \in \mathcal{DM}_m$ can be recursively defined as:

$$P_{s,d}(\sigma a,\rho) = \sum_{n\in\mathbb{N}}\mathbf{tr}((track_{s,d}(\sigma a))_{m,n}(\rho)) + P(\sigma,\rho),$$

and $P(\emptyset,\rho) = 0$.

Example 2. The following state diagram is an example of a quantum automaton over the input set $\Sigma = \{a, b\}$ and the state space $\mathbb{C}^2$ with the standard basis $\{|0\rangle, |1\rangle\}$.



The transition matrices are

$$T_a = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = H, T_b = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}.$$

The outcomes of states indicate whether they are final states: if the outcome is 1 then it is a final state. When receiving an input character $r \in \Sigma$, the automaton changes to a new state by the corresponding transition and measures the new state with the basis $\{|0\rangle, |1\rangle\}$. If the result is $\{|0\rangle\}$, the automaton will stop receiving inputs and doing transitions; otherwise, it will continue receiving the next input. The RQS for this automaton is $(X = \{x\}, s : \mathbf{1} \to \mathcal{Q}(X), d : X \to \mathcal{Q}(X+\mathbf{1})^{\Sigma})$ where

$$(s(x))_{2,2} = \{\mathcal{I}_2\}$$
$$(d(x)(a)(x))_{2,2} = \{P_{|1\rangle}H\}$$
$$(d(x)(a)(\surd))_{2,2} = \{P_{|0\rangle}H\}$$
$$(d(x)(b)(x))_{2,2} = \{P_{|1\rangle}T_b\}$$
$$(d(x)(b)(\surd))_{2,2} = \{P_{|0\rangle}T_b\}.$$

If the initial state is $|0\rangle$ and the input sequence is ab, then the acceptance probability is

$$\begin{aligned}&\mathbf{tr}((track_{s,d}(ab))_{2,2})(|0\rangle\langle 0|) + \mathbf{tr}((trace)_{s,d}(a)_{2,2})(|0\rangle\langle 0|)\\ =&\mathbf{tr}(P_{|0\rangle}T_bP_{|1\rangle}H|0\rangle\langle 0|(P_{|0\rangle}T_bP_{|1\rangle}H)^\dagger) + \mathbf{tr}(P_{|0\rangle}H|0\rangle\langle 0|(P_{|0\rangle}H)^\dagger)\\ =&\frac{1}{2}+\frac{1}{2}=1.\end{aligned}$$

Example 2 is a concrete instance of measure-many quantum finite automata (MM-QFA) [12], which is defined as follows:

Definition 7 (MM-QFA [12]). *A measure-many (1-way) quantum automaton (MM-QFA) is a sextuple $M = (H, \Sigma, \{U_a\}_{a\in\Sigma}, |s\rangle, H_{acc}, H_{rej})$ where H is a finite dimensional Hilbert space, Σ is a finite alphabet, U_a is a unitary operator on H, $|s\rangle \in H$ is the initial state, which is a unit ket, H_{acc} and H_{rej} are, respectively, the accepting subspace and the rejecting subspace of H, such that $H_{acc} \cap H_{rej} = \emptyset$.*

We now show how to construct the corresponding RQS for a MM-QFA.

Note that a bounded operator A on a Hilbert space H is a unitary operator if $A^\dagger A = AA^\dagger = I$ and a projection if $A = A^2$. The computation of a MM-QFA on an input sequence $w_1w_2\cdots w_n$ goes as follows. The operator U_{w_1} is first applied to the initial state $|s\rangle$ and then measure the resulting state $U_{w_1}|s\rangle$, which projects $U_{w_1}|s\rangle$ into a vector $|\phi'\rangle$ of one of the subspaces $H_{acc}, H_{rej}, H_{non}$, where H_{non} is the orthogonal complement of $H_{acc}\oplus H_{rej}$. In all cases the computation continues only if a projection P_{non} into H_{non} occurs. When no termination occurs, operators $U_{w_1}, \cdots, U_{w_n}$ are applied one after another, and after each such application the measurement is performed. The result of the computation can be seen as an application of the composed operator $U'_{w_n}\cdots U'_{w_1}|s\rangle$, where $U'_{w_i} = P_{non}U_{w_i}$. The probability that $w = w_1\cdots w_n$ is accepted can be recursively defined as

$$f(w) = \begin{cases} f(w') + \|P_{acc}U'_{w_n\cdots w_1}|s\rangle\|^2 & w' = w_1\cdots w_{n-1}\\ 0 & w = \emptyset\end{cases},$$

where P_{acc} is the projection into H_{acc}.

Given a MM-QFA $(H, \Sigma, \{U_a\}_{a\in\Sigma}, |s\rangle, H_{acc}, H_{rej})$ based on an n-dimensional Hilbert space, we can construct its corresponding RQS (X, s, d), where $X = \{x\}$ and for $a \in \Sigma$

$$\begin{aligned}(s(x))_{n,n} &= \{\mathcal{I}_n\}\\ (d(x)(a)(x))_{n,n} &= \{P_{non}U_a\}\\ (d(x)(a)(\surd))_{n,n} &= \{P_{acc}U_a\}.\end{aligned}$$

For any input sequence $\sigma \in \Sigma^*$, it is easy to verify the acceptance probability for the MM-QFA and the corresponding RQS are equal, which means

$$f(\sigma) = P_{s,d}(\sigma, |s\rangle\langle s|).$$

4 A Unifying Coalgebraic Framework for QLTS and RQS

Both QLTS and RQS follow the "*quantum data, classical control*" rule [22]. However, from the observers' perspective, the "classical control" part may be hidden from outside, and thus we focus on the change of "quantum data" and the probability of certain observational sequences. In this paper, we use the endofunctor

$$FX = X^{\Sigma} \times [0,1] \tag{1}$$

on the category **Conv** consisting of convex sets and convex maps, where Σ is an alphabet. Both QLTS and RQS can be modeled as F-coalgebras.

The reason for working on the category **Conv** is to ensure that the transitions of coalgebras are valid and any convex combination of states is still a state. For any density matrices $\rho_1, \cdots, \rho_r$ in $\mathcal{DM}_m$ and any positive numbers $\lambda_1, \cdots, \lambda_r$ such that $\lambda_1 + \cdots + \lambda_r = 1$,

$$\mathbf{tr}(\sum_{i=1}^{r} \lambda_i \rho_i) = \sum_{i=1}^{r} \lambda_i \mathbf{tr}(\rho_i) \leq \sum_{i=1}^{k} \lambda_i = 1$$

Then we can get $\sum_{i=1}^{r} \lambda_i \rho_i \in \mathcal{DM}_m$. Therefore, $\mathcal{DM}_m (m \in \mathbb{N})$ is a convex set. We denote the product $\prod_{n\in\mathbb{N}} \mathcal{DM}_n$ by $\widehat{\mathcal{DM}}$. Due to the existence of infinite product in the category **Conv**, $\widehat{\mathcal{DM}}$ is also a convex set.

A configuration of a QLTS $(X, s : \mathbf{1} \to X, c : X \to \mathcal{Q}(\Sigma \times X + \mathbf{1}))$ can be represented by an element γ in the convex set $\prod_{x\in X} \widehat{\mathcal{DM}} \cong \widehat{\mathcal{DM}}^{|X|}$. It means that the QLTS is likely to be in multiple positions with different density matrices simultaneously. Using $(\gamma)_{x,n}$ ($\gamma \in \prod_{x\in X} \widehat{\mathcal{DM}}$) to represent the $(n \times n)$-dimensional density matrix at the position x, we can have the corresponding F-coalgebra $CF_{s,c} = (\prod_{x\in X} \widehat{\mathcal{DM}}, \langle n_c, o_c \rangle)$, where for any $a \in \Sigma$,

$$(n_c(\gamma, a))_{x,n} = \sum_{k\in\mathbb{N}} \sum_{x'\in X} (c(x')(a, x))_{k,n}((\gamma)_{x',k})$$

$$o_c(\gamma) = \sum_{n\in\mathbb{N}} \sum_{k\in\mathbb{N}} \sum_{x\in X} \mathbf{tr}((c(x)(\surd))_{k,n}((\gamma)_{x,k}))$$

From the trace condition for quantum branching monad, it is easy to prove that n_c and o_c are convex maps. For $a \in \Sigma$ and $\sigma \in \Sigma^*$, let $n_c(\gamma, aw) = n_c(n_c(\gamma, a), w)$.

Lemma 1. *Given an initial density matrix $\rho \in \mathcal{DM}_m$, the initial configuration $(\gamma_0)_{x,n} = (s(x))_{m,n}(\rho)$. For any $\sigma \in \Sigma^*$, the following equations hold:*

$$(trace_{s,c}(\sigma))_{m,n}(\rho) = \sum_{k\in\mathbb{N}} \sum_{x'\in X} (c(x')(\surd))_{k,n}(n(\gamma_0, \sigma))$$

$$P_{s,c}(\sigma, \rho) = o_c(n_c(\gamma_0, \sigma))$$

Proof. The proof follows from the definition of trace semantics and mathematical induction.

Lemma 2. $\forall \gamma \in \prod_{x\in X} \widehat{\mathcal{DM}}. \sum_{\sigma\in\Sigma^*} o_c(n_c(\gamma,\sigma)) \leq \sum_{x\in X}\sum_{n\in\mathbb{N}} \mathbf{tr}((\gamma)_{x,n})$.

Proof. From the trace condition of quantum branching monad, we can assume that

$$\sum_{x\in X}\sum_{n\in\mathbb{N}} \mathbf{tr}((c(x))_{m,n}(\rho)) \leq \mathbf{tr}(\rho), \forall m \in \mathbb{N}, \forall \rho \in \mathcal{DM}_m.$$

If $\exists \rho \in \mathcal{DM}_m, \sum_{x\in X}\sum_{n\in\mathbb{N}} \mathbf{tr}((c(x))_{m,n}(\rho)) > \mathbf{tr}(\rho)$, let $\rho' = \frac{\rho}{\mathbf{tr}(\rho)} \in \mathcal{DM}_m$ and from linearity of quantum operations we can get

$$\sum_{x\in X}\sum_{n\in\mathbb{N}} \mathbf{tr}((c(x))_{m,n}(\rho')) > 1,$$

which is a contradiction with the trace condition. Combing the definitions of n_c and o_c, we can get

$$\sum_{x\in X}\sum_{n\in\mathbb{N}}\sum_{a\in\Sigma} \mathbf{tr}((n_c(\gamma,a))_{x,n}) + o_c(\gamma) \leq \sum_{x\in X}\sum_{n\in\mathbb{N}} \mathbf{tr}((\gamma)_{x,n}).$$

With this inequality, the lemma can be easily proved by using mathematical induction on the length of σ.

Theorem 1. $\forall \rho \in \mathcal{DM}_m. \sum_{\sigma\in\Sigma^*} P_{s,c}(\sigma,\rho) \leq \mathbf{tr}(\rho) \leq 1$.

Proof. The proof follows Lemma 1 and Lemma 2.

In the case of RQS, since after receiving a finite prefix of the input sequence, the system may transform into some density matrix in the final state $\surd$, we need to record density matrices at $\surd$. Thus, a configuration of the RQS $(X, s : \mathbf{1} \to X, d : X \to \mathcal{Q}(\mathbf{1} + X^{\Sigma}))$ can be represented by an element γ in the convex set $\prod_{x\in X\cup\{\surd\}} \widehat{\mathcal{DM}}$. The corresponding F-coalgebra $CF_{s,d} = (\prod_{x\in X\cup\{\surd\}} \widehat{\mathcal{DM}}, \langle n_d, o_d\rangle)$ is defined as follows where $x \in X$ and $a \in \Sigma$:

$$(n_d(\gamma,a))_{x,n} = \sum_{m\in\mathbb{N}}\sum_{x'\in X} (d(x')(a,x))_{m,n}((\gamma)_{x',m})$$

$$(n_d(\gamma,a))_{\surd,n} = (\gamma)_{\surd,n} + \sum_{m\in\mathbb{N}}\sum_{x\in X} (d(x)(\surd))_{m,n}((\gamma)_{x,m})$$

$$o_d(\gamma) = \sum_{n\in\mathbb{N}} \mathbf{tr}((\gamma)_{\surd,n})$$

Lemma 3. *With an initial density matrix $\rho \in \mathcal{DM}_m$, we have the initial configuration*

$$(\gamma_0)_{x,n} = (s(x))_{m,n}(\rho)$$
$$(\gamma_0)_{\surd,n} = 0$$

and for $\sigma \in \Sigma^*$,

$$(track_{s,d}(\sigma))_{m,n}(\rho) = \sum_{k\in\mathbb{N}}\sum_{x\in X} (d(x)(\surd))_{k,n}(n(\gamma_0,\sigma))$$
$$P_{s,d}(\sigma,\rho) = o_d(n_d(\gamma_0,\sigma)) \leq 1.$$

Proof. It can be proved by using the definition of the trace semantics and the mathematical induction.

5 Final Coalgebra and Bisimulation

For a functor T, a T-coalgebra $\omega : \Omega \to T\Omega$ is called a final coalgebra if it is a final object in the category of T-coalgebras and T-homomorphisms. The final coalgebra $([0,1]^{\Sigma^*}, \langle n_f, o_f \rangle)$ exists for the functor $FX = X^\Sigma \times [0,1]$, where $o_f(\beta) = \beta(\langle, \rangle)$ and $n_f(\beta)(a)(\sigma) = \beta(a\sigma)$. From any F-coalgebra $(X, \langle n, o \rangle)$ the unique behavior map $b : X \to [0,1]^{\Sigma^*}$ assigns to each state its behavior $b(x)(\sigma) = o(n(x, \sigma))$ and makes the following diagram commute:

$$\begin{array}{ccc} X & \xrightarrow{\quad b \quad} & [0,1]^{\Sigma^*} \\ {\scriptstyle \langle n, o \rangle}\downarrow & & \downarrow{\scriptstyle \langle n_f, o_f \rangle} \\ X^\Sigma \times [0,1] & \xrightarrow[b^\Sigma \times \mathrm{id}]{} & ([0,1^{\Sigma^*}])^\Sigma \times [0,1] \end{array}$$

We now come to the observational equivalence relationship for F-coalgebras. In coalgebra theory, there are two well-know notions of observational equivalence: *bisimulation* and *behavioral equivalence*. A bisimulation between two systems is intuitively a relation between their states together with a transition structure on it, while behavioral equivalence shows that two states can be mapped into the same state in another F-coalgebra by F-homomorphisms.

It has been demonstrated in [20] that, for the quantum branching monad $\mathcal{Q}$, $\mathcal{Q}$-bisimulation and behavioral equivalence do not coincide. However, for the functor F defined in (1), F-bisimulation coincides with behavioral equivalence.

Theorem 2. *Given two F-coalgebras $(X, \langle n_X, o_X \rangle)$ and $(Y, \langle n_Y, o_Y \rangle)$, $x \in X$ and $y \in Y$ are F-bisimlar iff they are behaviorally equivalent.*

Proof. ($\Rightarrow$) If x and y are F-bisimilar, there is a F-coalgebra $(R, \langle n_R, o_R \rangle)$ where $R \subseteq X \times Y$ such that the projection functions $\pi_1 : R \to X$ and $\pi_2 : R \to Y$ are F-homomorphisms and $(x, y) \in R$. Let b_X, b_Y, b_R be the behavioral maps from $(X, \langle n_X, o_X \rangle), (Y, \langle n_Y, o_Y \rangle), (R, \langle n_R, o_R \rangle)$ to the final coalgebra $([0,1]^{\Sigma^*}, \langle n_f, o_f \rangle)$. Due to the uniqueness of the behavior map, $b_X \circ \pi_1 = b_R = b_Y \circ \pi_2$. Thus $b_X(x) = b_X \circ \pi_1(x, y) = b_Y \circ \pi_2(x, y) = b_Y(y)$.

($\Leftarrow$) If x and y are behaviorally equivalent, we have $b_X(x) = b_Y(y)$. Let $R' = \{(u, v) | b_X(u) = b_Y(v)\}$. Since b_X and b_Y are both F-homomorphisms, if $b_X(u) = b_Y(v)$, we have $o_X(u) = o_Y(v)$ and $b_X(n_X(u)) = b_Y(n_Y(u))$. Let $n_{R'}(u, v) = (n_X(u), n_Y(v))$ and $o_{R'}(u, v) = o_X(u) = o_Y(v)$. Then we have that R' is a F-bisimulaiton.

Theorem 3. *Let $(X_1, \langle n_1, o_1 \rangle)$ and $(X_2, \langle n_2, o_2 \rangle)$ be two F-coalgebras. Two states $x_1 \in X_1$ and $x_2 \in X_2$ are behaviorally equivalent, iff they are mapped into the same state in the final coalgebra $([0,1]^{\Sigma^*}, \langle n_f, o_f \rangle)$.*

Proof. ($\Leftarrow$) By definition.

($\Rightarrow$) If x_1 and x_2 are behaviorally equivalent, then there exists a F-coalgebra $(X_3, \langle n_3, o_3 \rangle)$ and two F-homomorphism $f : X_1 \to X_3$ and $g : X_2 \to X_3$ with

$f(x_1) = g(x_2)$. Since $([0,1]^{\Sigma^*}, \langle n_f, o_f \rangle)$ is the final coalgebra, there exists a unique behavioral map b from $(X_3, \langle n_3, o_3 \rangle)$ to $([0,1]^{\Sigma^*}, \langle n_f, o_f \rangle)$. Due to the uniqueness of the behavior map b, $b \circ f$ and $b \circ g$ should be the behavioral maps from $(X_1, \langle n_1, o_1 \rangle)$ and $(X_2, \langle n_2, o_2 \rangle)$ to the final coalgebra $([0,1]^{\Sigma^*}, \langle n_f, o_f \rangle)$. Thus, $b \circ f(x_1) = b(f(x_1)) = b(g(X_2)) = b \circ g(x_2)$.

Theorem 4. *Given two QLTSs (X_1, s_1, c_1) and (X_2, s_2, c_2) with the same alphabet and their corresponding F-coalgebras CF_{s_1,c_1} and CF_{s_2,c_2}, if $trace_{s_1,c_1} = trace_{s_2,c_2}$ then for any density matrix $\rho \in \mathcal{DM}_m$, the corresponding initial configurations $(\gamma_0)_{x,n} = (s_1(x))_{m,n}(\rho), x \in X_1$ and $(\gamma_0')_{x,n} = (s_2(x))_{m,n}(\rho), x \in X_2$ are behaviorally equivalent.*

Proof. By definition.

Theorem 5. *Given two RQSs (X_1, s_1, d_1) and (X_2, s_2, d_2) with the same input alphabet and their corresponding F-coalgebras CF_{s_1,d_1} and CF_{s_2,d_2}, if $track_{s_1,d_1} = track_{s_2,d_2}$ then for any density matrix $\rho \in \mathcal{DM}_m$, the corresponding initial configurations*

$$
\begin{aligned}
(\gamma_0)_{x,n} &= (s_1(x))_{m,n}(\rho), x \in X_1 \\
(\gamma_0)_{\surd,n} &= 0 \\
(\gamma_0')_{y,n} &= (s_2(x))_{m,n}(\rho), x \in X_2 \\
(\gamma_0')_{\surd,n} &= 0
\end{aligned}
$$

are behaviorally equivalent.

Proof. By definition.

Theorem 6. *Given a QLTS (X, s, c) with the corresponding F-coalgebra $CF_{s,c}$ and a RQS (Y, t, d) with the corresponding F-coalgebra $CF_{t,d}$, $\gamma \in \prod_{x \in X} \widehat{\mathcal{DM}}$ and $\gamma' \in \prod_{y \in Y \cup \{\surd\}} \widehat{\mathcal{DM}}$ are behaviorally equivalent iff their behavior are both the empty map $\epsilon : \epsilon(\sigma) = 0, \sigma \in \Sigma^*$.*

Proof. ($\Leftarrow$) By definition.

($\Rightarrow$) Let b (b') be the behavior map from $CF_{s,c}$ ($CF_{t,d}$) to the final coalgebra $([0,1]^{\Sigma^*}, \langle n_f, o_f \rangle)$. If $b'(\gamma')$ is not ϵ, there exists σ' such that $b'(\gamma')(\sigma') > 0$. Since $b'(\gamma')(\sigma' a) \geq b'(\gamma')(\sigma')$, it is easy to get $\sum_{\sigma \in \Sigma^*} b'(\gamma')(\sigma) > 1$. If $b(\gamma) = b'(\gamma')$, then $\sum_{\sigma \in \Sigma^*} b(\gamma)(\sigma) > 1$, which is a contradiction with Lemma 2.

6 Simulation

Bisimulation relations require two bisimilar states to exhibit identical behavior. On the contrary, simulation relations are pre-orders on the state space which requires that whenever state y simulates state x, y can mimic all the behavior of x and the reverse is not guaranteed.

Definition 8 (Löwner partial order). *The order $\leq$ on the set $\mathcal{DM}_m$ of density matrices is defined by: $\rho_1 \leq \rho_2$ iff $\rho_2 - \rho_1$ is positive semi-definite.*

The following definition provides two possible orders for quantum operations, which is originally defined in [7].

Definition 9. *Let $\Phi, \Psi \in \mathcal{QO}_{m,n}$. There are two orders for quantum operations:*

- $\Phi \sqsubseteq \Psi$ *if* $\forall \rho \in \mathcal{DM}_m$, $\Phi(\rho) \leq \Psi(\rho)$.
- $\Phi \lesssim \Psi$ *if* $\forall \rho \in \mathcal{DM}_m$, $\mathbf{tr}(\Phi(\rho)) \leq \mathbf{tr}(\Psi(\rho))$.

Note that $\sqsubseteq$ is a partial order and thus also a pre-order, while $\lesssim$ is a pre-order.

Definition 10 (simulation for RQS). *Let (X, s, d) and (Y, t, e) be two RQSs with the same input alphabet Σ, for $a \in \Sigma$ and $m, n \in \mathbb{N}$:*

- *A* forward simulation *from (X, s, d) to (Y, t, e) is a function $f : X \to \mathcal{Q}Y$ that satisfies:*

$$\sum_{x \in X} \sum_{k \in \mathbb{N}} (f(x)(y))_{k,n} \circ (s(x))_{m,k} \sqsubseteq (t(y))_{m,n}$$

$$\sum_{x' \in X} \sum_{k \in \mathbb{N}} (f(x')(y))_{k,n} \circ (d(x)(a)(x'))_{m,k} \sqsubseteq \sum_{y' \in Y} \sum_{k \in \mathbb{N}} (e(y')(a)(y))_{k,n} \circ f(x)(y')_{m,k}$$

$$(d(x)(\surd))_{m,n} \sqsubseteq \sum_{y \in Y} \sum_{k \in \mathbb{N}} (e(y)(\surd))_{k,n} \circ (f(x)(y))_{m,k}.$$

- *A* backward simulation *from (X, s, d) to (Y, t, e) is a function $f : X \to \mathcal{Q}(Y)$ that satisfies:*

$$(t(y))_{m,n} \sqsubseteq \sum_{x \in X} \sum_{k \in \mathbb{N}} (f(x)(y))_{k,n} \circ (s(x))_{m,k}$$

$$\sum_{y' \in Y} \sum_{k \in \mathbb{N}} (e(y')(a)(y))_{k,n} \circ f(x)(y')_{m,k} \sqsubseteq \sum_{x' \in X} \sum_{k \in \mathbb{N}} (f(x')(y))_{k,n} \circ (d(x)(a)(x'))_{m,k}$$

$$\sum_{y \in Y} \sum_{k \in \mathbb{N}} (e(y)(\surd))_{k,n} \circ (f(x)(y))_{m,k} \sqsubseteq (d(x)(\surd))_{m,n}$$

If there exists a forward simulation (backward simulation) from (X, s, d) to (Y, t, e), we denote $(X, s, d) \sqsubseteq_F (Y, t, e)$ ($(X, s, d) \sqsubseteq_B (Y, t, e)$).

Definition 11 (weak simulation for RQS). *By replacing the order $\sqsubseteq$ with $\lesssim$ in the inequations in Definition 10, we can get the corresponding notion of* weak forward (backward) simulation *from (X, s, d) to (Y, t, e), denoted $(X, s, d) \lesssim_F (Y, t, e)$ ($(X, s, d) \lesssim_B (Y, t, e)$).*

Definition 12 (forward/backward morphism). *For two $\mathcal{F}$-coalgebras $(U, \alpha : U \to \mathcal{F}U)$ and $(V, \beta : V \to \mathcal{F}V)$, a forward morphism $h : \alpha \to \beta$ with respect to a simulation preorder $\leq$ is a homomorphism from U to V such that $\mathcal{F}h \cdot \alpha \leq \beta \cdot h$. Dually, h is called a backward morphism if $\beta \cdot h \leq \mathcal{F}h \cdot \alpha$.*

Theorem 7. *Given two RQSs (X, s, d) and (Y, t, e) with the same alphabet and their corresponding F-coalgebras $CF_{s,d} = (\prod_{x \in X \cup \{\surd\}} \widehat{\mathcal{DM}}, \langle n_d, o_d \rangle)$ and $CF_{t,e} = (\prod_{y \in Y \cup \{\surd\}} \widehat{\mathcal{DM}}, \langle n_e, o_e \rangle)$, if $(X, s, d) \sqsubseteq_F (Y, t, e)$ ($(X, s, d) \sqsubseteq_B (Y, t, e)$), there exists a forward (backward) morphism from $CF_{s,d}$ to $CF_{t,e}$.*

Proof. If $(X,s,d) \sqsubseteq_F (Y,t,e)$, there exists a forward simulation $f : X \to Q(Y)$ from (X,s,d) to (Y,t,e). We define the function $\overline{f} : \prod_{x\in X\cup\{\surd\}} \widehat{\mathcal{DM}} \to \prod_{y\in Y\cup\{\surd\}} \widehat{\mathcal{DM}}$ as:

$$(\overline{f}(\gamma))_{y,n} = \sum_{x\in X}\sum_{k\in\mathbb{N}} (f(x)(y))_{k,n}((\gamma)_{x,k}),\ (\overline{f}(\gamma))_{\surd,n} = (\gamma)_{\surd,n}.$$

Let $(g,r),(g',r') \in (\prod_{x\in X\cup\{\surd\}} \widehat{\mathcal{DM}})^{\Sigma} \times [0,1]$. We define $(g,r) \leq (g',r')$ if for any $a \in \Sigma$, $(g(a))_{x,n} \leq (g'(a))_{x,n}$, $(g(a))_{\surd,n} \leq (g'(a))_{\surd,n}$ and $r \leq r'$. It is easy to verify this order is a pre-order and $\overline{f}$ is a forward morphism from $CF_{s,d}$ to $CF_{t,e}$. Analogously, if $(X,s,d) \sqsubseteq_B (Y,t,e)$, we can find a backward morphism from $CF_{s,d}$ to $CF_{t,e}$.

Theorem 8. *Given two RQSs (X,s,d) and (Y,t,e) with the same alphabet and their corresponding F-coalgebras $CF_{s,d} = (\prod_{x\in X\cup\{\surd\}} \widehat{\mathcal{DM}}, \langle n_d, o_d\rangle)$ and $CF_{t,e} = (\prod_{y\in Y\cup\{\surd\}} \widehat{\mathcal{DM}}, \langle n_e, o_e\rangle)$, if $(X,s,d) \lesssim_F (Y,t,e)$ ($(X,s,d) \lesssim_B (Y,t,e)$), there exists a forward (backward) morphism from $CF_{s,d}$ to $CF_{t,e}$.*

Proof. The proof is similar to Theorem 7 except for the simulation order. Let $(g,r),(g',r') \in (\prod_{x\in X\{\surd\}} \widehat{\mathcal{DM}})^{\Sigma} \times [0,1]$ and here we define $(g,r) \leq (g',r')$ if for any $a \in \Sigma$, $\mathbf{tr}((g(a))_{x,n}) \leq \mathbf{tr}((g'(a))_{x,n})$, $\mathbf{tr}((g(a))_{\surd,n}) \leq \mathbf{tr}((g'(a))_{\surd,n})$ and $r \leq r'$.

7 Conclusion and Future Work

In this paper we propose the notion of reactive quantum system as a variant of QLTS, and provide a unifying coalgebraic semantic framework for both QLTS and RQS. In fact, the coalgebraic models for QLTS and RQS have the same behavior shape captured by the functor $F = -^{\Sigma} \times [0,1]$, for which the final coalgebra exists. Then we define the general notions of behavioral equivalence, bisimulation and simulation, with which we can compare the similarity of different quantum systems.

Besides QLTS and RQS, several other formalisms of quantum systems, such as quantum Turing machine [19], quantum process algebra [28] and quantum Markov decision processes [29], have been investigated in literature. On the other hand, a coalgebraic semantics of closed quantum systems has been proposed in [21]. One possible future work is to integrate more types of quantum systems into the unifying coalgebraic framework, then we can explore the relations between them coalgebraically.

Acknowledgement

We are indebted to Luis S. Barbosa and Renato Neves for helpful discussions regarding the coalgebraic framework. This work has been supported by the National Natural Science Foundation of China under grant no. 61772038, 61532019

and 61272160, and the Guangdong Science and Technology Department (Grant no.2018B010107004).

References

1. Ebrahim Ardeshir-Larijani, Simon J. Gay, and Rajagopal Nagarajan. Equivalence checking of quantum protocols. In *Proceedings of TACAS 2013*, volume 7795 of *LNCS*, pages 478–492. Springer, 2013.
2. Bob Coecke and Aleks Kissinger. *Picturing quantum processes.* Cambridge University Press, 2017.
3. Timothy A. S. Davidson. *Formal verification techniques using quantum process calculus.* PhD thesis, University of Warwick, Coventry, UK, 2012.
4. Yuxin Deng and Yuan Feng. Open bisimulation for quantum processes. In *Proceedings of TCS 2012*, volume 7604 of *LNCS*, pages 119–133. Springer, 2012.
5. Yuan Feng, Yuxin Deng, and Mingsheng Ying. Symbolic bisimulation for quantum processes. *ACM Transactions on Computational Logic*, 15(2):14:1–14:32, 2014.
6. Yuan Feng, Runyao Duan, and Mingsheng Ying. Bisimulation for quantum processes. *ACM Transactions on Programming Languages and Systems*, 34(4):17:1–17:43, 2012.
7. Yuan Feng, Nengkun Yu, and Mingsheng Ying. Model checking quantum markov chains. *Journal of Computer and System Sciences*, 79(7):1181–1198, 2013.
8. Simon J. Gay and Rajagopal Nagarajan. Communicating quantum processes. In *Proceedings of POPL 2005*, pages 145–157. ACM, 2005.
9. Ichiro Hasuo and Naohiko Hoshino. Semantics of higher-order quantum computation via geometry of interaction. *Annals of Pure and Applied Logic*, 168(2):404–469, 2017.
10. Mika Hirvensalo. Quantum automata theory - A review. In Werner Kuich and George Rahonis, editors, *Algebraic Foundations in Computer Science - Essays Dedicated to Symeon Bozapalidis on the Occasion of His Retirement*, volume 7020 of *LNCS*, pages 146–167. Springer, 2011.
11. Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 2016.
12. Attila Kondacs and John Watrous. On the power of quantum finite state automata. In *Proceedings of FOCS 1997*, pages 66–75. IEEE Computer Society, 1997.
13. Takahiro Kubota, Yoshihiko Kakutani, Go Kato, Yasuhito Kawano, and Hideki Sakurada. Application of a process calculus to security proofs of quantum protocols. In *Proceedings of the International Conference on Foundations of Computer Science (FCS)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2012.
14. Ugo Dal Lago and Alessandro Rioli. Applicative bisimulation and quantum λ-calculi. In *Proceedings of FSEN 2015*, volume 9392 of *LNCS*, pages 54–68. Springer, 2015.
15. Lvzhou Li and Yuan Feng. Quantum Markov chains: Description of hybrid systems, decidability of equivalence, and model checking linear-time properties. *Information and Computation*, 244:229–244, 2015.
16. Abel Molina and John Watrous. Revisiting the simulation of quantum turing machines by quantum circuits. *CoRR*, abs/1808.01701, 2018.

17. Cristopher Moore and James P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(1-2):275–306, 2000.
18. Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2002.
19. Harumichi Nishimura and Masanao Ozawa. Perfect computational equivalence between quantum turing machines and finitely generated uniform quantum circuit families. *Quantum Information Processing*, 8(1):13–24, 2009.
20. Hiroshi Ogawa. Coalgebraic approach to equivalences of quantum systems. Master's thesis, University of Tokyo, 2014.
21. Frank Roumen. Coalgebraic quantum computation. In *Proceedings of QPL 2012*, volume 158 of *EPTCS*, pages 29–38, 2014.
22. Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.
23. Ana Sokolova. *Coalgebraic analysis of probabilistic systems*. PhD thesis, Technische Universiteit Eindhoven, 2005.
24. Natsuki Urabe and Ichiro Hasuo. Generic forward and backward simulations III: quantitative simulations by matrices. In *Proceedings of CONCUR 2014*, volume 8704 of *LNCS*, pages 451–466. Springer, 2014.
25. Noson S. Yanofsky and Mirco Mannucci. *Quantum Computing for Computer Scientists*. Cambridge University Press, 2008.
26. Mingsheng Ying. *Foundations of Quantum Programming*. Morgan Kaufmann, 2016.
27. Mingsheng Ying and Yuan Feng. Quantum loop programs. *Acta Informatica*, 47(4):221–250, 2010.
28. Mingsheng Ying, Yuan Feng, Runyao Duan, and Zheng-Feng Ji. An algebra of quantum processes. *ACM Transactions on Computational Logic*, 10(3):19:1–19:36, 2009.
29. Shenggang Ying and Mingsheng Ying. Reachability analysis of quantum Markov decision processes. *Information and Computation*, 263:31–51, 2018.