COALGEBRAIC APPROACH TO EQUIVALENCES OF QUANTUM SYSTEMS
余代数を用いた量子的なシステムの等価性の考察

by

Hiroshi Ogawa
小川 浩志

A Senior Thesis
卒業論文

Submitted to
the Department of Information Science
the Faculty of Science, the University of Tokyo
on March 1, 2014
in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science

Thesis Supervisor: Ichiro Hasuo　蓮尾　一郎
Lecturer of Information Science

ABSTRACT

Quantum computation is commonly believed to have an advantage over classical one for certain problems or situations. In the context of communication protocols, quantum mechanics helps us to accomplish "unconditional security" that cannot be obtained by any classical protocols. As quantum cryptographic systems come into commercial use, there has been growing attention to formal verification technique for quantum protocols. To formalize quantum systems, we employ the notion of coalgebra, which is a mathematical/categorical framework that captures a wide variety of state-based transition systems. This abstract framework canonically extends some ideas in non-deterministic or probabilistic systems to quantum systems. In the thesis, we focus on three instances of such extensions. First, trace semantics and forward/backward simulation is defined with regard to quantum labeled transition systems (QLTS) that we introduce as a quantum counterpart of LTS or probabilistic LTS. We also illustrate them with some basic examples, such as the superdense coding protocol and the quantum teleportation protocol. Next, we discuss about two different types of equivalence, namely bisimilarity and behavioral equivalence. While it is known that the two notions coincide in probabilistic systems, we show that they do not coincide in quantum systems. Finally, by means of coalgebraic modal logic, we obtain modal logic that is expressive to recognize an behavioral equivalence of two quantum systems.

論文要旨

量子計算は特定の問題において古典計算を凌ぐものとして信じられている。特に通信プロトコルでは、量子力学の基本的な原理によって古典力学では達成し得ない完全な安全性を可能にしている。量子暗号システムが商業的に実用化されるなか、量子プロトコルの安全性を形式的に検証する手法についても重要性が高まってきている。そこで私達は量子的なシステムを形式化する手法として余代数という概念を用いた。余代数とは数学の一分野の圏論における概念であり、様々な種類の状態遷移系が抽象化して扱われる。この抽象的な枠組みは非決定的、確率的システムにおける手法を自然に量子的なシステムに拡張することが可能になり、この論文ではそのような拡張の例として次の三つを挙げる。一つ目として、軌跡意味論と順方向、逆方向の模倣関係を量子的ラベル付き遷移系に対して定義する。量子的ラベル付き遷移系とは非決定的ラベル付き遷移系や確率的ラベル付き遷移系の量子版である。これらの概念を量子プロトコルの基本的な例である高密度符号化プロトコルと量子テレポーテーションプロトコルをもとに解説する。二つ目として、二種類の等価性の概念である、振る舞い等価性と双方向模倣性について議論する。確率的システムにおいては、この二つの等価性は一致することが知られているが、私達は量子的なシステムにおいてはこれらが一致しないことを示す。三つ目として、余代数様相論理という枠組みを用いて、私達は振る舞い等価性と同等の表現力を持つ様相論理を得る。

Acknowledgements

I would like to express my deepest gratitude to my superviser Ichiro Hasuo whose comments and suggestions were valuable for my study. I also thank the members of Hasuo Laboratory for helpful discussions.

Contents

Chapter 1

Introduction

1.1 Quantum Computation

There have been wide interests in *quantum computation* and *quantum communication*. It is believed that quantum computation has potential to solve some problems more faster than classical computation. For example, Shor's algorithm [30] solves integer factorization problem in polynomial time and Grovers's algorithm [12] solves the problem of searching an unsorted database faster than classical algorithm. In terms of communication, a *quantum cryptography* has a striking advantage over classical one. *Quantum key distribution* realizes that two parties share a random secret key in an "*unconditionally secure*" way. These examples benefit from fundamental quantum principles such as a *no-cloning principle* and *entanglement* of quantum data.

However, traditionally, in order to find new quantum algorithms or protocols or verify the correctness, researchers have to engage in more elemental level approach, for instance the *quantum circuit model* or *quantum Turing machines*. To overcome this difficulty, there have been several high-level formalization for quantum computation. One of these approaches is *quantum programming language with denotational semantics* [15, 29], which provides highly mathematically abstract structure for quantum computation. Another one is *qCCS* [9], which is a variant of *Calculus of Communicating Systems* (*CCS*) with quantum flavor. Originally, CCS is used to model concurrent processes that include communication or message passing, therefore the technique of CCS can be extended to the verification of quantum communication protocols. *Quantum Markov chain* is also a formal model of quantum computation. In [10], a *model checking* approach with a quantum extension of *probabilistic computation tree logic* (PCTL) is adapted to verify the property of quantum systems.

1.2 Coalgebra

A *coalgebra*—a *categorical* dual notion of *algebra*—has been used as a general framework for modeling *state-based transition systems*. This abstract framework covers different types of transition: for example, *non-deterministic*, *probabilistic*, and some sort of *values weighted transition*. By using categorical terms, an F-coalgebra is defined by an arrow $X \to FX$ in a category $\mathcal{C}$ for a functor $F : \mathcal{C} \to \mathcal{C}$. Its simplicity captures very wide and general situations. If we take a powerset $F = \mathcal{P}$, the functions of the form $X \to \mathcal{P}X$ for a set X yield a family of non-deterministic systems. The relationship between two systems $X \to FX$ and $Y \to FY$ is also properly described in this categorical approach.

When we analyze a *non-deterministic automaton*, it is quite often the case

for us to examine the accepted language as a good representation of the system's behavior. For a *probabilistic automaton*, a probabilistic distribution on the accepted language represents the behavior of the systems because each execution of the system is determined by certain probability. There are a coalgebraic theory of *trace semantics* [14] and a coalgebraic description for *forward/backward simulations* [16], which includes the two situations above. The *soundness* theorem saying that "existence of a forward or backward simulation between two systems implies trace inclusion" is proved in an abstract setting [16]. In case of a non-deterministic situation, the inclusion relation of two systems' accepted languages is guaranteed by the existence of a simulation from one to another. In probabilistic settings, the simulation-based approach is used for verification of anonymity as a case study [17].

Whereas coalgebras capture various types of branching flavor, it has been investigated how to get a *modal logic* relevant to the type of systems parametric in a functor [27]. As one of the suitable criteria for logic, *expressivity* is sometimes discussed. A modal logic is expressive means that two states satisfying the same formulas are *behaviorally equivalent*. Saying the statement in contra-position, if two states are not behaviorally equivalent there is a testing formula that is satisfied by one but is not satisfied by the other one. This property is also called Hennessy-Milner property because originally it is proved in the non-deterministic setting by Hennessy and Milner [18]. In [22], the expressivity of modal logic for non-deterministic and probabilistic systems is proved.

1.3 Contributions

There are some works that give a categorical understanding for quantum computation [1, 20]. In the thesis, we present a coalgebraic modeling of quantum systems by using the *quantum branching monad* $\mathcal{Q}$, which is introduced by Hasuo and Hoshino [15] in order to obtain a denotational semantics of higher-order quantum programming language. This monad captures the type of a system's transitions which are weighted by a *quantum operation* like [10] and thus this formulation follows the principle of "quantum data, classical control" in [29]. We define *quantum labeled transition systems* (*QLTS*) and the abstract characterization by coalgebras allows us to apply the general trace and simulation theory to QLTS, too. We show that the simulation-based approach can be utilized to verify the correctness of quantum protocols. Following a probabilistic situation in [17], we also formalize the anonymity of a secret key in the BB84 protocol via QLTS and verify it.

Another contribution of the thesis is that we analyze a difference and a similarity between probabilistic systems and quantum systems in a coalgebraic perspective. A *valuation functor* is introduced in [22] to encompass ordinary *multiset functors* and the *distribution monad*. We find that the valuation functor also includes the quantum branching monad, therefore it enables us to examine the quantum branching monad and the distribution monad in the same framework with a suitable monoid respectively. we define two notions of *bisimilarity* for valuation functors. One is a standard characterization from a coalgebraic view in [3] and the other one is an equivalence class based characterization from a comparison with probabilistic systems [26]. In probabilistic systems, it is known that these two formulation coincides [33], however we show that two notions are different in quantum systems. In categorical terms, this result comes from the *preservation of weak pullback* of the functor, which is also comes from a *refinability* of a monoid (defined in [13]) that instantiates a valuation functor.

Moreover, we investigate a modal logic for quantum systems. We use the result of [22] and offer an expressive modal logic for quantum systems.

1.4 Organization of the Thesis

The thesis is structured as follows. In Chapter 2, we define quantum labeled transition systems (QLTS), and show that the trace semantics and forward/backward simulations for QLTS are canonically obtained. To illustrate this modeling, we describe with it some well-known protocols, for example, the *superdense coding protocol*, the *quantum teleportation protocol*, and the *BB84* protocol. In Chapter 3, two notions of *bisimulation* is defined in terms of valuation functors which generalizes the distribution functor and the quantum branching monad. We show that the two notions do not coincide in the case of the quantum branching monad. In Chapter 4, we recall a categorical setting which is called *dual adjunction* that describes modal logic coalgebraically and the expressive modal logic for the quantum branching monad is given, following [22]. In Chapter 5, we outline the main results in the thesis and point out some future work.

Chapter 2

Trace Semantics for Quantum Branching Monad

Some types of state-based systems are characterized as a *coalgebra* $X \to TFX$ with a *monad* T and a *functor* F. For example, when $T = \mathcal{P}$ (the powerset monad) and $F(X) = 1 + A \times X$, a coalgebra $X \to \mathcal{P}(1 + A \times X)$ gives rise to a non-deterministic automaton or a labeled transition system (LTS). If we use $T = \mathcal{D}$ (the distribution monad) instead of the powerset monad $\mathcal{P}$, a coalgebra $X \to \mathcal{D}(1 + A \times X)$ identifies a probabilistic automaton, in which each transition occurs by probability.

Each execution of an LTS yields a path over the label set. The set of the system's possible paths is called a *trace set*. In the case of a non-deterministic automaton, this trace set is described as the *accepted language*. For a probabilistic automaton, the trace of the system is as *probabilistic distribution* on the accepted language.

In [16], a general theory for the *trace semantics* of these systems is presented. The theory defines a trace semantics of $X \to TFX$ in **Sets** as a unique arrow from $X \to \overline{F}X$ to the final $\overline{F}$-coalgebra in the *Kleisli category* $\mathcal{K}\ell(T)$ for a monad M with a suitable order structure.

2.1 Quantum Branching Monad $\mathcal{Q}$ and Distributive Law

In order to apply these coalgebraic techniques to quantum systems, we first need a monad with a quantum flavor.

Before we recall the definition of monad $\mathcal{Q}$ from [15], we fix some notations for use of quantum theory.

Definition 2.1 (density matrix)**.** An *m-dimensional density matrix* is an matrix $\rho \in \mathbb{C}^{m \times m}$ which is positive and satisfies $\mathrm{tr}(\rho) \in [0, 1]$. We denote the set of all m-dimensional density matrices by $\mathcal{DM}_m$.

Definition 2.2 (quantum operation)**.** A *quantum operation* (QO) from a Hilbert space $\mathbb{C}^m$ to $\mathbb{C}^n$ is a function $\mathcal{E} : \mathcal{DM}_m \to \mathcal{DM}_n$ subject to the following axioms:

1. (Trace non-increasing) For all $\rho \in \mathcal{DM}_m$, wee have $\mathrm{tr}(\mathcal{E}(\rho)) \leq \mathrm{tr}(\rho)$
2. (Linearity) For $\rho_i \in \mathcal{DM}_m$ and $p_i \in [0, 1]$ with $\sum_i p_i \leq 1$, we have $\mathcal{E}(\sum_i p_i \rho_i) = \sum_i p_i \mathcal{E}(\rho_i)$.
3. (Complete positivity) An arbitrary "extension" of $\mathcal{E}$ of the form $\mathcal{I}_k \otimes \mathcal{E}$ maps a positive matrix to a positive one.

We denote the set of QOs from $\mathbb{C}^m$ to $\mathbb{C}^n$ by $\mathcal{QO}_{m,n}$.

There are alternative definitions other than the above one (see e.g. [28]). We sometimes use the *operator-sum representation* of a quantum operation. When a quantum operation $\mathcal{E} : \mathcal{DM}_m \to \mathcal{DM}_n$ satisfies $\mathcal{E}(\rho) = \sum_i E_i \rho (E_i)^\dagger$ for some $n \times m$ matrices (E_i), We use the notation by denoting $\mathcal{E} = \{E_i\}$.

Now, we review the monad $\mathcal{Q}$ [15].

Definition 2.3 (quantum branching monad $\mathcal{Q}$)**.** The *quantum branching monad* $\mathcal{Q} : \mathbf{Sets} \to \mathbf{Sets}$ is defined as follows:

$$\mathcal{Q}X := \{c : X \to \prod_{m,n \in \mathbb{N}} \mathcal{QO}_{m,n} \;\middle|\; \textit{the trace condition } (2.1)\}$$

$$\big(\mathcal{Q}(f)(c)(y)\big)_{m,n} := \sum_{x \in f^{-1}(y)} (c(x))_{m,n}$$

Here we denote the (m,n)-component of $s \in \prod_{m,n} \mathcal{QO}_{m,n}$ by $(r)_{m,n}$. The trace condition is:

$$\forall m. \forall \rho \in \mathcal{DM}_m. \sum_{x \in X} \sum_{n \in \mathbb{N}} (\mathrm{tr}(c(x))_{m,n}(\rho)) \leq 1 \tag{2.1}$$

The unit and the multiplication are:

$$(\eta_X(x)(x'))_{m,n} := \begin{cases} \{\, \mathcal{I}_m \,\} & \text{if } x = x' \text{ and } m = n \\ 0 & \text{otherwise} \end{cases}$$

$$(\mu_X(\phi)(x'))_{m,n} := \sum_{c \in \mathcal{Q}X} \sum_{k \in \mathbb{N}} \Big((c(x))_{k,n} \circ (\phi(c))_{m,k} \Big)$$

where $\mathcal{I}_m$ is a $m \times m$ identity matrix and 0 is a zero operator which maps any density matrix to zero.

In Chapter 3 and 4, we also use the monad $\mathcal{Q}_f$, which is a *finite support* variant of the original monad. The definition of *quantum finite branching monad* $\mathcal{Q}_f$ is slightly different from $\mathcal{Q}$, which requires another condition for $\mathcal{Q}_f X$ in the following way:

$$\mathcal{Q}_f X := \{c : X \to \prod_{m,n \in \mathbb{N}} \mathcal{QO}_{m,n} \;\middle|\; \textit{supp}(c) \text{ is finite and } c \text{ satisfies } (2.1)\} \tag{2.2}$$

where $\textit{supp}(c) = \{x \in X \;\middle|\; c(x) \neq \{0\}\}$.

The intuitive meaning of the coalgebra $X \to \mathcal{Q}Y$ is well explained by a piping analogy in [15]. This intuition is also crucial for defining a quantum system coalgebraically in Section 2.2.

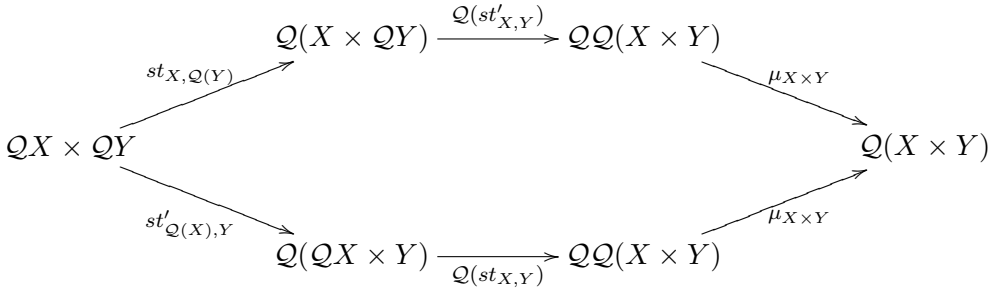
The three monads $\mathcal{L}, \mathcal{P}, \mathcal{D}$ is similarly dealt with in the context of trace semantics [16] and the monad $\mathcal{Q}$ is also characterized the same as $\mathcal{L}, \mathcal{P}, \mathcal{D}$ because the Kleisli category $\mathcal{K}\ell(\mathcal{Q})$ has a suitable order structure [15].

However, we cannot follow the same path as in [15] (for $\mathcal{L}, \mathcal{P}, \mathcal{D}$) because the monad $\mathcal{Q}$ lacks *commutativity* unlike the others. This lack of commutativity comes from that the multiplication of linear operators is not commutative.

Proposition 2.4. *The monad $\mathcal{Q}$ is a* strong monad*, but is not a* commutative monad*.*

Proof. Every monad in **Sets** is strong and it is known that a strength $\mathrm{st}_{X,y} : \mathcal{Q}X \times Y \Rightarrow \mathcal{Q}(X \times Y)$ is uniquely determined. Now we show that the below

diagram does not commute.



The *strength* $\mathrm{st}_{X,Y} : \mathcal{Q}X \times Y \Rightarrow \mathcal{Q}(X \times Y)$ of the monad Q is defined by:

$$\begin{aligned} st_{X,Y} &\colon \mathcal{Q}(X) \times Y \longrightarrow \mathcal{Q}(X \times Y) \\ &(c, y) \longmapsto \lambda(x, y') \in X \times Y. \begin{cases} c(x) & \text{if } y' = y \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

We take $c \in \mathcal{Q}X$ and $d \in \mathcal{Q}Y$, then the two ways from $\mathcal{Q}X \times \mathcal{Q}Y$ to $\mathcal{Q}(X \times Y)$ are calculated as follows:

$$\begin{aligned} (c, d) &\xmapsto{st_{X,\mathcal{Q}(Y)}} \lambda(x, d') \in X \times \mathcal{Q}(Y). \begin{cases} c(x) & \text{if } d' = d \\ 0 & \text{otherwise} \end{cases} \quad (= \xi) \\ &\xmapsto{\mathcal{Q}(st'_{X,Y})} \lambda e \in \mathcal{Q}(X \times Y). \Bigg(\sum_{\substack{(x',d') \in X \times \mathcal{Q}(Y) \\ st'(x',d') = e}} \xi(x', d')_{m,n} \Bigg)_{m,n} \\ &= \lambda e \in \mathcal{Q}(X \times Y). \Bigg(\sum_{\substack{x' \in X \\ st'(x',d) = e}} c(x')_{m,n} \Bigg)_{m,n} \\ &\xmapsto{\mu_{X\times Y}} \lambda(x, y) \in X \times Y. \Bigg(\sum_{e \in \mathcal{Q}(X\times Y)} \sum_{k \in \mathbb{N}} e(x, y)_{k,n} \circ \Big(\sum_{\substack{x' \in X \\ st'(x',d) = e}} c(x')_{m,k} \Big) \Bigg)_{m,n} \\ &= \lambda(x, y) \in X \times Y. \Bigg(\sum_{e \in \mathcal{Q}(X\times Y)} \sum_{k \in \mathbb{N}} \sum_{\substack{x' \in X \\ st'(x',d) = e}} e(x, y)_{k,n} \circ c(x')_{m,k} \Bigg)_{m,n} \\ &= \lambda(x, y) \in X \times Y. \Bigg(\sum_{\substack{e \in \mathcal{Q}(X\times Y) \\ st'(x,d) = e}} \sum_{k \in \mathbb{N}} e(x, y)_{k,n} \circ c(x)_{m,k} \Bigg)_{m,n} \\ &= \lambda(x, y) \in X \times Y. \Bigg(\sum_{k \in \mathbb{N}} d(y)_{k,n} \circ c(x)_{m,k} \Bigg)_{m,n} \quad (= \alpha(c, d)). \end{aligned}$$

Similarly,

$$\begin{aligned} (c, d) &\xmapsto{st'_{\mathcal{Q}(X),Y}} \dots \\ &\xmapsto{st_{X,Y}} \dots \\ &\xmapsto{\mu_{X\times Y}} \lambda(x, y) \in X \times Y. \Bigg(\sum_{k \in \mathbb{N}} c(x)_{k,n} \circ d(y)_{m,k} \Bigg)_{m,n} \quad (= \beta(c, d)). \end{aligned}$$

These two functions are apparently different, as the following counter example

shows:

$$\begin{aligned} X = Y &:= \{*\} \\ (c(*))_{m,n}(\rho) &:= \begin{cases} E(\rho)E^\dagger & \text{if } m = n = 2 \\ 0 & \text{otherwise} \end{cases} \\ (d(*))_{m,n}(\rho) &: = \begin{cases} E'(\rho)E'^\dagger & \text{if } m = n = 2 \\ 0 & \text{otherwise} \end{cases} \\ E &: = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ E' &: = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ \rho &: = |0\rangle\langle 0|. \end{aligned}$$

In this example, E is a projection operator to the basis $|1\rangle$ and E' is a Hadamard operator. Then, two functions $\alpha(c,d), \beta(c,d)$ are not equal:

$$\begin{aligned} \alpha(c,d)(*,*)_{2,2}(\rho) &= EE'|0\rangle\langle 0|E'^\dagger E^\dagger \\ &= \frac{1}{2} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ \beta(c,d)(*,*)_{2,2}(\rho) &= E'E|0\rangle\langle 0|E^\dagger E'^\dagger \\ &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

□

In [16, Lemma 2.4], a sufficient condition for the existence of a *distributive law* between a monad T and a functor F is given. The condition requires T to be a commutative monad and F to be a *shapely functor* [23]. We modify this lemma in order to cover the monad $\mathcal{Q}$ that is not commutative.

Proposition 2.5. *There is a distributive law $\lambda : FT \Rightarrow TF$ for a strong monad $T : \mathbf{Sets} \to \mathbf{Sets}$ and a* linear-polynomial functor $F : \mathbf{Sets} \to \mathbf{Sets}$. *A linear-polynomial functor is defined inductively by the following BNF notation:*

$$F, G, F_i ::= \mathrm{id} \mid A \mid A \times G \mid \coprod_{i \in I} F_i.$$

where A denotes the constant functor into a set A.

Proof. In the proof of [16, Lemma 2.4], a double strength map $\mathrm{dst}_{X,Y} : TX \times TY \to T(X \times Y)$ is needed for the case of $F = F_1 \times F_2$ on the inductive construction of a shapely functor F. However, we only use a strength map $st_{X,Y} : TX \times A \to T(X \times A)$ in order to get a distributive law with linear-polynomial functors.

We only show the case of $F = \Sigma \times F$ of inductive construction of F. See [16, Lemma 2.4] for the other cases. Assume $FX = A \times GX$ and there is a distributive law $\lambda^G : GT \Rightarrow TG$. The distributive law $\lambda^F : FT \Rightarrow TF$ is obtained as a composition:

$$A \times GTX \xrightarrow{1_A \times \lambda^G_X} A \times TGX \xrightarrow{\mathrm{st}'_{A,GX}} T(A \times GX)$$

where $\mathrm{st}'_{X,Y} : X \times FY \to F(X \times Y)$ is defined as $X \times FY \xrightarrow{\langle \pi', \pi \rangle} FY \times X \xrightarrow{\mathrm{st}_{Y,X}} F(Y \times X) \xrightarrow{F(\langle \pi', \pi \rangle)} F(X \times Y)$.
The two arrows $1_A \times \lambda^G_X$ and $\mathrm{st}'_{A,GX}$ are natural in X, and so λ^F_X is. We check that the natural transformation λ^F is compatible with T's monad structure. For the unit,

$$\begin{aligned}
\lambda^F_X \circ F\eta_X &= \mathrm{st}'_{A,GX} \circ (1_A \times \lambda^G_X) \circ F\eta_X \\
&= \mathrm{st}'_{A,GX} \circ (1_A \times \eta_{GX}) \\
&= \eta_{A \times GX} \\
&= \eta_{FX}.
\end{aligned}$$

For the multiplication.

$$\begin{aligned}
\mu_{FX} \circ F\lambda^F_X \circ \lambda^F_{TX} &= \mu_{A \times GX} \circ T(st'_{A,GX}) \circ T(1_A \times \lambda^G_X) \circ st'_{A,GTX} \circ (1_A \times \lambda^G_X) \\
&= \mu_{A \times GX} \circ T(st'_{A,GX}) \circ st'_{A,TGX} \circ (1_A \times T\lambda^G_X) \circ (1_A \times \lambda^G_X) \\
&= st'_{A,X} \circ (1_A \times \mu_{GX}) \circ (1_A \times T\lambda^G_X) \circ (1_A \times \lambda^G_X) \\
&= st'_{A,X} \circ (1_A \times \lambda^G_X) \circ (1_A \times G\mu_X) \\
&= \lambda^F_X \circ F\mu_X.
\end{aligned}$$

□

Every linear-polynomial functor F is isomorphic to the functor $F' = A + B \times \mathrm{id}$ for some fixed sets A, B. That is why we call this family of functors linear-polynomial. The notion of linear-polynomial functor is a restricted one compared to shapely functors, therefore the functor $F = 1 + \mathrm{id} \times \mathrm{id}$ which specifies a signature of a binary tree is not included in it. However, some interesting examples can be written in this formulation as see later.

2.2 Trace Semantics for Quantum LTS

Our first aim is to apply a trace semantics by coinduction [16] to the quantum branching monad $\mathcal{Q}$.

Lemma 2.6. *Let* F : **Sets** $\to$ **Sets** *be a linear-polynomial functor. The* lifting functor $\overline{F} : \mathcal{K}\ell(\mathcal{Q}) \to \mathcal{K}\ell(\mathcal{Q})$ *induced by a distributive law constructed in Proposition 2.5 is* locally continuous*, and thus is* locally monotone*.*

Proof. It is known that there is a bijection correspondence between a lifting functor $\overline{F} : \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ and a distributive law $\lambda : FT \to TF$ for monad T. The construction from one to another is in [14].

We only deal with the case of $F = A \times F$. The other cases of the inductive construction can be proved similarly in [16, Lemma 2.2.8]. We show $\overline{F}(\bigsqcup_i f_i) = \bigsqcup_i (\overline{F} f_i)$ for an ω-chain (f_i) of arrows $X \to \mathcal{Q}(Y)$ in **Sets**. A functor $\overline{F} : \mathcal{K}\ell(\mathcal{Q}) \to \mathcal{K}\ell(\mathcal{Q})$ works on arrows:

$$\begin{aligned}
\overline{F}(X \xrightarrow{f} \mathcal{Q}(Y)) &= (FX \xrightarrow{Ff} F\mathcal{Q}Y \xrightarrow{\lambda^F_Y} \mathcal{Q}FY) \\
&= (A \times GX \xrightarrow{1_A \times Gf} A \times G\mathcal{Q}Y \xrightarrow{1_A \times \lambda^G_Y} A \times \mathcal{Q}GY \xrightarrow{\mathrm{st}'_{A,GY}} \mathcal{Q}(A \times GY)) \\
&= (A \times GX \xrightarrow{1_A \times \overline{G}f} A \times \mathcal{Q}GY \xrightarrow{\mathrm{st}'_{A,GY}} \mathcal{Q}(A \times GY))
\end{aligned}$$

where $f : X \to Y$ in $\mathcal{K}\ell(\mathcal{Q})$.
First, we show $\mathrm{st}'_{A,X} : A \times \mathcal{Q}(X) \to \mathcal{Q}(A \times X)$ is continuous, that is, for an ω-chain $(u_i) \in \mathcal{Q}(X)^{\mathbb{N}}$ and $a \in A$,

$$\mathrm{st}'_{A,X}(a, \bigsqcup_i u_i) = \bigsqcup_i \mathrm{st}'_{A,X}(a, u_i).$$

If $a' = a$,

$$\begin{aligned}
\mathrm{st}'_{A,X}(a, \bigsqcup_i u_i)(a', x) &= (\bigsqcup_i u_i)(x) \\
&= \bigsqcup_i (u_i(x)) \\
&= \bigsqcup_i (\mathrm{st}'_{A,X}(a, u_i)(a', x)) \\
&= \bigsqcup_i (\mathrm{st}'_{A,X}(a, u_i))(a', x).
\end{aligned}$$

If $a' \neq a$,

$$\begin{aligned}
\mathrm{st}'_{A,X}(a, \bigsqcup_i u_i)(a', x) &= 0 \\
&= \bigsqcup_i (\mathrm{st}'_{A,X}(a, u_i))(a', x).
\end{aligned}$$

And now, for $(a, s) \in A \times GX$,

$$\begin{aligned}
(\overline{F}(\bigsqcup_i f_i))(a, s) &= (\mathrm{st}'_{A,GY} \circ (1_A \times \overline{G}(\bigsqcup_i f_i)))(a, s) \\
&= \mathrm{st}'_{A,GY}(a, \overline{G}(\bigsqcup_i f_i)(s)) \\
&= \mathrm{st}'_{A,GY}(a, (\bigsqcup_i (\overline{G}(f_i)))(s)) && \text{by the inductive hypothesis} \\
&= \mathrm{st}'_{A,GY}(a, \bigsqcup_i (\overline{G}(f_i)(s))) && \text{by the pointwise order} \\
&= \bigsqcup_i \mathrm{st}'_{A,GY}(a, \overline{G}(f_i)(s)) && \text{by the continuity of st} \\
&= \bigsqcup_i (\mathrm{st}'_{A,GY} \circ (1_A \times \overline{G} f_i))(a, s) \\
&= (\bigsqcup_i (\mathrm{st}'_{A,GY} \circ (1_A \times \overline{G} f_i)))(a, s) && \text{by the pointwise order} \\
&= (\bigsqcup_i (\overline{F} f_i))(a, s)
\end{aligned}$$

□

Proposition 2.7 (trace situation for $\mathcal{Q}$)**.** *The monad $\mathcal{Q}$, a linear-polynomial functor F, and a distributive law $\lambda : F\mathcal{Q} \Rightarrow \mathcal{Q}F$ constructed in Proposition 2.5 satisfies the following three conditions; and therefore by [16, Theorem 3.3] the* initial F-algebra $\alpha : FA \xrightarrow{\cong} A$ *in* **Sets** *gives rise to a* final $\overline{F}$-coalgebra $\eta_{FA} \circ \alpha^{-1} : A \to \overline{F}A$ *in* $\mathcal{K}\ell(Q)$.

1. *The Kleisli category* $\mathcal{K}\ell(\mathcal{Q})$ *is* **Cppo**-enriched *and composition in* $\mathcal{K}\ell(\mathcal{Q})$ *is* left-strict.

2. *The lifting functor* $\overline{F} : \mathcal{K}\ell(\mathcal{Q}) \to \mathcal{K}\ell(\mathcal{Q})$ *induced by the distributive law* λ *is locally monotone.*

3. *The functor* F *preserves* ω*-colimits in* **Sets**.

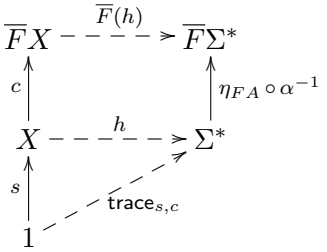
Proof. (1) is shown in [15]. (2) is from Lemma 2.6. (3) is established because any shapely functor is so [14]. □

Now, we introduce the notion of *quantum labeled transition system* (QLTS), which is given as an instance of (T, F)-systems in [14] for the choice of parameters $T = \mathcal{Q}$ and $F = 1 + \Sigma \times \mathrm{id}$. QLTS is corresponds to LTS or probabilistic LTS (PLTS) at a categorical level of abstraction.

Definition 2.8 (quantum LTS and its trace semantics)**.** A QLTS (X, s, c) is specified by a set X and a pair of functions $s : 1 \to \mathcal{Q}X$ and $c : X \to \mathcal{Q}(1+\Sigma\times X)$, which is also a pair of arrows in $\mathcal{K}\ell(\mathcal{Q})$:

$$1 \xrightarrow{s} X \xrightarrow{c} 1 + \Sigma \times X.$$

The *trace semantics* of a QLTS (X, s, c) is defined by the composed arrow $\mathsf{trace}_{s,c}$ in the diagram:



in $\mathcal{K}\ell(\mathcal{Q})$, $F = 1 + \Sigma \times \mathrm{id}$

Here $\alpha : 1 + \Sigma \times \Sigma^* \to \Sigma^*$ is the initial algebra in **Sets** with the carrier set Σ^* of finite words on Σ and the arrow $h : X \to A$ is the unique homomorphism induced by the final $\overline{F}$-coalgebra. This coincidence of a initial algebra and a final coalgebra is just explained in Proposition 2.7.

To give an illustrative explanation, we follow one execution of a QLTS. Here we use the word "state" for two distinct meaning. One indicates a "quantum state" as a density matrix. The other one indicates a state in a "state space" of the system. This contrast of two "state" is just the principle "quantum data, classical control" in [29].

One execution of a QLTS is as follows. First we prepare any initial quantum state $\rho \in \mathcal{DM}_m$. The quantum state ρ is taken into some state $x \in X$ of the system and evolves into $\rho' = (s(x))_{m,n}(\rho) \in \mathcal{DM}_n$. Then, in each step of the transition between states $x, x' \in X$, some action $a \in \Sigma$ occurs and the quantum states evolves into $(c(x)(a, x'))_{n,l}(\rho') \in \mathcal{DM}_l$. After iteration of such transitions, finally the system terminates and the last quantum state is obtained as $\rho'' = (c(x')(\checkmark))_{l,k}(\rho') \in \mathcal{DM}_k$.

As each transition of a probabilistic system e.g. Markov chain is determined by probabilistic distribution, each step of a QLTS is determined by "quantum distribution" which is from the trace condition (2.1).

In Definition 2.8, trace semantics is described abstractly as an arrow $1 \to \Sigma^*$ in $\mathcal{K}\ell(\mathcal{Q})$. However, this definition yields a concrete one which is a feasible notion for QLTS. From now, we identify a function $1 \to X$ as an element of X. The trace

semantics is calculated recursively (corecursively) as follows. First the unique homomorphism $h : X \to \Sigma^*$ is here: for $\sigma \in \Sigma^*$,

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

Then, the trace semantics is obtained:

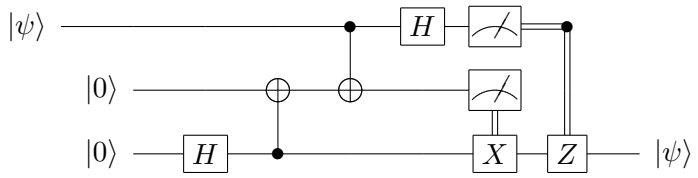
$$(\mathsf{trace}_{s,c}(\sigma))_{m,n} = \sum_{x \in X} \sum_{k \in \mathbb{N}} (h(x)(\sigma))_{k,n} \circ (s(x))_{m,k}.$$

This operator $(\mathsf{trace}_{s,c}(\sigma))_{m,n}$ can be seen as an accumulated quantum operation along paths that cause a sequence of actions σ through the system. Therefore, the probability of observing the actions $\sigma \in \Sigma^*$ with an initial state $\rho \in \mathcal{DM}_m$ is also represented as the *trace value*

$$\sum_{n \in \mathbb{N}} \mathrm{tr}\Big(\big(\mathsf{trace}_{s,c}(\sigma)\big)_{m,n}(\rho) \Big) \in [0, 1].$$

To further illustrate this modeling of QLTS and its trace semantics, we present some examples.

Example 2.9 (quantum teleportation protocol)**.** The first example is the *quantum teleportation protocol*. We show two different representation of the quantum teleportation protocol; one is by the *quantum circuits*:



and the other one is by the QLTS (when $\Sigma = \{\tau\}$) with $s : \mathcal{Q}X$ and $c : X \to \mathcal{Q}(1 + X)$:

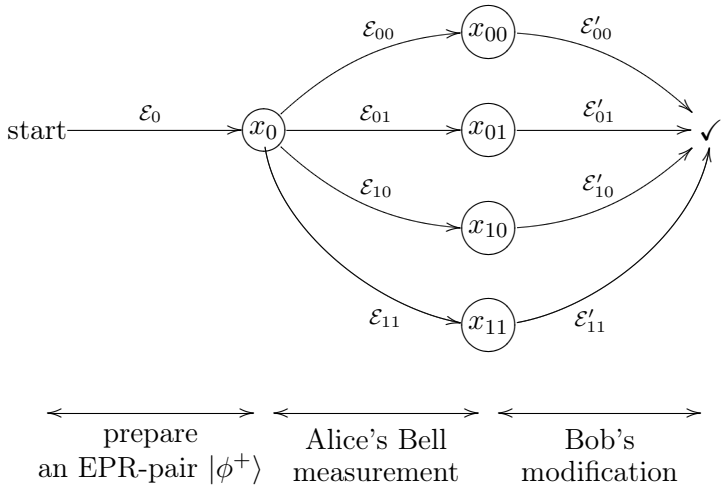
$$(s(x_o))_{2,8} = \lambda\rho \in \mathcal{DM}_2.\, \rho \otimes \left|\phi^+\right\rangle\!\left\langle\phi^+\right| \quad (= \mathcal{E}_0)$$

$$(c(x_0)(x_i))_{8,2} = \begin{cases} \{\, (\langle\phi^+| \otimes \mathcal{I}_2)\, \} & (= \mathcal{E}_{00}) \quad \text{if } i = 00 \\ \{\, (\langle\phi^-| \otimes \mathcal{I}_2)\, \} & (= \mathcal{E}_{01}) \quad \text{if } i = 01 \\ \{\, (\langle\psi^+| \otimes \mathcal{I}_2)\, \} & (= \mathcal{E}_{10}) \quad \text{if } i = 10 \\ \{\, (\langle\psi^-| \otimes \mathcal{I}_2)\, \} & (= \mathcal{E}_{11}) \quad \text{if } i = 11 \end{cases}$$

$$(c(x_i)(\checkmark))_{2,2}(\rho) = \begin{cases} \{\, \mathcal{I}_2 \,\} & (= \mathcal{E}'_{00}) \quad \text{if } i = 00 \\ \{\, X \,\} & (= \mathcal{E}'_{01}) \quad \text{if } i = 01 \\ \{\, Z \,\} & (= \mathcal{E}'_{10}) \quad \text{if } i = 10 \\ \{\, ZX \,\} & (= \mathcal{E}'_{11}) \quad \text{if } i = 11 \end{cases}$$

where H is a *Hadamard matrix*, X, Z are respectively the *Pauli-X matrix* and the *Pauli-Z matrix*, and $|\phi^+\rangle, |\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle$ are *Bell states*. Here some omitted part of the function definition means that a zero operator is assigned for it, for example $c(x_0)(x_1)_{m,n} = 0$ if $m \neq 8$ or $m \neq 8$. The QLTS is also depicted informally as an

automaton with *explicit termination*:



This automaton-like representation may be very intuitive for the reader.

Then, we obtain the trace semantics of the QLTS calculated as below:

$$\begin{aligned}(\mathsf{trace}_{s,c}(\tau^2))_{2,2} &= \left(\sum_{i\in\{00,01,10,11\}} \mathcal{E}'_i \circ \mathcal{E}_i\right) \circ \mathcal{E}_1 \circ \mathcal{E}_o \\ &= \{\, \mathcal{I}_2 \,\}\end{aligned}$$

This result of the identity operation means that for any qubit Alice starts with, Bob finally recovers the original qubit after the execution of this protocol.

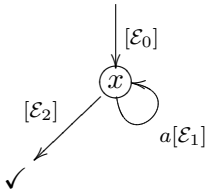
Example 2.10 (quantum loop programs)**.** We show another example of QLTS for describing *quantum programs with output*. Here we use similar example like [10], in which a qMC is given to interpret the meaning of a simple quantum loop program without output. A quantum program is here:

- **function** (q_0 : 1-qubit)
- **while** $\mathrm{Meas}_{|0\rangle,|1\rangle}(q_0) == |0\rangle$ **do**
- output a; $q_0 := H(q_0)$;
- **end**
- **return** q_0;

This program (function) means as follows: first any qubit is given as q_o; a *projective measurement* in the bases $|0\rangle, |1\rangle$ is applied to q_o; if the outcome of the measurement is $|1\rangle$, then this function returns a quantum bit q_o; otherwise it prints a symbol a and the Hadamard operator is applied to q_o and another while iteration continues. The QLTS that describes the behavior or the program is constructed with $s : 1 \to \mathcal{Q}X$ and $c : X \to \mathcal{Q}(1 + \{a\} \times X)$:

$$\begin{aligned}(s(x))_{2,2} &= \{\, \mathcal{I}_2 \,\} \quad (= \mathcal{E}_0) \\ (c(x)(a,x))_{2,2} &= \{\, HP_{|0\rangle} \,\} \quad (= \mathcal{E}_1) \\ (c(x)(\checkmark))_{2,2} &= \{\, P_{|1\rangle} \,\} \quad (= \mathcal{E}_2)\end{aligned}$$

where $P_{\langle i|}$ is a *projection matrix* of $\langle i|$. We show the QLTS as an automaton:



Then, the trace semantics of the QLTS is obtained:

$$((\mathsf{trace}_{s,c})(a^n))_{2,2} = \mathcal{E}_2 \circ \mathcal{E}_1^n \circ \mathcal{E}_0.$$

If we start with a qubit $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and the program outputs "$\underbrace{a \dots a}_{n}$", then the last quantum state is calculated as follows:

$$\begin{aligned} \rho &= \big((\mathsf{trace}_{s,c})(a^n)\big)_{2,2}(|\psi\rangle\langle\psi|) \\ &= |\phi\rangle\langle\phi| \end{aligned} \qquad \text{with } |\phi\rangle = \left(\frac{1}{\sqrt{2}}\right)^n \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Therefore, we observe the words "$\underbrace{a \dots a}_{n}$" by the probability $\mathrm{tr}(|\phi\rangle\langle\phi|) = \left(\frac{1}{2}\right)^n$.

2.3 Simulation for Quantum LTS

We follow [14] and further examine QLTS in terms of a *simulation* for two systems. we write down concretely a *forward/backward simulation* for QLTS.

Definition 2.11 (simulation for QLTS)**.** Let (X, s, d) and (Y, t, d) be two QLTS. A *forward simulation* from (Y, t, d) to (X, s, c) is a function $f : X \to \mathcal{Q}(Y)$ satisfies:

$$\begin{aligned} \sum_{x\in X}\sum_{k\in\mathbb{N}} (f(x)(y))_{k,n} \circ (s(x))_{m,k} &\sqsupseteq (t(y))_{m,n} \\ \sum_{x'\in X}\sum_{k\in\mathbb{N}} (f(x')(y))_{k,n} \circ (c(x)(a,x'))_{m,k} &\sqsupseteq \sum_{y'\in Y}\sum_{k\in\mathbb{N}} (d(y')(a,y))_{k,n} \circ f(x)(y')_{m,k} \\ (c(x)(\checkmark))_{m,n} &\sqsupseteq \sum_{y\in Y}\sum_{k\in\mathbb{N}} (d(y)(\checkmark))_{k,n} \circ (f(x)(y))_{m,k} \end{aligned}$$

where $a \in \Sigma$. A *backward simulation* from (X, s, c) to (Y, t, d) is a function $f : X \to \mathcal{Q}(Y)$ satisfies:

$$\begin{aligned} \sum_{x\in X}\sum_{k\in\mathbb{N}} (f(x)(y))_{k,n} \circ (s(x))_{m,k} &\sqsubseteq (t(y))_{m,n} \\ \sum_{x'\in X}\sum_{k\in\mathbb{N}} (f(x')(y))_{k,n} \circ (c(x)(a,x'))_{m,k} &\sqsubseteq \sum_{y'\in Y}\sum_{k\in\mathbb{N}} (d(y')(a,y))_{k,n} \circ f(x)(y')_{m,k} \\ (c(x)(\checkmark))_{m,n} &\sqsubseteq \sum_{y\in Y}\sum_{k\in\mathbb{N}} (d(y)(\checkmark))_{k,n} \circ (f(x)(y))_{m,k} \end{aligned}$$

where $a \in \Sigma$. We denote $(X, s, c) \sqsubseteq_{\mathbf{F}} (Y, t, d)$ ($(X, s, c) \sqsubseteq_{\mathbf{B}} (Y, t, d)$) if there is a forward simulation (backward simulation) from (s, c) to (t, d).

The existence of simulation—forward or backward—between two systems guarantees that the behavior of one system is simulated by another one.

Proposition 2.12 (simulation is a sound condition for trace inclusion)**.** *Let* $1 \xrightarrow{s} X \xrightarrow{c} \overline{F}X$ *and* $1 \xrightarrow{t} Y \xrightarrow{d} \overline{F}Y$ *be two QLTSs with* $F = 1 + \Sigma \times X$*.*

1. $(s, c) \sqsubseteq_{\mathbf{F}} (t, d) \implies \mathsf{trace}_{s,c} \sqsubseteq \mathsf{trace}_{t,d}$
2. $(s, c) \sqsubseteq_{\mathbf{B}} (t, d) \implies \mathsf{trace}_{s,c} \sqsubseteq \mathsf{trace}_{t,d}$

Proof. See [14, Theorem 6.1]. □

At the end of this chapter, we show two protocols—superdense coding protocol and quantum teleportation protocol—and prove the correctness of the protocols by giving simulations with *specification systems*.

Example 2.13 (superdense coding protocol)**.** We first model the superdense coding protocol in the QLTS framework. The label set $\Sigma = \{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}$ is a set of messages that Alice sends and Bob receives. We assume that Alice's choice of a message is dependent on the probabilities $\{p_1, p_2, p_3, p_4\}$; Alice sends a_i by the probability p_i. The superdense coding protocol is described by the QLTS (X, s, c):

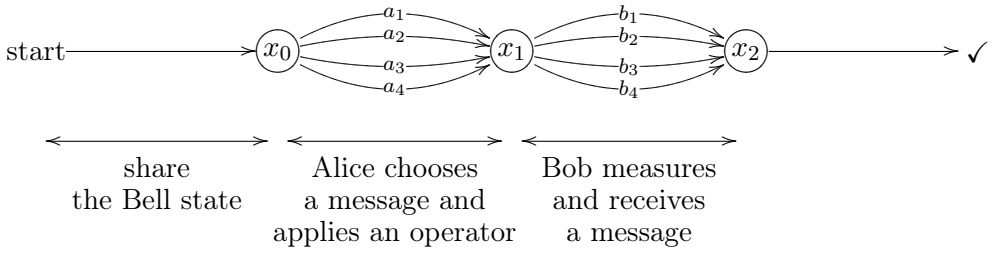
$$(s(x_0))_{1,4} = \{\, |\phi^+\rangle \,\}$$

$$(c(x_0)(a_i, x_1))_{4,4} = \begin{cases} \{\, \sqrt{p_1}\, \mathcal{I}_4 \,\} & \text{if } i = 1 \\ \{\, \sqrt{p_2}\, (Z \otimes \mathcal{I}_2) \,\} & \text{if } i = 2 \\ \{\, \sqrt{p_3}\, (X \otimes \mathcal{I}_2) \,\} & \text{if } i = 3 \\ \{\, \sqrt{p_4}\, (iY \otimes \mathcal{I}_2) \,\} & \text{if } i = 4 \end{cases}$$

$$(c(x_1)(b_i, x_2))_{4,1}(\rho) = \begin{cases} \{\, \langle\phi^+| \,\} & \text{if } i = 1 \\ \{\, \langle\phi^-| \,\} & \text{if } i = 2 \\ \{\, \langle\psi^+| \,\} & \text{if } i = 3 \\ \{\, \langle\psi^-| \,\} & \text{if } i = 4 \end{cases}$$

$$(c(x_2)(\checkmark))_{1,1} = \{\, \mathcal{I}_1 \,\}$$

where Y is the *Pauli-Y matrix*. The automaton-like representation is here:



The trace semantics of the QLTS is obtained:

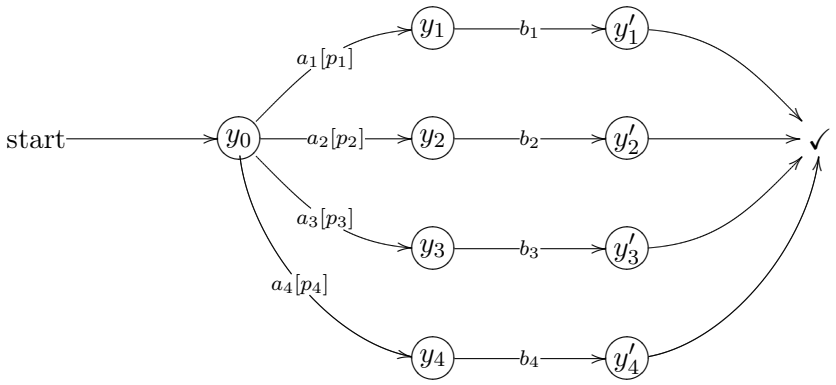
$$(\mathsf{trace}_{s,c}(a_i \cdot b_j))_{1,1} = \begin{cases} \{\, \sqrt{p_i}\, \mathcal{I}_1 \,\} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

This means that Bob receive the Alice's message whatever message Alice wants to send.

Next, we give a specification system suitable to the superdense protocol. A specification is given as a QLTS (Y, t, d):

$$(t(y_0))_{1,1} = \{\, \mathcal{I}_1 \,\}$$
$$(d(y_0)(a_i, y_i))_{1,1} = \{\, \sqrt{p_i}\, \mathcal{I}_1 \,\}$$
$$(d(y_i)(b_i, y_i'))_{1,1} = \{\, \mathcal{I}_1 \,\}$$
$$(d(y_i')(\checkmark))_{1,1} = \{\, \mathcal{I}_1 \,\}$$

This QLTS can be described as a simple probabilistic automaton:



There is a function $f : Y \to \mathcal{Q}X$ which makes $(X, s, c) \sqsubseteq_{\mathbf{F}} (Y, t, d)$ and $(Y, t, d) \sqsubseteq_{\mathbf{B}} (X, s, c)$:

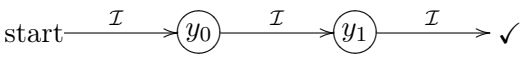
$$
\begin{aligned}
(f(y_0)(x_0))_{1,4} &= \{\, |\phi^+\rangle \,\} \\
(f(y_i)(x_1))_{1,4} &= \begin{cases} \{\, \sqrt{p_1}\, \mathcal{I}_4 |\phi^+\rangle \,\} & \text{if } i = 1 \\ \{\, \sqrt{p_2}\, (Z \otimes \mathcal{I}_2) |\phi^+\rangle \,\} & \text{if } i = 2 \\ \{\, \sqrt{p_3}\, (X \otimes \mathcal{I}_2) |\phi^+\rangle \,\} & \text{if } i = 3 \\ \{\, \sqrt{p_4}\, (iY \otimes \mathcal{I}_2) |\phi^+\rangle \,\} & \text{if } i = 4 \end{cases} \\
(f(y'_i)(x_2))_{1,1} &= \{\, \mathcal{I}_1 \,\}
\end{aligned}
$$

Therefore, we establish that the trace semantics of two systems—the protocol and its specification—coincides, i.e. $\mathsf{trace}_{s,c} = \mathsf{trace}_{t,d}$.

Example 2.14 (quantum teleportation protocol revisited)**.** We recall Example 2.9 and find a simulation between a specification systems. A specification is given as a QLTS (Y, t, d):

$$
\begin{aligned}
(t(y_0))_{2,2} &= \{\, \mathcal{I}_2 \,\} \\
(d(y_0)(y_1))_{2,2} &= \{\, \mathcal{I}_2 \,\} \\
(d(y_1)(\checkmark))_{2,2} &= \{\, \mathcal{I}_2 \,\}
\end{aligned}
$$

the automaton-like representation is so simple:



Then, we find a simulation $f : Y \to \mathcal{Q}X$:

$$
\begin{aligned}
(f(y_0)(x_0))_{2,8} &= \rho \in \mathcal{DM}_2.\, \rho \otimes |\phi^+\rangle\langle\phi^+| \quad (= \mathcal{F}) \\
(f(y_1)(x_i))_{2,2} &= \begin{cases} \{\, (\langle\phi^+| \otimes \mathcal{I}_2) \,\} \circ \mathcal{F} & \text{if } i = 00 \\ \{\, (\langle\phi^-| \otimes \mathcal{I}_2) \,\} \circ \mathcal{F} & \text{if } i = 01 \\ \{\, (\langle\psi^+| \otimes \mathcal{I}_2) \,\} \circ \mathcal{F} & \text{if } i = 10 \\ \{\, (\langle\psi^-| \otimes \mathcal{I}_2) \,\} \circ \mathcal{F} & \text{if } i = 11 \end{cases}
\end{aligned}
$$

This makes $(X, s, c) \sqsubseteq_{\mathbf{F}} (Y, t, d)$ and $(Y, t, d) \sqsubseteq_{\mathbf{B}} (X, s, c)$, therefore the trace equivalence of two systems is proved.

2.4 Anonymity Analysis via QLTS

In this section, we analyze the probabilistic anonymity [7,17] of a key distribution in the *BB84* protocol [6]. In [17], it is shown that the Dining Cryptographers protocol satisfies probabilistic anonymity and the Crowds protocol satisfies the probable innocense—weaker notion of probabilistic anonymity—by searching a simulation with a specification. Here, however, the simulation based verification technique is not applied to the BB84 protocol. We only induce the idea of probabilistic anonymity to the BB84 protocol and calculate them by hand. As future work, we aim to verify some quantum protocols with simulation based approach because this approach might be more scalable.

The BB84 protocol serves as a secure way of sharing a secret key between two parties (Alice and Bob). We follow the simple procedure of the protocol in [10]. The protocol goes in the following way:

1. Alice generates two random n-bit strings $K_1^A \dots K_n^A$ and $B_1^A \dots B_n^A$.
2. Alice prepares an n-qubit string $Q_1 \dots Q_n$, according to the two strings previously generated, like $Q_i = H\big|K_i^A\big\rangle$.
3. Alice sends qubits $Q_1 \dots Q_n$ to Bob via the quantum channel.
4. Bob generates a random n-bit string $B_1^B \dots B_n^B$.
5. Bob measures each qubit Q_i received from Alice by $\{|0\rangle, |1\rangle\}$ if $B_i^B = 0$ or by $\{|+\rangle, |-\rangle\}$ if $B_i^B = 1$ and let the results of the measurements be an n-bit string $K_1^B \dots K_n^B$.
6. Bob announces his choice of measurement bases $B_1^B \dots B_n^B$ to Alice via the classical channel, and then Alice also announces her bases $B_1^A \dots B_n^A$ to Bob via the classical channel.
7. Alice and Bob discard the bits in $K_1^A \dots K_n^A$ and $K_1^B \dots K_n^B$ where the corresponding bits of bases are not equal, that is, $B_i^A \neq B_i^B$ and they share the remaining bits as a secret key.

In this procedure, there is an observer (Eve) of classical messages through the classical channel. For simplicity we assume that the observer do not touch the quantum channel and no noise occurs in the quantum channel. To deal with more practical situations is our future work.

The behavior of the protocol is determined by the probability. Here we give an informal definition of probabilistic anonymity.

Definition 2.15 (anonymity)**.** The protocol satisfies *probabilistic anonymity* if for any observation o and for any keys k, k',

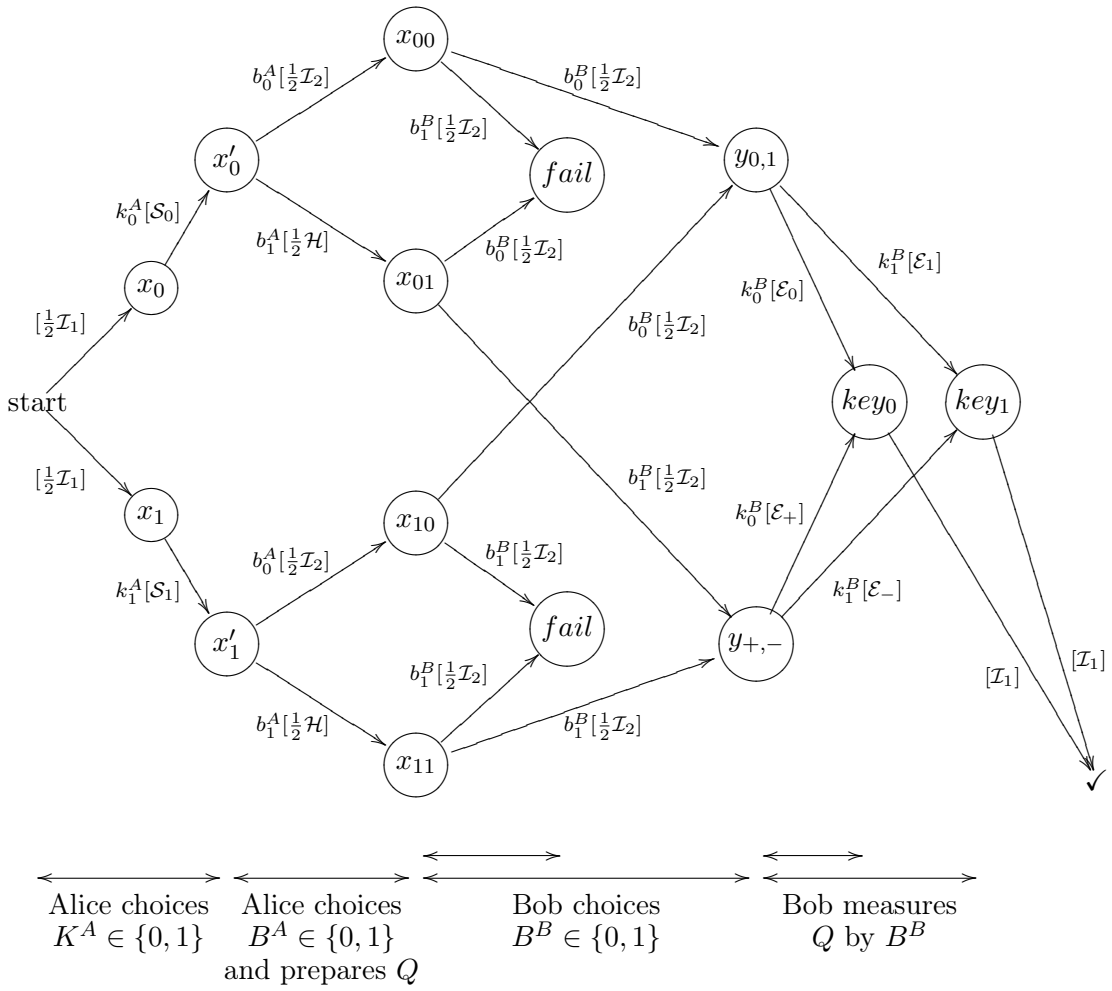
$$Pr(\, k \mid o \,) = Pr(\, k' \mid o \,)$$

Here $Pr(\, k \mid o \,)$ denotes the conditional probability for the event "given that Alice and Bob communicate classical messages o, they share the key k at the last of the protocol."

This definition of anonymity represents that there is no leaking of the information about the key shared by Alice and Bob, therefore Eve cannot determine the key from the classical messages Alice and Bob communicate.

First we present a QLTS which describes the BB84 protocol in the simplest case of $n = 1$, and next we formalize the anonymity of the secret key via the QLTS. Finally the anonymity of the protocol is verified.

Now, the QLTS of the BB84 protocol is given:



where $\mathcal{I}_i$ is a identity operator from $\mathcal{DM}_i$ to $\mathcal{DM}_i$, $\mathcal{H} = \{ H \}$ (a Hadamard operation), $\mathcal{S}_i = \{ |i\rangle \}$ (a preparation operation), and $\mathcal{E}_i = \{ \langle i| \}$ (a measurement operation). The zero operator is assigned for the other component omitted in the edges of the diagram.

The actions Σ in the QLTS are defined as follows:

$$\Sigma := Sec + Obs$$
$$Sec := \{k_0^A, k_1^A, k_0^B, k_1^B\}$$
$$Obs := \{b_0^A, b_1^A, b_0^B, b_1^B\}$$

The two superscripts A, B stand for Alice and Bob, respectively. Each $k_i^X \in Sec$ indicates that $X \in \{A, B\}$ intends to share a key $i \in \{0, 1\}$, and each $b_i^X \in Obs$ stands for a basis $i \in \{0, 1\}$ chosen by $X \in \{A, B\}$. In this formalization, we assume that Eve can observe only $Obs = \{b_0^A, b_1^A, b_0^B, b_1^B\}$ of actions.

Each state of the QLTS has a concrete meaning as annotated by $\overleftrightarrow{\ldots}$ at the bottom of the QLTS. The state $x_{i,j}$ specifies the step of the protocol in which Alice randomly generates $K^A = i$ and $B^A = j$. The state $fail$ means that if Alice's basis B^A and Bob's basis B^B are not equal then the protocol unsuccessfully

terminates. The states $y_{0,1}, y_{+,-}$ are the steps in which Bob is ready to measure a qubit by bases $\{|0\rangle, |1\rangle\}$ and $\{|+\rangle, |-\rangle\}$, respectively. The states key_o, key_1 means that the protocol successfully terminates with a secret key 0 and 1, respectively.

We obtain the probabilistic behavior of the protocol from the trace semantics of the QLTS, which is a probability distribution on the sequence of actions. We denote it by $Pr : \Sigma^* \to [0,1]$: for $\sigma \in \Sigma^*$, $Pr(\sigma) = (\mathsf{trace}(\sigma))_{1,1}(1)$.

The following notations for predicates on Σ^* are defined: for $i, j \in \{0,1\}$,

$$K_{i,j} = \{ \dots k_i^A \dots k_j^B \dots \ \in \Sigma^* \}$$
$$O_{i,j} = \{ \dots b_i^A \dots b_j^B \dots \ \in \Sigma^* \},$$

Each predicate on Σ^* specifies a certain set of execution paths through the protocol. For example, the predicate $K_{0,0}$ is a set of a protocol execution in which Alice and Bob accomplish to share the secret key 0. The predicate $O_{0,1}$ is a set of a protocol execution in which Eve observes the announcements of bases, Alice's 0 and Bob's 1. We also denote the probability of the predicate $P \subseteq \Sigma^*$ by $Pr(P) = \sum_{\sigma \in P} Pr(\sigma)$.

The correctness of the protocol—Alice and Bob share the same bit—is described by the two conditions:

$$Pr(K_{0,1}) = 0 \text{ and } Pr(K_{1,0}) = 0. \tag{2.3}$$

Moreover, the anonymity of a secret key in the BB84 protocol is formally defined, which is induced from Definition 2.15.

$$Pr(K_{0,0} \mid O_{i,j}) = Pr(K_{1,1} \mid O_{i,j}), \quad \text{for all } i, j \in \{0,1\}. \tag{2.4}$$

Here the conditional probability $Pr(P_1 \mid P_2)$ represents $Pr(P_1 \cap P_2)/Pr(P_2)$ for two predicates P_1, P_2.

Eventually, the probability distribution Pr is calculated as follows:

$$Pr(k_0^A \cdot b_0^A \cdot b_0^B \cdot k_0^B) = \left(\mathcal{I}_1 \circ \mathcal{E}_0 \circ \frac{1}{2}\mathcal{I}_2 \circ \frac{1}{2}\mathcal{I}_2 \circ \mathcal{S}_0 \circ \frac{1}{2}\mathcal{I}_1 \right)(1) = 1/8,$$
$$Pr(k_0^A \cdot b_0^A \cdot b_0^B \cdot k_1^B) = \left(\mathcal{I}_1 \circ \mathcal{E}_1 \circ \frac{1}{2}\mathcal{I}_2 \circ \frac{1}{2}\mathcal{I}_2 \circ \mathcal{S}_0 \circ \frac{1}{2}\mathcal{I}_1 \right)(1) = 0,$$
$$Pr(k_0^A \cdot b_1^A \cdot b_1^B \cdot k_0^B) = \left(\mathcal{I}_1 \circ \mathcal{E}_+ \circ \frac{1}{2}\mathcal{I}_2 \circ \frac{1}{2}\mathcal{H} \circ \mathcal{S}_0 \circ \frac{1}{2}\mathcal{I}_1 \right)(1) = 1/8,$$
$$Pr(k_0^A \cdot b_1^A \cdot b_1^B \cdot k_1^B) = \left(\mathcal{I}_1 \circ \mathcal{E}_- \circ \frac{1}{2}\mathcal{I}_2 \circ \frac{1}{2}\mathcal{H} \circ \mathcal{S}_0 \circ \frac{1}{2}\mathcal{I}_1 \right)(1) = 0,$$
$$Pr(k_1^A \cdot b_0^A \cdot b_0^B \cdot k_0^B) = \left(\mathcal{I}_1 \circ \mathcal{E}_0 \circ \frac{1}{2}\mathcal{I}_2 \circ \frac{1}{2}\mathcal{I}_2 \circ \mathcal{S}_1 \circ \frac{1}{2}\mathcal{I}_1 \right)(1) = 0,$$
$$Pr(k_1^A \cdot b_0^A \cdot b_0^B \cdot k_1^B) = \left(\mathcal{I}_1 \circ \mathcal{E}_1 \circ \frac{1}{2}\mathcal{I}_2 \circ \frac{1}{2}\mathcal{I}_2 \circ \mathcal{S}_1 \circ \frac{1}{2}\mathcal{I}_1 \right)(1) = 1/8,$$
$$Pr(k_1^A \cdot b_1^A \cdot b_1^B \cdot k_0^B) = \left(\mathcal{I}_1 \circ \mathcal{E}_+ \circ \frac{1}{2}\mathcal{I}_2 \circ \frac{1}{2}\mathcal{H} \circ \mathcal{S}_1 \circ \frac{1}{2}\mathcal{I}_1 \right)(1) = 0,$$
$$Pr(k_1^A \cdot b_1^A \cdot b_1^B \cdot k_1^B) = \left(\mathcal{I}_1 \circ \mathcal{E}_- \circ \frac{1}{2}\mathcal{I}_2 \circ \frac{1}{2}\mathcal{H} \circ \mathcal{S}_1 \circ \frac{1}{2}\mathcal{I}_1 \right)(1) = 1/8.$$

For other $\sigma \in \Sigma^*$, the probability 0 is assigned.

Therefore, the two properties—correctness (2.3) and anonymity (2.4)—of the BB84 protocol is verified. As for the correctness,

$$Pr(K_{0,1}) = Pr(K_{1,0}) = 0.$$

As for the anonymity,

$$
\begin{aligned}
Pr(K_{0,0} \mid O_{0,0}) &= Pr(K_{1,1} \mid O_{0,0}) = 1/8, \\
Pr(K_{0,0} \mid O_{0,1}) &= Pr(K_{1,1} \mid O_{0,1}) = 0, \\
Pr(K_{0,0} \mid O_{1,0}) &= Pr(K_{1,1} \mid O_{1,0}) = 0, \\
Pr(K_{0,0} \mid O_{1,1}) &= Pr(K_{1,1} \mid O_{1,1}) = 1/8.
\end{aligned}
$$

Chapter 3

Bisimilarity and Behavioral Equivalence

We devoted the previous chapter to QLTS as a model of quantum systems, which is a *generative system* with explicit termination. From now, in order to examine more fundamental aspects of this modeling, we deal with a quantum system as a coalgebra $X \to \mathcal{Q}X$ (or $X \to \mathcal{Q}_f X$) instead of $X \to \mathcal{Q}(1 + \Sigma \times X)$. Here $\mathcal{Q}_f$ is a *finite support* variant of the monad $\mathcal{Q}$ (see (2.2)). This type of a coalgebra $X \to \mathcal{Q}X$ is much like a quantum Markov chain in [10].

In this chapter, we discuss some types of equivalence for two quantum systems. One is the trace equivalence that is explained in the previous chapter. Another is *bisimilarity* whose coalgebraic characterization is given by Aczel and Mendler [3].

Definition 3.1 (F-bisimulation and F-bisimilarity)**.** Let $F : \mathbf{Sets} \to \mathbf{Sets}$ be an endofunctor on **Sets** and $c : X \to FX$ and $d : Y \to FY$ be two coalgebras of the functor F. A relation $R \in X \times Y$ is a *F-bisimulation* for two coalgebras c, d if there exists a map $e : R \to FR$ that makes two projections π, π' homomorphisms to respective coalgebras:

$$\begin{array}{ccccc} FX & \xleftarrow{\mathcal{Q}(\pi)} & FR & \xrightarrow{\mathcal{Q}(\pi')} & FY \\ \uparrow{\scriptstyle c} & & \uparrow{\scriptstyle e} & & \uparrow{\scriptstyle d} \\ X & \xleftarrow{\pi} & R & \xrightarrow{\pi'} & Y \end{array} .$$

Two states $x \in X$ and $y \in Y$ are *F-bisimilar* if there is a F-bisimulation R which satisfies $(x, y) \in R$.

This abstract definition yields concrete ones for different types of systems. For example, for probabilistic systems or Markov chains, which are represented by $\mathcal{D}$-coalgebras, a relation $R \in X \times Y$ is a $\mathcal{D}$-bisimulation for $c : X \to \mathcal{D}X$ and $d : Y \to \mathcal{D}Y$ if for each pair $(x, y) \in R$ there is a *weight function* $w : R \to [0, 1]$ which is added up to 1 with $c(x)(x') = \sum_{(x',y') \in R} w(x', y')$ for all $x' \in X$ and $d(y)(y') = \sum_{(x',y') \in R} w(x', y')$ for all $y' \in Y$. This weight-function-based description is the well-known concrete definition e.g. in [4]. A $\mathcal{Q}$-bisimulation (or $\mathcal{Q}_f$-bisimulation) is defined in the same manner, except that the range of a weight function w becomes the set $\mathcal{QO}$ of quantum operations with the trace condition.

For probabilistic systems, there is an alternative formulation of equivalence by Larsen and Skou [26]. Larsen and Skou define "*probabilistic bisimulation*" as follows: an equivalence relation $R \in X \times X$ is a probabilistic bisimulation for two systems $c, d : X \to \mathcal{D}X$ with the same carrier if for each pair $(x, y) \in R$ the following sums probabilities coincide:

$$\sum_{x' \in Q} c(x)(x') = \sum_{x' \in Q} d(y)(x'), \quad \text{for each } R\text{-equivalence class } Q \in X/R.$$

In [33], it is shown that two notions of bisimulation—bisimulation in Definition 3.1, and probabilistic bisimulation by Larsen and Skou—coincides.

In the first section of this chapter, we will also defined this Larsen and Skou's equivalence-class-based bisimulation for quantum systems, and we will show that the two notions do not coincide unlike a probabilistic situation.

3.1 Two notions of Bisimulation for Valuation Functor

First, we employ a useful idea in [22], which generalize a multiset functor to encompass the distribution functor $\mathcal{D}$ and the monad $\mathcal{Q}_f$. This generic notion of multiset functor is also used in next chapter to describe a modal logic for quantum systems.

Definition 3.2 (valuation functor $\mathcal{V}_O$)**.** For M and O where

- $(M, +, 0, \leq)$ is a partially ordered commutative monoid and satisfies:

$$\forall x, y \in M.\, x \leq x + y; \tag{3.1}$$

- O is a downward-closed subset of M,

a *valuation functor* $\mathcal{V}_O : \mathbf{Sets} \to \mathbf{Sets}$ is defined by:

$$\mathcal{V}_O X = \{\phi : X \to O \;\big|\; supp(\phi) \text{ is finite} \quad \text{and} \quad \sum_{x \in X} \phi(x) \in O\}$$

$$\mathcal{V}_O(f)(\phi) = \lambda y \in Y. \sum_{x \in f^{-1}(y)} \phi(x)$$

where X is a set, $f : X \to Y$ is a function, and $\phi \in \mathcal{V}_O X$ is a *valuation*.

As shown in the original paper [22], an ordinary multiset functor is indeed acquired by $O = M$ and the distribution functor $\mathcal{D}$ is obtained with $M = (\mathbb{R}^{\geq 0}, 0, +)$ and $O = [0, 1] \subseteq \mathbb{R}^{\geq 0}$. We can also get a functor $\mathcal{Q}_f$ as an instance of a valuation functor $\mathcal{V}_O$.

Proposition 3.3. *The functor $\mathcal{Q}_f$ is a valuation functor.*

Proof. The trace condition (2.1) is used to define $\mathcal{Q}_f$. This *normlizing factor* can be gained in terms of a monoid with partial order and its downward-closed subset.

We have a commutative monoid $M = \prod_{m,n} \mathcal{S}_{m,n}$. Here $\mathcal{S}_{m,n}$ is a set of completely positive super-operators from a Hilbert space $\mathbb{C}^{2^m}$ to $\mathbb{C}^{2^n}$, which are not required to be trace non-increasing (unlike quantum operations). The addition operation, the unit, and the partial order are defined pointwise: for $s, s' \in \prod_{m,n} \mathcal{S}_{m,n}$,

$$(s + s')_{m,n}(\rho) := (s)_{m,n}(\rho) + (s')_{m,n}(\rho)$$
$$(0)_{m,n}(\rho) := 0$$
$$s \leq s' \overset{def}{\iff} \forall m \in \mathbb{N}. \forall \rho \in \mathcal{DM}_m. \forall n \in \mathbb{N}. (s)_{m,n}(\rho) \sqsubseteq (s')_{m,n}(\rho)$$

where 0 at the right side is a zero matrix and $\sqsubseteq$ is the Löwner partial order.

The subset $Tr\mathcal{QO}$ of $\prod_{m,n} \mathcal{S}_{m,n}$ is defined as:

$$s \in Tr\mathcal{QO} \overset{def}{\iff} \forall m \in \mathbb{N}. \forall \rho \in \mathcal{DM}_m. \sum_{n \in \mathbb{N}} \left(\mathrm{tr}\Big((s)_{m,n}(\rho) \Big) \right) \leq 1$$

Now, we show that $\prod_{m,n} \mathcal{S}_{m,n}$ and $Tr\mathcal{QO}$ satisfy the condition in Definition 3.2. For two elements $s, s' \in \prod_{m,n} \mathcal{S}_{m,n}$,

$$\begin{aligned}(s+s')_{m,n}(\rho) - (s)_{m,n}(\rho) &= ((s)_{m,n}(\rho) + (s')_{m,n}(\rho)) - (s)_{m,n}(\rho)) \\ &= (s')_{m,n}(\rho).\end{aligned}$$

The super-matrix $(s')_{m,n}(\rho)$ is positive, therefore $s \leq s + s'$.

If we take $s, s' \in \prod_{m,n} \mathcal{S}_{m,n}$ with $s \leq s'$ and $s' \in Tr\mathcal{QO}$, the inequality $(s)_{m,n}(\rho) \sqsubseteq (s')_{m,n}(\rho)$ implies $\mathrm{tr}\big((s)_{m,n}(\rho)\big) \leq \mathrm{tr}\big((s')_{m,n}(\rho)\big)$, thus:

$$\begin{aligned}\sum_{n\in\mathbb{N}} \Big(\mathrm{tr}\Big((s)_{m,n}(\rho)\Big)\Big) &\leq \sum_{n\in\mathbb{N}} \Big(\mathrm{tr}\Big((s)_{m,n}(\rho)\Big)\Big) \\ &\leq 1,\end{aligned}$$

therefore $s \in Tr\mathcal{QO}$. □

We review a bisimulation in Definition 3.1 for a valuation functor $\mathcal{V}_O$.

Definition 3.4 ($\mathcal{V}_O$-bisimulation)**.** Let $\mathcal{V}_O$ be a valuation functor, and $c : X \to \mathcal{V}_O X$ and $d : Y \to \mathcal{V}_O Y$ be two coalgebras. A relation $R \in X \times Y$ is a *$\mathcal{V}_O$-bisimulation* for c, d if for each pair $(x, y) \in R$, there is a *weight function* $w : X \times Y \to O$ for $c(x)$ and $d(y)$ w.r.t R. Here a weight function $w : X \times Y \to O$ of $\mu \in \mathcal{V}_O X$ and $\nu \in \mathcal{V}_O Y$ w.r.t R is satisfying:

- $supp(w) \subseteq R$;
- $\sum_{(x,y)\in X\times Y} w(x, y) \in O$;
- $\mu(x) = \sum_{y\in Y} w(x, y)$, for all $x \in X$;
- $\nu(y) = \sum_{x\in X} w(x, y)$, for all $y \in Y$.

Now, the notion of probabilistic bisimulation by Larsen and Skou is extended to a valuation functor $\mathcal{V}_O$. To avoid the confusion, we call this type of bisimulation *LS-bisimulation*.

Definition 3.5 (LS-bisimulation)**.** Let $\mathcal{V}_O$ be a valuation functor and $c, d : X \to \mathcal{V}_O X$ be two coalgebras of $\mathcal{V}_O$ with the same carrier X. An equivalence relation $R \subseteq X \times X$ is an *LS-bisimulation* for c, d if R satisfies: for all $x, y \in X$,

$$\begin{aligned}&(x, y) \in R \\ &\quad \Longrightarrow \sum_{x'\in Q} c(x)(x') = \sum_{x'\in Q} d(y)(x'), \quad \text{for each } R\text{-equivalence class } Q \in X/R.\end{aligned}$$

At first sight, it seems to be difficult to categorically formalize this equivalence-class-based characterization, but this can be identified as a *behavioral equivalence* which is another notion of coalgebraic equality [21].

Definition 3.6 (behavioral equivalence)**.** Let $F : \mathbf{Sets} \to \mathbf{Sets}$ be an endofunctor on **Sets** and $c : X \to FX$ and $d : Y \to FY$ be two coalgebras. Two elements $x \in X$ and $y \in Y$ are called *behaviorally equivalent* if there are a coalgebra $e : Z \to FZ$ and a cospan of coalgebra homomorphisms $h : X \to Z$ and $k : Y \to Z$:

$$\begin{array}{ccccc} FX & \xrightarrow{F(h)} & FZ & \xleftarrow{F(k)} & FY \\ {\scriptstyle c}\uparrow & & {\scriptstyle e}\uparrow & & \uparrow{\scriptstyle d} \\ X & \xrightarrow{h} & Z & \xleftarrow{k} & Y \end{array} \qquad \text{with} \quad h(x) = k(y).$$

An LS-bisimulation can be described as a *quotient coalgebra* $b : X/R \to \mathcal{V}_O X/R$ which forms a cospan with a *quotient map* $q : X \to X/R$:

$$\begin{array}{ccccc} \mathcal{V}_O X & \xrightarrow{\mathcal{V}_O(q)} & \mathcal{V}_O(X/R) & \xleftarrow{\mathcal{V}_O(q)} & \mathcal{V}_O X \\ {\scriptstyle c}\uparrow & & {\scriptstyle b}\uparrow & & {\scriptstyle d}\uparrow \\ X & \xrightarrow{q} & X/R & \xleftarrow{q} & X \end{array}$$

by taking $b : X/R \to \mathcal{V}_O X/R$ as $b([x])(Q) = \sum_{x' \in Q} c(x)(x') = \sum_{x' \in Q} d(x)(x')$ where x is a representative element of an equivalence class $[x]$. The well-definedness of a function b is ensured by the condition of LS-bisimulation. Thus, this diagram shows that each element $(x, y) \in R$ is behaviorally equivalent.

It is well known that the two notions—F-bisimilarity in Definition 3.1 and behavioral equivalence—are coincide for functors that preserve weak pullbacks [31] [21]. Therefore, it is natural to obtain the relationship of two notions of bisimulation—$\mathcal{V}_O$-bisimulation and LS-bisimulation—to the following form.

Proposition 3.7. *Let $\mathcal{V}_O$ be a valuation functor and $c, d : X \to \mathcal{V}_O X$ be two coalgebras. For an equivalence relation $R \in X \times X$, consider the following conditions:*

1. *R is a $\mathcal{V}_O$-bisimulation for c, d;*
2. *R is an LS-bisimulation for c, d.*

We have 1 *implies* 2*; moreover, 2 implies 1 in case $\mathcal{V}_O$ preserves weak pullbacks.*

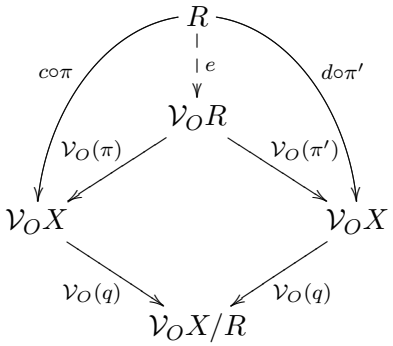
Proof. For an equivalence relation $R \in X \times X$, the below diagram forms both pullback and pushout with $q : X \to X/R$ is a quotient map:

$$\begin{array}{ccccc} & & R & & \\ & {\scriptstyle \pi}\swarrow & & \searrow{\scriptstyle \pi'} & \\ X & & & & X \\ & {\scriptstyle q}\searrow & & \swarrow{\scriptstyle q} & \\ & & X/R & & \end{array}$$

First we show 1⇒2. Assume that there is a coalgebra $e : R \to \mathcal{V}_O R$ and two projections π, π' are homomorphisms, then a pushout $b : X/R \to \mathcal{V}_O X/R$ is obtained by using that the forgetful functor $U : \mathbf{Coalg}(\mathcal{D}) \to \mathbf{Sets}$ creates colimits. The pushout equalizes two projection homomorphisms with the quotient map q, hence we get $q \circ \pi = q \circ \pi'$ as a homomorphism from q to b.
Then, for all $(x, y) \in R$,

$$\begin{aligned} \lambda Q \in X/R. \sum_{x' \in Q} c(x)(x') &= \lambda Q \in X/R. \sum_{x' \in q^{-1}(Q)} c(x)(x') \\ &= (\mathcal{D}(q) \circ c)(x) \\ &= (b \circ q)(x) \\ &= (b \circ q \circ \pi')(x, y) \\ &= (b \circ q \circ \pi)(x, y) \\ &= (b \circ q)(y) \\ &= (\mathcal{D}(q) \circ d)(y) \\ &= \lambda Q \in X/R. \sum_{x' \in Q} c(x)(x') \end{aligned}$$

Next we show 2⇒1 using that $\mathcal{V}_O$ preserves weak-pullbacks. Conversely, we assume LS-bisimulation and then we obtain a quotient coalgebra with a cospan homomorphism q. We get a pullback $\pi, \pi' : R \to X$ from $q : X \to X/R$ and then $\mathcal{V}_O(\pi), \mathcal{V}_O(\pi') : \mathcal{V}_O R \to \mathcal{V}_O X$ becomes a weak-pullback because $\mathcal{V}_O$ preserves weak-pullbacks. Consequently, a mediating map $e : R \to \mathcal{D}R$ is obtained by the role of a weak pullback:



□

3.2 Refinability and Preservation of Weak-Pullback

In the next step, the *weak-pullback-preserving* property for a valuation functor is discussed. We follow [13] and review some properties of monoids.

Definition 3.8 ((m,n)-refinability)**.** Let M be a commutative monoid with additive structure $(M, 0, +)$. For $m, n \in \mathbb{N}$, a monoid M is (m,n)-*refinable* if for two sequences $r_1, \ldots, r_m, c_1, \ldots, c_n$ with an equality:

$$\sum_i r_i = \sum_j c_j$$

there is an $m \times n$-matrix which has elements $(m_{i,j})$ with:

$$\begin{cases} r_i = \sum_j m_{i,j} \\ c_j = \sum_i m_{i,j} \end{cases} \quad \text{which is illustrated as} \quad \begin{array}{ccc|c} m_{1,1} & \cdots & m_{1,n} & r_1 \\ \vdots & \ddots & \vdots & \vdots \\ m_{m,1} & \cdots & m_{m,n} & r_2 \\ \hline c_1 & \cdots & c_n & \end{array}$$

The refinability for all m, n can be derived from a smaller case and positivity. *Positivity* here means that $m_1 + m_2 = 0$ implies $m_1 = m_2 = 0$ for all $m_1, m_2 \in M$.

Proposition 3.9. *A commutative monoid $(M, +, 0)$ is (m,n)-refinable for all $m, n \in \mathbb{N}$ if and only if M is $(2,2)$-refinable and positive.*

Proof. See [13, Prop. 5.10]. □

In general, a multiset functor does not preserve weak pullbacks, but a monoid with refinability and positivity guarantees the preserving property.

Proposition 3.10. *Let $M = (M, 0, +)$ be a commutative monoid. A multiset functor $\mathcal{M}_M$ preserves weak pullbacks if and only if M is $(2,2)$-refinable and positive.*

Proof. See [13, Theorem 5.13]. □

In order to cover probabilistic or quantum systems, we slightly generalize the previous definitions and results in [13] to valuation functors in Definition 3.2. For a subset O of a commutative monoid M that satisfies the condition of Definition 3.2, O is (m,n)-*refinable* if for two sequences $r_1,\dots,r_m,c_1,\dots,c_n$ with $\sum_i r_i = \sum_j c_j \in O$ there is $(m_{i,j})$ with $r_i = \sum_j m_{i,j}$ and $c_j = \sum_i m_{i,j}$.

The condition of Definition 3.2 implies M is positive, thus we get a characterization of a weak-pullback-preserving property for valuation functors as the below proposition.

Proposition 3.11. *Let a commutative monoid $M = (M, 0, +)$ and $O \in M$ satisfy the condition of Definition 3.2. A valuation functor $\mathcal{V}_O$ preserves weak pullbacks if and only if O is $(2,2)$-refinable.*

Proof. We can prove this in a way similar to [13, Theorem.5.13]. □

The distribution functor $\mathcal{D}$ is clearly an example because the real interval $[0,1]$ is $(2,2)$-refinable, therefore $\mathcal{D}$ preserves weak pullbacks. Additionally, the system whose transition is weighted by natural number is specified by a multiset functor with a commutative monoid $(\mathbb{N}, +, 0)$. The natural number $\mathbb{N}$ is $(2,2)$-refinable and so $\mathcal{M}_{\mathbb{N}}$ preserves weak pullbacks, hence we can also apply the characterization of bisimulation for such systems as in Proposition 3.7.

3.3 Refinability of Quantum Operations

When it comes to quantum systems, the refinability of the set of quantum operations fails. We first show an equality of quantum operations from $\mathcal{DM}_2$ to $\mathcal{DM}_1$ that cannot be divided into a 2×2-matrix.

Lemma 3.12. *Four quantum operations $\{|0\rangle\}, \{|1\rangle\}, \{|+\rangle\}, \{|-\rangle\}$ satisfy:*

$$\langle 0|\_|0\rangle + \langle 1|\_|1\rangle = \langle +|\_|+\rangle + \langle -|\_|-\rangle,$$

Each side of equation represents an operation of taking a trace value of $\rho \in \mathcal{DM}(C^2)$. There is no decomposition $(\mathcal{E}_{i,j})$ where:

$$\begin{array}{cc|c} \mathcal{E}_{1,1} & \mathcal{E}_{1,2} & \langle 0|\_|0\rangle \\ \mathcal{E}_{2,1} & \mathcal{E}_{2,2} & \langle 1|\_|1\rangle \\ \hline \langle +|\_|+\rangle & \langle +|\_|+\rangle & \end{array} \tag{3.2}$$

Proof. We assume there are four operations $(\mathcal{E}_{i,j})$ make the decomposition above. Let $\mathcal{E}_{1,1} = \{E^{(i)} : i \in I\}$ and $\mathcal{E}_{1,2} = \{F^{(j)} : j \in J\}$ be the operator-sum representation of $\mathcal{E}_{1,1}$ and $\mathcal{E}_{1,2}$, respectively. The top row of the matrix implies:

$$\langle 0|\_|0\rangle = \sum_{i\in I} E^{(i)}(\_)(E^{(i)})^\dagger + \sum_{j\in J} E^{(j)}(\_)(E^{(j)})^\dagger.$$

We apply $|1\rangle\langle 1| \in \mathcal{DM}(\mathbb{C}^2)$ to each side of the equation:

$$0 = \sum_{i\in I} |E^{(i)}|1\rangle|^2 + \sum_{j\in J} |E^{(j)}|1\rangle|^2.$$

Thus, $E^{(i)}|1\rangle = 0$ for all $i \in I$ and $F^{(j)}|1\rangle = 0$ for all $j \in J$. It must be that $\mathcal{E}_{1,1} = \{r|0\rangle\}$ and $\mathcal{E}_{1,2} = \{(1-r)|0\rangle\}$ with $r \in [0,1]$ We follow the same analysis for the left column of the matrix and obtain $\mathcal{E}_{1,1} = \{r'|+\rangle\}$ and $\mathcal{E}_{2,1} = \{(1-r')|+\rangle\}$ with $r' \in [0,1]$. It is obvious that any of $\mathcal{E}, \mathcal{E}', \mathcal{F}, \mathcal{F}'$ must not be 0. However, $\{r|0\rangle\} = \{r'|+\rangle\}$ with $r \neq 0, r' \neq 0$ is contradiction, therefore the decomposition of the table (3.2) is impossible. □

We obtain the main result easily from the above counter example.

Proposition 3.13. *$Tr\mathcal{QO}$ is not $(2,2)$-refinable.*

Proof. We take four elements $r_1, r_2, c_1, c_2 \in Tr\mathcal{QO}$ as follow:

$$(r_1)_{m,n}(\rho) := \begin{cases} \langle 0|\rho|0\rangle & \text{if } m=2, n=1 \\ 0 & \text{otherwise} \end{cases}$$

$$(r_2)_{m,n}(\rho) := \begin{cases} \langle 1|\rho|1\rangle & \text{if } m=2, n=1 \\ 0 & \text{otherwise} \end{cases}$$

$$(c_1)_{m,n}(\rho) := \begin{cases} \langle +|\rho|+\rangle & \text{if } m=2, n=1 \\ 0 & \text{otherwise} \end{cases}$$

$$(c_2)_{m,n}(\rho) := \begin{cases} \langle -|\rho|-\rangle & \text{if } m=2, n=1 \\ 0 & \text{otherwise} \end{cases}$$

The decomposition of $r_1 + r_2 = c_1 + c_2$ leads to the same contradiction for the $(2,1)$-component of the operators. □

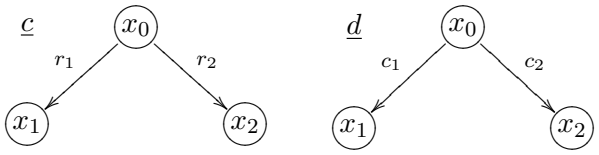
This failure of refinability implies that $\mathcal{Q}$-bisimulation cannot be characterized by equivalence class by Proposition 3.7 and 3.11. Finally, we give an example of two concrete quantum systems whose states are not $\mathcal{Q}$-bisimilar but behaviorally equivalent. This is indeed a consequence of discussions so far.

Proposition 3.14. *Let two coalgebras $c, d : X \to \mathcal{Q}X$ be:*

$$c\colon X \to \mathcal{Q}X \qquad\qquad d\colon X \to \mathcal{Q}X$$

$$c(x)(x') := \begin{cases} r_1 & \text{if } x = x_0, x' = x_1 \\ r_2 & \text{if } x = x_0, x' = x_2 \\ 0 & \text{otherwise} \end{cases} \qquad d(x)(x') := \begin{cases} c_1 & \text{if } x = x_0, x' = x_1 \\ c_2 & \text{if } x = x_0, x' = x_2 \\ 0 & \text{otherwise} \end{cases}$$

where $X = \{x_o, x_1, x_2\}$ and r_1, r_2, c_1, c_2 are specified previously in Proposition 3.13. Those two are more intuitively described as automata:



The two states x_0 and x_0 are behaviorally equivalent for c, d, but they are not $\mathcal{Q}$-bisimilar for c, d.

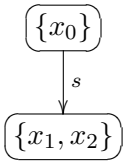
Proof. Behaviorally equivalence is showed as follows. We take an equivalence relation $R \in X \times X$ as:

$$R := \{(x_o, x_o), (x_1, x_1), (x_2, x_2), (x_1, x_2), (x_2, x_1)\}.$$

A quotient coalgebra $e : X/R \to \mathcal{Q}(X/R)$ is obtained as a cospan of two systems:

$$e\colon X/R \to \mathcal{Q}(X/R)$$

$$e(z)(z') := \begin{cases} s & \text{if } z = \{x_o\}, z' = \{x_1, x_2\} \\ 0 & \text{otherwise} \end{cases}$$



where $s \in Tr\mathcal{QO}$ represents an operation that takes a trace value of density matrix and $s = r_1 + r_2 = c_1 + c_2$:

$$(s)_{m,n}(\rho) := \begin{cases} \mathrm{tr}(\rho) & \text{if } m = 2, n = 1 \\ 0 & \text{otherwise} \end{cases}$$

However, there is no bisimulation $R \in X \times X$ with $(x_0, x_0) \in R$. We prove this by contradiction. First, assume a bisimulation R for the coalgebras c and d and satisfies $(x_0, x_0) \in R$. Then, There is a weight function $w \in \mathcal{Q}R$ such that:

$$c(x_0)(x') = \sum_{(x',y') \in R} w(x', y') \qquad \text{for all } x' \in X$$
$$d(x_0)(y') = \sum_{(x',y') \in R} w(x', y') \qquad \text{for all } y' \in Y$$

This requirements to $w \in \mathcal{Q}(R)$ is represented as a table:

	y_1	y_2	y_3	
x_1	$w(x_1, y_1)$	$w(x_1, y_2)$	$w(x_1, y_3)$	0
x_2	$w(x_2, y_1)$	$w(x_2, y_2)$	$w(x_2, y_3)$	r_1
x_3	$w(x_3, y_1)$	$w(x_3, y_2)$	$w(x_3, y_3)$	r_2
	0	c_1	c_2	

However, this situation leads to the contradiction in the proof of Proposition 3.13. Therefore, x_o and x_o cannot be bisimilar for c, d. □

Chapter 4

Modal Logic and Expressivity

In the previous chapters, we have discussed some types of equivalence—trace equivalence, bisimilarity, behavioral equivalence—for transition systems. Especially in the case of non-deterministic systems, it is well known that bisimilarity can be characterized by the logical approach, which means there is a modal logic that is powerful enough to distinguish non-bisimilar states. This is first formulated by Hennessy and Milner in [18], so such a property of a modal logic is called the *Hennessy-Milner property*.

There are some categorical approaches to describing a connection between coalgebras and modal logics. In [22], the expressivity of some general class of logics is shown in a categorical setting for some types of transition systems such as non-determinism, probabilistic, and monoid-weighted systems. We follow [22] and find out a modal logic that is suitable to characterize the behavioral equivalence of quantum systems.

4.1 Coalgebraic Modal Logic with Dual Adjunction

First, we explain the categorical setting for combining two notions, coalgebras and a modal logic.

Definition 4.1 (coalgebraic modal logic, abstractly)**.** Let $\mathcal{C}, \mathcal{A}$ be a category and T, P, L, F be a functor written in the diagram:

$$T \circlearrowright \mathcal{C}^{\mathrm{op}} \underset{F}{\overset{P}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} \mathcal{A} \circlearrowright L$$

Moreover, we assume a some constructs:

- a natural transformation $\sigma : LP \Rightarrow PT$;
- the functor L has an initial algebra with $L(\mathrm{Form}) \xrightarrow{\cong} \mathrm{Form}$;
- the category $\mathcal{C}$ carries a *factorization system* $(\mathfrak{M}, \mathfrak{E})$ (see e.g. [19].)

A functor $L : A \to A$ together with a natural transformation $\sigma : LP \Rightarrow PT$ is called a *coalgebraic modal logic* for the functor $T : \mathcal{C} \to \mathcal{C}$.

In this general setting, each component indeed has an concrete meaning as follows. For an arbitrary coalgebra $c : X \to TX$ in the base category $\mathcal{C}$, we obtain a L-algebra by the *contravariant* functor $P : \mathcal{C}^{\mathrm{op}} \to \mathcal{A}$ and the natural transformation $\sigma : LP \Rightarrow PT$:

$$LPX \xrightarrow{\sigma_X} PTX \xrightarrow{P(c)} PX.$$

Then, the initial L-algebra gives rise to a unique homomorphism $[\![-]\!] : \mathrm{Form} \to PX$ by the initiality:

$$\begin{array}{ccc} L(\mathrm{Form}) & \xrightarrow{L([\![-]\!])} & LPX \\ {\scriptstyle\cong}\downarrow & & \downarrow{\scriptstyle P(c)\circ\sigma_X} \\ \mathrm{Form} & \xrightarrow[[\![-]\!]]{} & PX \end{array}$$

This unique map can be recognized as an *interpretation function* which maps a formula $\phi \in \mathrm{Form}$ to its *interpretation* $[\![\phi]\!]_c \in PX$. By the adjunction $F \dashv P$, the *transpose* of $[\![-]\!]$ is obtained as a theory map $th_c : X \to F(\mathrm{Form})$ mapping a state $x \in X$ to the set of formulas $th_c(x) \in F(\mathrm{Form})$ that x satisfies.

For now, expressivity of a coalgebraic modal logic is characterized as the following form.

Definition 4.2 (expressivity)**.** Assume the situation in Definition 4.1. The relation $\equiv_c \subseteq X \times X$ of a *logically indistinguishable* pair of states is characterized as an equalizer:

$$\equiv_c \dashrightarrow X \times X \underset{th_c\circ\pi'}{\overset{th_c\circ\pi}{\rightrightarrows}} F(\mathrm{Form})$$

The coalgebraic modal logic is *expressive* if each logically indistinguishable elements $x \equiv_c x'$ are behaviorally equivalent for all coalgebra $c : X \to TX$.

The inverse implication of expressivity is not difficult to be shown. However, in order to characterize expressivity property, another thing matters. There is a bijective correspondence of two natural transformations:

$$\frac{LP \overset{\sigma}{\Rightarrow} PT}{TF \underset{\overline{\sigma}}{\Rightarrow} FL}$$

which is obtained through the adjunction $F \dashv P$. For coalgebraic modal logic σ, its transpose is written as $\overline{\sigma}$. Then, we present the well known result of a coalgebraic modal logic and its expressivity.

Proposition 4.3. *In the situation of Definition 4.1, the coalgebraic modal logic is expressive if the functor T* preserves abstract monos *and the transpose $\overline{\sigma} : TF \Rightarrow FL$ is* componentwise abstract mono*.*

Proof. See [24]. □

The abstract setting for a valuation functor $\mathcal{V}_O$ is shown in [22]. The *modality* $\Box_o : \mathcal{P}X \to \mathcal{P}\mathcal{V}_O X$ is given as:

$$\Box_o(U) := \{\phi \in \mathcal{V}_O(X) \mid \sum_{x\in U} \phi(x) \geq o\} \tag{4.1}$$

where $o \in O$, $U \in X$. From this modality, one obtain a special case in Definition 4.1.

Proposition 4.4. *Let M, O satisfy the conditions in Definition 3.2. The valuation functor $\mathcal{V}_O$ carries a coalgebraic modal logic: an endofunctor $\mathcal{K}_O : \mathbf{MSL} \to \mathbf{MSL}$ and a natural transformation $\boxtimes : \mathcal{K}_O\mathcal{P} \to \mathcal{P}\mathcal{V}_O$ with:*

$$\mathcal{V}_O \circlearrowright \mathbf{Sets}^{\mathrm{op}} \underset{\mathcal{F}}{\overset{\mathcal{P}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} \mathbf{MSL} \circlearrowright \mathcal{K}_O$$

Each part of the diagram is specified as follows:

- **MSL** *is a category of meet semilattices and meet preserving functions;*
- $\mathcal{P} : \mathbf{Sets}^{\mathrm{op}} \to \mathbf{MSL}$ *is a contravariant powerset functor;*
- $\mathcal{F} : \mathbf{MSL} \to \mathbf{Sets}^{\mathrm{op}}$ *maps a meet semilattice A to the set $\mathcal{F}(A) \subseteq \mathcal{P}(A)$ of its filters.*

Moreover, if M is cancellative*, that is: for all $x, y, z \in M$,*

$$x + z \leq y + z \implies x \leq y, \tag{4.2}$$

the transpose of $\boxtimes$ is componentwise mono, therefore the coalgebra modal logic involved is expressive.

Proof. An endofunctor $\mathcal{K}_O : \mathbf{MSL} \to \mathbf{MSL}$ is constructed from the modality in (4.1). For detail, see [22, Prop. 11, Theorem 13]. □

4.2 Expressive Modal Logic for Quantum Systems

To use this result in a quantum situation $T = \mathcal{Q}_f$, it is enough to show that a monoid $\prod_{m,n} \mathcal{S}_{m,n}$ is cancellative.

Proposition 4.5. *$\prod_{m,n} \mathcal{S}_{m,n}$ is cancellative.*

Proof. The additive structure and the partial order on $\prod_{m,n} \mathcal{S}_{m,n}$ are defined in Proposition 3.3. For $s_1, s_2, s_3 \in \prod_{m,n} \mathcal{S}_{m,n}$, we assume $s_1 + s_3 \leq s_2 + s_3$.

$$\begin{aligned}(s_2)_{m,n}(\rho) - (s_1)_{m,n}(\rho) &= ((s_2)_{m,n}(\rho) + (s_3)_{m,n}(\rho)) - ((s_1)_{m,n}(\rho) + (s_3)_{m,n}(\rho)) \\ &= (s_2 + s_3)_{m,n}(\rho) - (s_1 + s_3)_{m,n}(\rho)\end{aligned}$$

Therefore $s_1 \leq s_2$. □

The abstract discussion so far induces a concrete expressive modal logic for quantum systems. From now, we present a modal logic for the functor $\mathcal{Q}_f$ and we call this logic *QML* (quantum modal logic).

Definition 4.6 (syntax of QML)**.** The syntax of QML is as follows:

$$\phi ::= \top \;\big|\; \phi_1 \wedge \phi_2 \;\big|\; \Box_{\mathcal{E}} \phi$$

where $\mathcal{E} \in Tr\mathcal{QO}$.

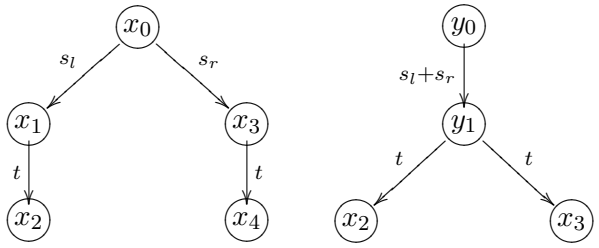
Definition 4.7 (semantics of QML)**.** Let $c : X \to \mathcal{Q}X$ be a $\mathcal{Q}$ coalgebra. For any state $x \in X$, the *satisfaction relation* $\vDash_c$ is defined inductively as:

$$\begin{aligned} x \vDash_c \top &\qquad always \\ x \vDash_c \phi_1 \wedge \phi_2 &\iff x \vDash_c \phi_1 \;\text{ and }\; x \vDash_c \phi_2 \\ x \vDash_c \Box_{\mathcal{E}} \phi &\iff \sum_{x' \vDash_c \phi} c(x)(x') \sqsupseteq \mathcal{E} \end{aligned}$$

where $\mathcal{E} \in Tr\mathcal{QO}$.

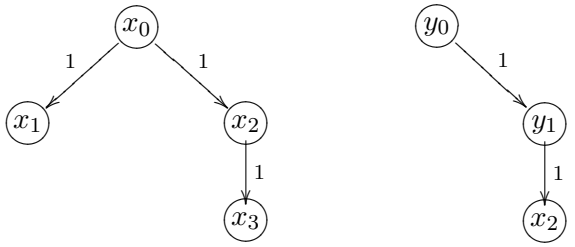
Finally, we give some simple and fundamental examples in order to illustrate the QML more concretely. These examples follow ones from some other settings, such as non-determinism in [18], and probabilistic systems in [26].

Example 4.8. First, the conjunction operator of QML is needed to separate two systems below:

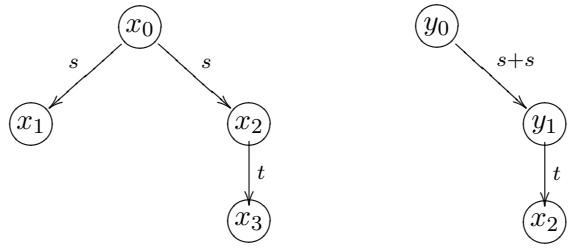


For a formula $\phi = (\Box_{s_l+s_r}\Box_t\top) \wedge (\Box_{s_l+s_r}\Box_t\top)$, the both states x_o and y_o satisfies ϕ. However, if we take a formula $\psi = \Box_{s_l+s_r}(\Box_t\top \wedge \Box_t\top)$, the right state y_0 satisfies ψ but the left one x_o does not satisfy it. Hence, we can see that conjunction is necessary to specify the occasion a branch occurs.

Next, the syntax of QML does not contain a negation operator. This is because the set of quantum operators is cancellative unlike boolean algebra $(\{0,1\}, \vee)$. While the following non-determinism systems:



cannot be distinguished without negation (a negation formula $\Box_1\neg\Box_1\top$ distinguishes them), the cancellative multiset systems (which include quantum systems):



can be separated by some formulas, for example $\phi = \Box_{s+s}\Box_t\top$.

Chapter 5

Conclusions and Future Work

We have analyzed the quantum monad $\mathcal{Q}$ (or its finitary variant $\mathcal{Q}_f$) from a coalgebraic view. A novel notion of QLTS is obtained as an instance of coalgebra $X \to \mathcal{Q}FX$ with a linear-polynomial functor $F = 1 + \Sigma \times \mathrm{id}$. As this type of a quantum system, we modeled some well-known examples—the superdense protocol and the quantum teleportation protocol—and discovered a simulation with a specification system of each protocol, respectively. This is a new technique to verify quantum systems induced from categorical abstraction of branching types. We will refine the simulation bases approach for practical use and explore some examples for verification. For the BB84 protocol, we formalized the anonymity property of a secret key and verified its anonymity with the assumption that an eavesdropper cannot attack quantum data. We will improve this formalization of anonymity property to apply much wider quantum protocols.

Moreover, we have investigated some feature of the systems which are specified by coalgebras in the form $X \to \mathcal{Q}_f X$. Using the analogy to probabilistic systems, we define a new characterization of bisimulation—LS-bisimulation— however this notion is weaker than a standard coalgebraic bisimulation for quantum systems. We have also found an expressive modal logic with quantum modality by means of coalgebraic modal logic. These are minor contributions, however we expect that they can be foundation for future work somewhat.a

The coalgebraic characterization of quantum systems will further benefit from categorical generalization. *Structural operational semantics* (SOS) is utilized to properly describe a specification of *process calculi* [2]. Various flavors of SOS are well studied by using a bialgebra—an algebra represents a syntax, a coalgebra describes a behavior—and a distributive law in the categorical framework [25]. This framework successfully obtained appropriate syntactic formats for LTS [32] and probabilistic transition systems [5]. We also seek a suitable design for a quantum process calculus.

Another direction of the thesis's future work is to explore the quantum variant of the *Giry monad* that is an endofunctor on the category of *measurable spaces* and *measurable functions* [11]. While the coalgebras of the distribution monad represent discrete Markov chains, Giry monad (or some variants) coalgebras describe systems called *Markov processes* [8]. This notion formalizes the probabilistic systems whose state space is *continuous* and which evolves their stages according to a probabilistic law, which should suitably model real world systems in a physics sense. Therefore, we hope to define a monad like the Giry monad which has quantum taste.

References

[1] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *LICS*, pages 415–425. IEEE Computer Society, 2004.

[2] L. Aceto, W. Fokkink, and C. Verhoef. Structural operational semantics. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, pages 197–292. Elsevier, 2001.

[3] P. Aczel and N. Mendler. A final coalgebra theorem. In D. H. Pitt, A. Poigné, and D. E. Rydeheard, editors, *Category Theory and Computer Science*, number 389 in Lect. Notes Comp. Sci., pages 357–365. Springer, Berlin, 1989.

[4] Christel Baier. Polynomial time algorithms for testing probabilistic bisimulation and simulation. In Rajeev Alur and Thomas A. Henzinger, editors, *CAV*, volume 1102 of *Lecture Notes in Computer Science*, pages 50–61. Springer, 1996.

[5] F. Bartels. *On generalised coinduction and probabilistic specification formats. Distributive laws in coalgebraic modelling.* PhD thesis, Free Univ. Amsterdam, 2004.

[6] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, page 175, India, 1984.

[7] Mohit Bhargava and Catuscia Palamidessi. Probabilistic anonymity. In Martín Abadi and Luca de Alfaro, editors, *CONCUR 2005*, volume 3653 of *Lect. Notes Comp. Sci.*, pages 171–185. Springer, 2005.

[8] J. Desharnais, A. Edalat, and P. Panangaden. A logical characterization of bisimulation for labeled markov processes. In *Logic in Computer Science*, 1998.

[9] Yuan Feng, Runyao Duan, and Mingsheng Ying. Bisimulation for quantum processes. *ACM Trans. Program. Lang. Syst.*, 34(4):17, 2012.

[10] Yuan Feng, Nengkun Yu, and Mingsheng Ying. Model checking quantum markov chains. *J. Comput. Syst. Sci.*, 79(7):1181–1198, 2013.

[11] Michele Giry. A categorical approach to probability theory. In *Proc. Categorical Aspects of Topology and Analysis*, volume 915 of *Lect. Notes Math.*, pages 68–85, 1982.

[12] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *STOC*, pages 212–219. ACM, 1996.

[13] H. Peter Gumm and Tobias Schröder. Monoid-labeled transition systems. *Electr. Notes Theor. Comput. Sci.*, 44(1):185–204, 2001.

[14] Ichiro Hasuo. Generic forward and backward simulations. In Christel Baier and Holger Hermanns, editors, *International Conference on Concurrency Theory (CONCUR 2006)*, volume 4137 of *Lect. Notes Comp. Sci.*, pages 406–420. Springer, Berlin, 2006.

[15] Ichiro Hasuo and Naohiko Hoshino. Semantics of higher-order quantum computation via geometry of interaction. In *LICS*, pages 237–246. IEEE Computer Society, 2011.

[16] Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace semantics via coinduction. *Logical Methods in Comp. Sci.*, 3(4:11), 2007.

[17] Ichiro Hasuo, Yoshinobu Kawabe, and Hideki Sakurada. Probabilistic anonymity via coalgebraic simulations. *Theor. Comp. Sci.*, 411(22–24):2239–2259, 2010.

[18] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journ. ACM*, 32(1):137–161, 1985.

[19] J. Hughes and B. Jacobs. Factorization systems and fibrations. In R. Blute and P. Selinger, editors, *Category Theory and Computer Science 2002*, number 69 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2003. `www.elsevier.nl/locate/entcs/volume69.html`.

[20] Bart Jacobs. Coalgebraic walks, in quantum and turing computation. In Martin Hofmann, editor, *FOSSACS*, volume 6604 of *Lecture Notes in Computer Science*, pages 12–26. Springer, 2011.

[21] Bart Jacobs. Introduction to coalgebra. Towards mathematics of states and observations. Draft of a book (ver. 2.0), available online, 2012.

[22] Bart Jacobs and Ana Sokolova. Exemplaric expressivity of modal logics. *J. Log. Comput.*, 20(5):1041–1068, 2010.

[23] C. B. Jay. A semantics for shape. *Science of Comput. Progr.*, 25:251–283, 1995.

[24] Bartek Klin. Coalgebraic modal logic beyond **Sets**. In *MFPS XXIII*, volume 173, pages 177–201. Elsevier, Amsterdam, 2007.

[25] Bartek Klin. Bialgebras for structural operational semantics: An introduction. *Theor. Comput. Sci.*, 412(38):5043–5069, 2011.

[26] Kim Guldstrand Larsen and Arne Skou. Bisimulation through probabilistic testing. *Inf. & Comp.*, 94(1):1–28, 1991.

[27] L. S. Moss. Coalgebraic logic. *Ann. Pure & Appl. Logic*, 96(1-3):277–317, 1999. *Erratum* in *Ann. Pure & Appl. Logic*, 99(1-3):241–259, 1999.

[28] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

[29] Peter Selinger. Towards a quantum programming language. *Math. Struct. in Comp. Sci.*, 14(4):527–586, 2004.

[30] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS*, pages 124–134. IEEE Computer Society, 1994.

[31] Sam Staton. Relating coalgebraic notions of bisimulation. *Logical Methods in Comp. Sci.*, 7(1), 2011.

[32] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Logic in Computer Science*, pages 280–291. IEEE, Computer Science Press, 1997.

[33] E. P. de Vink and J. J. M. M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theor. Comp. Sci.*, 221:271–293, 1999.