On Coinduction and Quantum Lambda Calculi*

Yuxin Deng[1], Yuan Feng[2], and Ugo Dal Lago[3]

1 **Shanghai Key Laboratory of Trustworthy Computing, East China Normal University**
2 **University of Technology Sydney, Australia**
3 **Università di Bologna and INRIA**

Abstract

In the ubiquitous presence of linear resources in quantum computation, program equivalence in linear contexts, where programs are used or executed once, is more important than in the classical setting. We introduce a linear contextual equivalence and two notions of bisimilarity, a state-based and a distribution-based, as proof techniques for reasoning about higher-order quantum programs. Both notions of bisimilarity are sound with respect to the linear contextual equivalence, but only the distribution-based one turns out to be complete. The completeness proof relies on a characterisation of the bisimilarity as a testing equivalence.

1998 ACM Subject Classification F.3.2 Semantics of Programming Languages

Keywords and phrases Quantum lambda calculi; contextual equivalence; bisimulation

1 Introduction

Since two decades ago, the theory of quantum computing has attracted considerable research efforts. Benefiting from the superposition of quantum states, quantum computing may provide remarkable speedup over its classical analogue [32, 16, 17]. As a consequence, a wealth of models and programming languages for describing quantum computation have been introduced.

For many reasons, the functional paradigm fits very well in the picture. One successful attempt in this direction is QUIPPER [15], an expressive functional higher-order language that can be used to program a diverse set of non-trivial quantum algorithms and can generate quantum gate representations using trillions of gates. The group led by Svore introduced LIQUi$|\rangle$ as a modular software architecture designed to control quantum hardware [34]: it enables easy programming, compilation, and simulation of quantum algorithms and circuits. In spite of the success of language design, the semantic foundation of quantum programming languages is not well established. In a series of papers, Selinger and co-authors try to find a denotational semantics for higher-order quantum computation [28, 29, 30, 31]. In the most recent one [24], they propose a denotational model that is adequate with respect to an operational semantics. However, full abstraction for a higher-order language with both classical and quantum resources still remains an open problem.

In quantum mechanics, a fundamental principle is the no-cloning theorem of quantum resources. From a type-theoretic point of view, quantum resources are linear and can be described by linear types in quantum programming languages. How to define appropriate program equivalences for this kind of languages is an interesting problem. Some preliminary

* Partially supported by the National Natural Science Foundation of China (61173033, 61261130589, 61428208) and ANR 12IS02001 PACE. The second author is also supported by Australian Research Council under grant No. DP130102764 and the CAS/SAFEA International Partnership Program for Creative Research Team.

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

results towards this direction have been obtained by omitting quantum effects and only considering nondeterminism and linearity in a functional language [8]. For that restricted setting, a notion of *linear contextual equivalence* is introduced and shown to be nicely related to trace equivalence. Linear contextual equivalence is a special form of contextual equivalence [23] in which the observable behaviour of programs is tested by executing them (at most) once.

In the current work we investigate the operational semantics of the typed quantum λ-calculus proposed in [24]. We aim to develop coinductive proof techniques for linear contextual equivalence (written $\simeq$) of quantum programs. We first define a labelled transition system for the quantum λ-calculus. It is in fact a probabilistic labelled transition system (pLTS) because probability distributions arise naturally when quantum systems are measured. In the underlying pLTS, we consider two notions of probabilistic bisimilarity: one (written $\sim_s$) is state-based because it is directly defined over states and then lifted to distributions; the other ($\sim_d$) is distribution-based as it is a relation between distributions. The relation $\sim_s$ is essentially the probabilistic bisimilarity originally defined by Larsen and Skou [22], representing a branching time semantics. In contrast, the relation $\sim_d$ is strictly coarser. It is in the style of [13, 18], representing a linear time semantics. Both $\sim_s$ and $\sim_d$ are sound proof techniques for $\simeq$, which requires to prove that they are congruence relations. We show the congruence property by adapting Howe's method to the quantum setting. We also find that $\sim_d$ provides a complete proof technique for $\simeq$. In order to prove full-abstraction, we first characterise $\sim_d$ as a testing equivalence $=^{\mathcal{T}}$ given by a simple testing language. Since all the tests in the testing language can be simulated by linear contexts, we obtain that $\simeq \subseteq =^{\mathcal{T}}$, which implies that $\simeq \subseteq \sim_d$. To some extent this is a generalisation of the aforementioned coincidence result obtained in [8] because the distribution-based probabilistic bisimilarity $\sim_d$ captures a notion of trace equivalence in the probabilistic setting very intuitively.

1.1 Other Related Work

Reasoning about program equality in higher-order languages is challenging. Most of the time equivalence between two programs requires them to exhibit the same observable behaviour under any context. To alleviate the burden of dealing with all the contexts, a useful way out is to develop operational methods for proving program equivalence. For example, Abramsky's *applicative bisimulation* [1] has attracted a lot of attention, not only in the classical setting [14, 19, 25, 26], but also in the probabilistic setting [4, 2]. In [4] a notion of probabilistic applicative bisimulation is shown to be a sound technique for proving contextual equivalence. However, completeness fails and can only be recovered when pure, deterministic λ-terms are considered and a coupled logical bisimulation is used in place of applicative bisimulation. In [2] a probabilistic call-by-value λ-calculus is considered, where a probabilistic applicative bisimilarity is shown to be a sound *and complete* proof technique for contextual equivalence. Recently, the third author and Rioli have studied applicative bisimulation in a purely linear quantum λ-calculus, obtaining a soundness result [3]. Following this line of research, our work is carried out in a quantum setting and uses a distribution-based bisimilarity to characterise linear contextual equivalence. We also examine a state-based bisimilarity. It corresponds to the probabilistic applicative bisimilarity discussed above. The characterisation of state-based probabilistic bisimilarity by a set of tests, shown with an involved proof in [33] and with a tradition dated back to [22], is essential for the completeness proof of [2]. For distribution-based bisimilarity, however, a much simpler characterisation exists.

Variants of probabilistic bisimulation have already been used to compare the behaviour of quantum processes [21, 11, 5, 12, 10]. However, as far as we know, the current work is the

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

first to explore operational techniques based on probabilistic bisimulation to reason about contextual equivalence of fully-featured higher-order quantum programs.

1.2 Structure of the Paper

In Section 2, we introduce the syntax, the reduction semantics, and the labelled transition semantics of a quantum λ-calculus. A linear contextual equivalence and two bisimilarities are defined. In Section 3 we show that the two notions of bisimilarity are congruence relations included in linear contextual equivalence. The distribution-based bisimilarity is shown to coincide with a testing equivalence in Section 4. By exploiting this result, we show that the distribution-based bisimilarity is complete with respect to linear contextual equivalence. Finally, we conclude in Section 5.

2 A Quantum λ-Calculus

Following [24] we introduce the syntax and operational semantics of a quantum λ-calculus.

2.1 Syntax

In this typed language, types are given by the following grammar:

$$A, B, C ::= \mathsf{qubit} \mid A \multimap B \mid\ !(A \multimap B) \mid 1 \mid A \otimes B \mid A \oplus B \mid A^l.$$

Here $A \multimap B$ is the usual linear function type and $(!A \multimap B)$ is a non-linear function type. For any arbitrary type A, the type $!A$ can be simulated by $!(1 \multimap A)$. The qubit type is used to classify terms that represent qubit information. Tensor and sum types are standard. We use the notation $A^{\otimes n}$ for A tensored n times. The type A^l denotes finite lists of type A.

Terms are built up from constants and variables, using the following constructs:

$$\begin{array}{rcll}
M, N, P & ::= & x & \text{Variables} \\
& \mid & \lambda x^A . M \mid M\, N & \text{Abstractions and applications} \\
& \mid & \mathtt{skip} \mid M; N & \text{Skip and sequential compositions} \\
& \mid & M \otimes N \mid \mathtt{let}\ x^A \otimes y^B = M\ \mathtt{in}\ N & \text{Tensor products and projections} \\
& \mid & \mathtt{in}_l\, M \mid \mathtt{in}_r\, M & \text{Sums} \\
& \mid & \mathtt{match}\ P\ \mathtt{with}\ (x^A : M \mid y^B : N) & \text{Matches} \\
& \mid & \mathtt{split}^A & \text{Split} \\
& \mid & \mathtt{letrec}\ f^{A \multimap B} x = M\ \mathtt{in}\ N & \text{Recursions} \\
& \mid & \mathtt{new} \mid \mathtt{meas} \mid \mathtt{U} & \text{Quantum operators}
\end{array}$$

Most of the language constructs are standard. The tensor product and tensor projection are related to linearity. The quantum operators are used to prepare quantum systems. The constant $\mathtt{U}$ ranges over a set of elementary unitary transformations on quantum bits. Two typical examples are the Hadamard gate $\mathtt{H}$ and the controlled-not gate $\mathtt{N_c}$.

Variables appearing in the λ-binder, the $\mathtt{let}$-binder, the $\mathtt{match}$-binder, and the $\mathtt{letrec}$-binder are bound variables. We write $fv(M)$ for the set of free variables in term M. We will not distinguish *α-equivalent* terms, which are terms syntactically identical up to renaming of bound variables. If M and N are terms and x is a variable, then $M\{N/x\}$ denotes the term resulting from substituting N for all free occurrences of x in M. More generally, given a list $N_1, \ldots, N_n$ of terms and a list $x_1, \ldots, x_n$ of distinct variables, we write $M\{N_1/x_1, \ldots, N_n/x_n\}$ or simply $M\{\tilde{N}/\tilde{x}\}$ for the result of simultaneously substituting each N_i for free occurrences in M of the corresponding variable x_i.

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY

Values are special terms in the following form

$$V, W ::= x \mid c \mid \lambda x^A.M \mid V \otimes W \mid \mathtt{in}_l\, V \mid \mathtt{in}_r\, W,$$

where c ranges over the set of constants $\{\mathtt{skip}, \mathtt{split}^A, \mathtt{meas}, \mathtt{new}, \mathtt{U}\}$. As syntactic sugar we write $\mathsf{bit} = 1 \oplus 1$, $\mathtt{tt} = \mathtt{in}_r\,\mathtt{skip}$, and $\mathtt{ff} = \mathtt{in}_l\,\mathtt{skip}$.

A *typing assertion* takes the form $\Delta \vdash M : A$, where Δ is a finite partial function from variables to types, M is a term, and A is a type. We write $dom(\Delta)$ for the domain of Δ. We call Δ *exponential* (resp. *linear*) whenever $\Delta(x)$ is (resp. is not) a !-type for each $x \in dom(\Delta)$. We write $!\Delta$ for a context that is exponential. The *type assignment relation* consists of all typing assertions that can be derived from the axioms and rules in Figure 1, where the contexts Δ' and Δ'' are assumed to be linear. The notation $\Delta, x : A$ denotes the partial function which properly extends Δ by mapping x to A, so it is assumed that $x \notin dom(\Delta)$. Similarly, in the notation Δ, Δ' the domains of Δ and Δ' are assmued to be disjoint. We write $\mathcal{P}rog(A) = \{M \mid \emptyset \vdash M : A\}$ for the set of all closed programs of type A.

For any typing assertion $\Delta \vdash M : A$, it is not difficult to see that $fv(M) \subseteq dom(\Delta)$. Let $fqv(M)$ be the set of free variables of qubit type in term M and $fcv(M)$ collects all other types of free variables. Thus we have $fv(M) = fcv(M) \cup fqv(M)$ for any term M. We often separate the free quantum variables in M from the type environment Δ and write $\Delta' \rhd M : A$ where $\Delta', x_1 : \mathsf{qubit}, \ldots, x_n : \mathsf{qubit} = \Delta$ with $\{x_1, \ldots, x_n\} = fqv(M)$. Let a *proved expression* be a triple (Δ, M, A) such that $\Delta \rhd M : A$. If $\Delta = x_1 : A_1, \ldots, x_n : A_n$, a Δ-*closure* is a substitution $\{\tilde{M}/\tilde{x}\}$ where each $M_i \in \mathcal{P}rog(A_i)$. If $\mathcal{R}$ is a relation on terms M with $fcv(M) = \emptyset$, its *open extension*, $\mathcal{R}^\circ$, is the least relation between proved expressions such that

$$(\Delta, M, A)\ \mathcal{R}^\circ\ (\Delta, N, A) \text{ iff } M\{\tilde{P}/\tilde{x}\}\ \mathcal{R}\ N\{\tilde{P}/\tilde{x}\} \text{ for any } \Delta\text{-closure } \{\tilde{P}/\tilde{x}\}\ .$$

We will write $\Delta \rhd M\ \mathcal{R}\ N : A$ to mean that $(\Delta, M, A)\ \mathcal{R}^\circ\ (\Delta, N, A)$. Sometimes we omit the type information if it is not important and simply write $\Delta \rhd M\ \mathcal{R}\ N$.

2.2 The Reduction Semantics

The reduction semantics is defined in terms of an abstract machine simulating the behaviour of the QRAM model [20].

▶ **Definition 1.** A *quantum closure* is a triple $[q, l, M]$ where

- q is a normalized vector of $\mathbb{C}^{2^n}$, for some integer $n \geq 0$. It is called the *quantum state*;
- M is a term, not necessarily closed;
- l is a *linking function* that is an injective map from $fqv(M)$ to the set $\{1, \ldots, n\}$.

We write $dom(l)$ for the domain of l. The notation $l \uplus m$ stands for the union of two linking functions l and m (viewed as two sets of pairs) if their domains are disjoint, otherwise it is undefined. A closure $[q, l, M]$ is *total* if l is surjective, thus a bijection. In that case we write l as $\langle x_1, \ldots, x_n \rangle$ if $dom(l) = \{x_1, \ldots, x_n\}$ and $l(x_i) = i$ for all $i \in \{1 \ldots n\}$. A quantum closure $C = [q, l, M]$ is well typed and has type A in Δ whenever $dom(l) = \{x_1, \ldots, x_m\}$ and $\Delta, x_1 : \mathsf{qubit}, \ldots, x_m : \mathsf{qubit} \vdash M : A$. In this case we write $\Delta \rhd C : A$. The notion of α-equivalence extends naturally to quantum closures. So, e.g., the two states $[q, \langle y \rangle, \lambda x^A.y]$ and $[q, \langle z \rangle, \lambda x^A.z]$ are deemed equivalent. With a slight abuse of language, we call a closure $[q, l, V]$ a *value* when the term V is a value. Most often we will work with closed quantum closures, i.e. with those closures C such that $\emptyset \rhd C : A$. We let Cl be the set of closed quantum closures. For the sake of simplicity, we often assume that the quantum closures we work with are typable without stating it explicitly.

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\frac{A \text{ linear}}{!\Delta, x : A \vdash x : A} \qquad \frac{}{!\Delta, x : !(A \multimap B) \vdash x : A \multimap B} \qquad \frac{!\Delta \vdash V : A \multimap B \quad V \text{ value}}{!\Delta \vdash V : !(A \multimap B)}$$

$$\frac{\Delta, x : A \vdash M : B}{\Delta \vdash \lambda x^A.M : A \multimap B} \qquad \frac{!\Delta, \Delta' \vdash M : A \multimap B \quad !\Delta, \Delta'' \vdash N : A}{!\Delta, \Delta', \Delta'' \vdash MN : B}$$

$$\frac{}{!\Delta \vdash \texttt{skip} : 1} \qquad \frac{!\Delta, \Delta' \vdash M : 1 \quad !\Delta, \Delta'' \vdash N : A}{!\Delta, \Delta', \Delta'' \vdash M; N : A}$$

$$\frac{!\Delta, \Delta' \vdash M : A \quad !\Delta, \Delta'' \vdash N : B}{!\Delta, \Delta', \Delta'' \vdash M \otimes N : A \otimes B} \qquad \frac{!\Delta, \Delta' \vdash M : A \otimes B \quad !\Delta, \Delta'', x : A, y : B \vdash N : C}{!\Delta, \Delta', \Delta'' \vdash \texttt{let } x^A \otimes y^B = M \texttt{ in } N : C}$$

$$\frac{!\Delta, \Delta' \vdash M : A}{!\Delta, \Delta' \vdash \texttt{in}_l\, M : A \oplus B} \qquad \frac{!\Delta, \Delta' \vdash M : B}{!\Delta, \Delta' \vdash \texttt{in}_r\, M : A \oplus B}$$

$$\frac{!\Delta, \Delta' \vdash P : A \oplus B \quad !\Delta, \Delta'', x : A \vdash M : C \quad !\Delta, \Delta'', y : B \vdash N : C}{!\Delta, \Delta', \Delta'' \vdash \texttt{match } P \texttt{ with } (x^A : M \mid y^B : N) : C}$$

$$\frac{}{!\Delta \vdash \texttt{split}^A : A^l \multimap 1 \oplus (A \otimes A^l)} \qquad \frac{!\Delta, \Delta' \vdash M : 1 \oplus (A \otimes A^l)}{!\Delta, \Delta' \vdash M : A^l}$$

$$\frac{!\Delta, f : !(A \multimap B), x : A \vdash M : B \quad !\Delta, \Delta', f : !(A \multimap B) \vdash N : C}{!\Delta, \Delta' \vdash \texttt{letrec } f^{A \multimap B} x = M \texttt{ in } N : C}$$

$$\frac{}{!\Delta \vdash \mathsf{new} : \mathsf{bit} \multimap \mathsf{qubit}} \qquad \frac{}{!\Delta \vdash \mathsf{meas} : \mathsf{qubit} \multimap \mathsf{bit}} \qquad \frac{\texttt{U} \text{ of arity } n}{!\Delta \vdash \texttt{U} : \mathsf{qubit}^{\otimes n} \multimap \mathsf{qubit}^{\otimes n}}$$

Figure 1 Typing Rules.

We now introduce the notion $C \overset{p}{\rightsquigarrow} D$ to mean that closed quantum closure C reduces to D immediately with probability $p \in [0, 1]$. Formally, the one-step reduction $\overset{p}{\rightsquigarrow}$ between quantum closures is the smallest relation including the axioms from Figure 2 together with the structural rule

$$\frac{[q,\, l,\, M] \overset{p}{\rightsquigarrow} [r,\, i,\, N]}{[q,\, j \uplus l,\, \mathcal{E}[M]] \overset{p}{\rightsquigarrow} [r,\, j \uplus i,\, \mathcal{E}[N]]}$$

where $\mathcal{E}$ is any *evaluation context* generated by the grammar

$$\begin{aligned}\mathcal{E} ::= [\,] \mid \mathcal{E}\, M \mid V\, \mathcal{E} \mid \mathcal{E}; M \mid \mathcal{E} \otimes M \mid V \otimes \mathcal{E} \mid \texttt{in}_l\, \mathcal{E} \mid \texttt{in}_r\, \mathcal{E} \\ \mid \texttt{let } x^A \otimes y^B = \mathcal{E} \texttt{ in } M \mid \texttt{match } \mathcal{E} \texttt{ with } (x^A : M \mid y^B : N).\end{aligned}$$

In the two reduction rules for `new`, the quantum state q has size n, and x is a fresh variable. In the rule for unitary transformations, the quantum state r is obtained by applying the k-ary unitary gate $\texttt{U}$ to the qubits $l(x_1), \ldots, l(x_k)$. In other words, $r = (\sigma \circ (\texttt{U} \otimes id) \circ \sigma^{-1})(q)$, where σ is the action on $\mathbb{C}^{2^n}$ of any permutation over $\{1, \ldots, n\}$ such that $\sigma(i) = l(x_i)$ whenever $i \leq k$. In the rules for measurements, we assume that if q_0 and q_1 are normalized quantum states of the form $\sum_j \alpha_j |\varphi_j\rangle \otimes |0\rangle \otimes |\phi_j\rangle$, $\sum_j \beta_j |\varphi_j\rangle \otimes |1\rangle \otimes |\phi_j\rangle$, then r_0 and r_1 are respectively $\sum_j \alpha_j |\varphi_j\rangle \otimes |\phi_j\rangle$, $\sum_j \beta_j |\varphi_j\rangle \otimes |\phi_j\rangle$, where the vectors φ_j has dimension $l(x) - 1$. The reduction semantics defined above employs a *call-by-value* evaluation strategy.

▶ **Lemma 2** (Totality). *Let C and D be two quantum closures and $C \overset{p}{\rightsquigarrow} D$. If C is total then so is D.* ◀

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$
\begin{array}{l}
[q,\, l,\, (\lambda x^A.M)V] \overset{1}{\rightsquigarrow} [q,\, l,\, M\{V/x\}] \\
[q,\, l,\, \mathtt{let}\ x^A \otimes y^B = V \otimes W\ \mathtt{in}\ N] \overset{1}{\rightsquigarrow} [q,\, l,\, N\{V/x, W/y\}] \\
[q,\, l,\, \mathtt{skip}; N] \overset{1}{\rightsquigarrow} [q,\, l,\, N] \\
[q,\, l,\, \mathtt{split}^A\, V] \overset{1}{\rightsquigarrow} [q,\, l,\, V] \\
[q,\, l,\, \mathtt{match}\ \mathtt{in}_l\, V\ \mathtt{with}\ (x^A : M \mid y^B : N)] \overset{1}{\rightsquigarrow} [q,\, l,\, M\{V/x\}] \\
[q,\, l,\, \mathtt{match}\ \mathtt{in}_r\, V\ \mathtt{with}\ (x^A : M \mid y^B : N)] \overset{1}{\rightsquigarrow} [q,\, l,\, N\{V/y\}] \\
[q,\, l,\, \mathtt{letrec}\ f^{A \multimap B} x = M\ \mathtt{in}\ N] \overset{1}{\rightsquigarrow} [q,\, l,\, N\{(\lambda x^A.\mathtt{letrec}\ f^{A \multimap B} x = M\ \mathtt{in}\ M)/f\}] \\
[q,\, \emptyset,\, \mathtt{new}\, \mathtt{ff}] \overset{1}{\rightsquigarrow} [q \otimes |0\rangle,\, \{x \mapsto n+1\},\, x] \\
[q,\, \emptyset,\, \mathtt{new}\, \mathtt{tt}] \overset{1}{\rightsquigarrow} [q \otimes |1\rangle,\, \{x \mapsto n+1\},\, x] \\
[\alpha q_0 + \beta q_1,\, \{x \mapsto i\},\, \mathtt{meas}\, x] \overset{|\alpha|^2}{\rightsquigarrow} [r_0,\, \emptyset,\, \mathtt{ff}] \\
[\alpha q_0 + \beta q_1,\, \{x \mapsto i\},\, \mathtt{meas}\, x] \overset{|\beta|^2}{\rightsquigarrow} [r_1,\, \emptyset,\, \mathtt{tt}] \\
[q,\, l,\, \mathtt{U}(x_1 \otimes \cdots \otimes x_k)] \overset{1}{\rightsquigarrow} [r,\, l,\, (x_1 \otimes \cdots \otimes x_k)]
\end{array}
$$

Figure 2 Small-Step Axioms.

▶ **Lemma 3** (Type safety). *Let $C = [q, l, M]$ be a closed quantum closure. Then either M is a value or the total probability of all one-step reductions from C is* 1. ◀

By Lemma 3 we see that the reduction semantics induces a Markov chain $(Cl, \to)$, where Cl is the set of all closed quantum closures and $\to \subseteq Cl \times \mathcal{D}(Cl)$ is the transition relation with $C \to \mu$ satisfying $\mu(D) = p$ iff $C \overset{p}{\rightsquigarrow} D$ for some $p > 0$. Here $\mathcal{D}(Cl)$ stands for all probability subdistributions over Cl and μ is a full distribution over all successor quantum closures of C.

For any $\mu = \sum_i p_i \cdot [q_i, l_i, M_i]$, let $env(\mu) = \sum_i p_i \cdot tr_{fqv(M)} q_i q_i^\dagger$ be the reduced quantum state of the qubits not referred to by M. In particular, if each $[q_i, l_i, M_i]$ in the support of μ is a total quantum closure, we then have $env(\mu) = |\mu|$.

In order to investigate the long-term behaviour of a Markov chain, we introduce the notion of extreme derivative from [7]. We first need to lift the relation $\to$ to be a transition relation between subdistributions: $\mu \to \nu$ if $\nu = \sum_{C \in \lceil \mu \rceil} \mu(C) \cdot \mu_C$ and $C \to \mu_C$ for each C in $\lceil \mu \rceil$, the support of the subdistribution μ.

▶ **Definition 4** (Extreme derivative). Suppose we have subdistributions μ, $\mu_n^{\to}$, $\mu_n^{\times}$ for $n \geq 0$ with the following properties:

$$\mu = \mu_0^{\to} + \mu_0^{\times}; \qquad \forall k \geq 0.\ \mu_k^{\to} \to \mu_{k+1}^{\to} + \mu_{k+1}^{\times};$$

and each $\mu_k^{\times}$ is stable in the sense that $C \not\to$, for all $C \in \lceil \mu_k^{\times} \rceil$. Then we call $\rho := \sum_{k=0}^{\infty} \mu_k^{\times}$ an *extreme derivative* of μ, and write $\mu \Rightarrow \rho$.

Let C be a quantum closure in the Markov chain $(Cl, \to)$. The extreme derivative of the point distribution on C that assigns probability 1 to C, written $\overline{C}$, is unique, and we use it for the denotation of C, indicated as $[\![C]\!]$. So we always have $\overline{C} \Rightarrow [\![C]\!]$. Note that in the presence of divergence $[\![C]\!]$ may be a proper subdistribution.

Extreme derivatives can also be defined by a *big-step* semantics given by using a binary relation $\Downarrow$ between quantum closures and value distributions. Some of the rules of it can be found in Figure 3 (the others are similar). In the rules, ε stands for the empty subdistribution, and the notation $|\mu|$ stands for the size of the subdistribution μ, i.e. $\sum_{C \in \lceil \mu \rceil} \mu(C)$. Finally,

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\frac{}{C \Downarrow \varepsilon} \qquad \frac{}{[q,l,V] \Downarrow \overline{[q,l,V]}} \qquad \frac{[q,l,M] \Downarrow \sum_{k\in K} p_k \cdot \overline{[r_k,i_k,L_k]} \quad \{[r_k,i_k,N] \Downarrow \mu_k\}_{k\in K}}{[q,l,M;N] \Downarrow \sum_{k\in K} p_k \mu_k}$$

$$\frac{[q,l,M] \Downarrow \sum_{k\in K} p_k \cdot \overline{[r_k,i_k,V_k]} \quad \{[r_k,i_k,N] \Downarrow \mu_k\}_{k\in K}}{[q,l,M \otimes N] \Downarrow \sum_{k\in K} p_k (V_k \otimes \mu_k)} \qquad \frac{[q,l,M] \Downarrow \sum_{k\in K} p_k \cdot \overline{[r_k,i_k,V_k]}}{[q,l,\mathtt{in}_l\, M] \Downarrow \sum_{k\in K} p_k (\mathtt{in}_l\, \mu_k)}$$

$$\frac{[q,l,M] \Downarrow \sum_{k\in K} p_k \cdot \overline{[r_k,i_k,V_k \otimes W_k]} \quad \{[r_k,i_k,(N\{V_k/x, W_k/y\})] \Downarrow \mu_k\}_{k\in K}}{[q,l,\mathtt{let}\ x^A \otimes y^B = M\ \mathtt{in}\ N] \Downarrow \sum_{k\in K} p_k \mu_k}$$

$$\frac{\begin{array}{c}[q,l,M] \Downarrow \sum_{k\in K} p_k \cdot \overline{[r_k,i_k,V_k]} \\ \{[r_k,i_k,(N\{W_k/x\}] \Downarrow \rho_k\}_{V_k=\mathtt{in}_l\, W_k} \quad \{[r_k,i_k,(L\{W_k/x\}] \Downarrow \xi_k\}_{V_k=\mathtt{in}_r\, W_k}\end{array}}{[q,l,\mathtt{match}\ M\ \mathtt{with}\ (x^A : N \mid y^B : L)] \Downarrow \sum_{V_k=\mathtt{in}_l\, W_k} p_k \rho_k + \sum_{V_k=\mathtt{in}_r\, W_k} p_k \xi_k}$$

Figure 3 Big-Step Semantics Rules — Selection.

term constructors are, with abuse of notation, applied to quantum closures and subdistributions in the natural way, e.g., $\mathtt{in}_l\,(\sum_{k\in K} p_k \cdot \overline{[q,l,M]})$ stands for $\sum_{k\in K} p_k \cdot \overline{[q,l,\mathtt{in}_l\, M]}$.

▶ **Lemma 5.** $[\![C]\!] = \sup\{\mu \mid C \Downarrow \mu\}$, *where the supremum of subdistributions are computed component-wisely.* ◀

Following [8] we would like to give an alternative characterisation of linear contextual equivalence. Intuitively, as usual, a context is a term with a unique hole, and a linear context is a context where programs under examination will be evaluated and used *exactly once*. We are interested in *closing* contexts.

▶ **Definition 6.** A *linear context* (or simply a *context*) is a term with a hole, written $\mathcal{C}(\Delta; A)$, such that $\mathcal{C}[M]$ is a closed program whenever the hole is filled in by a term M, where $\Delta \rhd M : A$, and the hole lies in linear position.

Following [2], we require that the observable behaviour of a quantum closure C is its probability of convergence $|[\![C]\!]|$.

▶ **Definition 7.** The *linear contextual preorder* is the typed relation $\sqsubseteq$ defined as follows: $\Delta \rhd M \sqsubseteq N : A$ if for every linear context $\mathcal{C}$, quantum state q and linking function l such that $\emptyset \rhd \mathcal{C}(\Delta; A) : B$, and both $[q,l,\mathcal{C}[M]]$ and $[q,l,\mathcal{C}[N]]$ are total quantum closures, it holds that $|[\![[q,l,\mathcal{C}[M]]]\!]| \leq |[\![[q,l,\mathcal{C}[N]]]\!]|$. *Linear contextual equivalence* is the typed relation $\simeq$ by letting $\Delta \rhd M \simeq N : A$ just when $\Delta \rhd M \sqsubseteq N : A$ and $\Delta \rhd N \sqsubseteq M : A$.

2.3 A Probabilistic Labelled Transition System

In [14], Gordon defines explicitly a labelled transition system in order to illustrate the bisimulation technique in PCF. We follow this idea to define a *probabilistic* labelled transition system for the quantum λ-calculus, upon which we can define probabilistic bisimulations.

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\frac{}{[q,l,x_1\otimes\cdots\otimes x_n]\xrightarrow{\mathtt{i\,U}}[q,l,\mathtt{U}(x_1\otimes\cdots\otimes x_n)]}\qquad\frac{}{[q,l,x]\xrightarrow{\mathtt{i\,meas}}[q,l,\mathtt{meas}\,x]}$$

$$\frac{}{[q,\emptyset,\mathtt{skip}]\xrightarrow{\mathtt{skip}}[q,\emptyset,\mathbf{\Omega}]}\qquad\frac{\emptyset\rhd V:A\multimap B\quad \emptyset\rhd W:A}{[q,\,l,\,V]\xrightarrow{@[r,W]}[q,\,l\uplus r,\,VW]}$$

$$\frac{\emptyset\rhd\mathtt{in}_l\,V:A\oplus B\quad x:A\rhd M:C}{[q,l,\mathtt{in}_l\,V]\xrightarrow{\mathtt{l}\,[r,M]}[q,l\uplus r,M\{V/x\}]}\qquad\frac{\emptyset\rhd\mathtt{in}_r\,V:A\oplus B\quad y:B\rhd M:C}{[q,l,\mathtt{in}_r\,V]\xrightarrow{\mathtt{r}\,[r,M]}[q,l\uplus r,M\{V/y\}]}$$

$$\frac{\emptyset\rhd V\otimes W:A\otimes B\qquad x:A,y:B\rhd M:C}{[q,l,V\otimes W]\xrightarrow{\otimes[r,M]}[l\uplus r,M\{V/x,W/y\}]}\qquad\frac{}{C\xrightarrow{eval}[\![C]\!]}$$

Figure 4 Labelled Transition Rules for Quantum Closures.

Transition rules are listed in Figure 4: we make the typing of terms explicit in the rules as the type system plays an important role in defining the operational semantics of typed terms. A transition takes the form $C\xrightarrow{a}\mu$, where C is a quantum closure, μ is a subdistribution over quantum closures, and a is an action. If μ is a point distribution $\overline{D}$, we simply write the transition as $C\xrightarrow{a}D$. Note that non-total quantum closures are needed here to specify the operational semantics: we cannot work only with total quantum closures due to entanglement. For example, we should allow for a non-total quantum closure like $[\frac{|00\rangle+|11\rangle}{\sqrt{2}},\{x\mapsto 1\},x]$, where the quantum variable x refers to the first qubit of an entangled quantum state.

The last rule in Figure 4 says that term reductions are considered as internal transitions that are abstracted away; external transitions are labelled by *actions*. Intuitively, external transitions represent the way terms interact with environments (or contexts). For instance, a λ-abstraction can "consume" (application of itself to) a term, which is supplied by the environment as an argument, and forms a β-reduction. The rule for `skip` says that what it can provide to the environment is the value of itself, and after that it cannot provide any information, hence no external transitions can occur any more. We represent this by a transition, labelled by the value of the constant, into a non-terminating program $\mathbf{\Omega}$ of appropriate type.

The set of quantum closures Cl together with the transition rules in Figure 4 yields a probabilistic labelled transition system (pLTS). It is in fact a reactive system in the sense that if $C\xrightarrow{a}\mu_1$ and $C\xrightarrow{a}\mu_2$ then $\mu_1=\mu_2$ for all $C\in Cl$, that is no two outgoing transitions leaving a quantum closure are labelled by the same action. Below we recall Larsen and Skou's probabilistic bisimulation [22]. We first review a way of lifting a binary relation $\mathcal{R}$ over a set S to be a relation $\mathcal{R}^\dagger$ over the set of subdistributions on S given in [9][1].

▶ **Definition 8.** Let S,T be two countable sets and $\mathcal{R}\subseteq S\times T$ be a binary relation. The lifted relation $\mathcal{R}^\dagger\subseteq\mathcal{D}(S)\times\mathcal{D}(T)$ is defined by letting $\mu\;\mathcal{R}^\dagger\;\nu$ iff $\mu(X)\leq\nu(\mathcal{R}(X))$ for all $X\subseteq S$. Here we write $\mathcal{R}(X)$ for the set $\{t\in T\mid\exists s\in X.\ s\;\mathcal{R}\;t\}$ and $\mu(X)$ is the accumulation probability $\sum_{s\in X}\mu(s)$.

The definition by Larsen and Skou, when instantiated on the pLTS of closed quantum closures, looks as follows:

[1] There are several different but equivalent formulations of the lifting operation; see e.g. [27, 6].

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

▶ **Definition 9.** A *probabilistic simulation* is a preorder $\mathcal{R}$ on closed quantum closures such that whenever $(C, D) \in\mathcal{R}$ we have that:

- $env(C) = env(D)$;
- $[\![C]\!] \ \mathcal{R}^{\dagger} \ [\![D]\!]$;
- if C, D are values then $C \xrightarrow{a} \mu$ implies $D \xrightarrow{a} \nu$ with $\mu \ \mathcal{R}^{\dagger} \ \nu$.

A *probabilistic bisimulation* is a relation $\mathcal{R}$ such that both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are probabilistic simulations. Let $\preceq$ and $\sim_s$ be the largest probabilistic simulation and bisimulation, called similarity and bisimilarity, respectively. Bisimilarity and similarity are relations on closed quantum closures, but can be generalized to open closures as follows:

Suppose $\Delta \rhd M, N : A$. We write $\emptyset \rhd M \preceq N : A$ if $[q, l, M] \preceq [q, l, N]$, and $\Delta \rhd M \sim_s N : A$ if $[q, l, M] \sim_s [q, l, N]$, for any q and l such that $[q, l, M]$ and $[q, l, N]$ are both typable quantum closures.

For reactive pLTSs, the kernel of probabilistic similarity is probabilistic bisimilarity. That is, $\sim_s \, = \, \preceq \cap \preceq^{-1}$. This of course also applies to the specific pLTS we are working with. The probabilistic bisimilarity defined above is a binary relation between states, and thus sometimes called state-based bisimilarity. Alternatively, it is possible to directly define a (sub)distribution-based bisimilarity by comparing actions emitted from subdistributions. In order to do so, we first define a transition relation between subdistributions.

▶ **Definition 10.** We write $\mu \xrightarrow{a} \rho$ if $\rho = \sum_{s \in \lceil \mu \rceil} \mu(s) \cdot \mu_s$, where μ_s is determined as follows:

- either $s \xrightarrow{a} \mu_s$
- or there is no ν with $s \xrightarrow{a} \nu$, and in this case we set $\mu_s = \varepsilon$.

Note that this is a weaker notion of transition relation between subdistributions, compared with that defined on Page 6. If $\mu \xrightarrow{a} \mu'$ then some (not necessarily all) states in the support of μ can perform action a. For example, consider the two states s_2 and s_3 in Figure 5. Since $s_2 \xrightarrow{c} \overline{s_4}$ and s_3 cannot perform action c, the distribution $\mu = \frac{1}{2}\overline{s_2} + \frac{1}{2}\overline{s_3}$ can make the transition $\mu \xrightarrow{c} \frac{1}{2}\overline{s_4}$ to reach the subdistribution $\frac{1}{2}\overline{s_4}$. Let μ be a subdistribution over a reactive pLTS. After performing any action, it can reach a unique subdistribution. That is, if $\mu \xrightarrow{a} \nu$ and $\mu \xrightarrow{a} \rho$ then $\nu = \rho$. Given a subdistribution μ, we denote by $[\![\mu]\!]$ the subdistribution $\sum_{C \in \lceil \mu \rceil} \mu(C) \cdot [\![C]\!]$.

▶ **Definition 11.** A *distribution-based bisimulation* is a binary relation $\mathcal{R}$ on subdistributions such that $\mu \ \mathcal{R} \ \nu$ implies

- $env(\mu) = env(\nu)$;
- $[\![\mu]\!] \ \mathcal{R} \ [\![\nu]\!]$;
- if μ and ν are value distributions and $\mu \xrightarrow{a} \rho$, then $\nu \xrightarrow{a} \xi$ for some ξ with $\rho \ \mathcal{R} \ \xi$, and vice-versa.

Let $\sim_d$ be the largest distribution-based bisimulation. Suppose $\emptyset \rhd M, N : A$. We write $\emptyset \rhd M \sim_d N : A$ if $[\![[q, l, M]]\!] \sim_d [\![[q, l, N]]\!]$ for any q and l such that $[q, l, M]$ and $[q, l, N]$ are quantum closures.

It is not difficult to see that $s \sim_s t$ implies $\overline{s} \sim_d \overline{t}$ but not the other way around, as witnessed by the following example.

▶ **Example 12.** Consider the two states s and t in Figure 5. We can construct the relation

$$\mathcal{R}= \{(\overline{s}, \overline{t}), \left(\overline{s_1}, \frac{1}{2}\overline{t_1} + \frac{1}{2}\overline{t_2}\right), \left(\frac{1}{2}\overline{s_2} + \frac{1}{2}\overline{s_3}, \frac{1}{2}\overline{t_3} + \frac{1}{2}\overline{t_4}\right), (\overline{s_2}, \overline{t_3}), (\overline{s_3}, \overline{t_4}), (\overline{s_4}, \overline{t_5}))\}$$

and check that $\mathcal{R}$ is a distribution-based bisimulation. Therefore, we have $\overline{s} \sim_d \overline{t}$. Note that the point distribution at state s_1 is related to the distribution $\frac{1}{2}\overline{t_1} + \frac{1}{2}\overline{t_2}$. However,

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

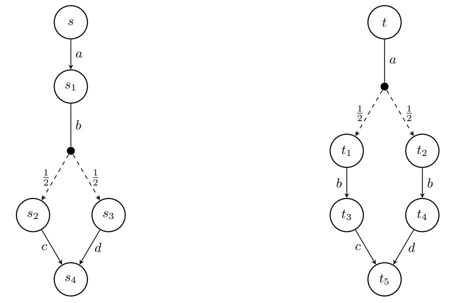

Figure 5 $s \not\sim_s t$.

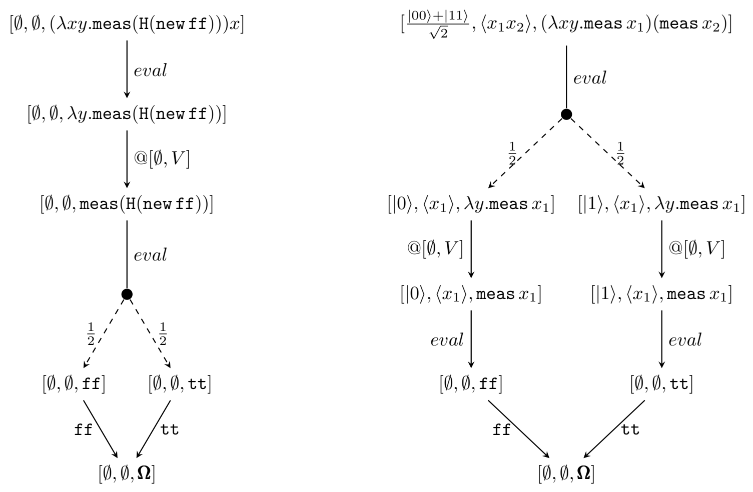

Figure 6 Two Quantum Closures not Related by $\sim_s$.

we have $s \not\sim_s t$ because after performing action a the state s evolves into the point distribution $\overline{s_1}$, and the only candidate transition from t to match this is $t \xrightarrow{a} \mu$ where $\mu = \frac{1}{2}\overline{t_1} + \frac{1}{2}\overline{t_2}$. But then the condition $\overline{s_1} \sim_s{}^{\dagger} \mu$ is invalid because there is no way to split s_1 into two different states such that they are bisimilar to t_1 and t_2 respectively. In the quantum λ-calculus this distinction between state-based and distribution-based bisimulations also exists. For example, the quantum closures $[\emptyset, \emptyset, (\lambda xy.\mathtt{meas}(\mathtt{H}(\mathtt{new}\,\mathtt{ff})))x]$ and $[\frac{|00\rangle+|11\rangle}{\sqrt{2}}, \langle x_1x_2\rangle, (\lambda xy.\mathtt{meas}\,x_1)(\mathtt{meas}\,x_2)]$ exhibit similar behaviour as states s and t, respectively, as depicted in Figure 6. ◀

3 Congruence

In this section we show that both $\sim_s$ and $\sim_d$ are congruence relations. The proof for $\sim_s$ is more complicated, so we take it as an example and give the details. The case for $\sim_d$ follows the same schema. The basic idea is to make use of Howe's method [19, 26], which requires to start from an initial relation $\mathcal{R}$, define a precongruence candidate $\mathcal{R}^H$, a precongruence relation by construction, and then to show the coincidence of that relation with the initial relation.

▶ **Definition 13.** Let $\mathcal{R}$ be a typed relation on quantum closures. Its *compatible refinement*

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY

$$\dfrac{\Delta \rhd c \quad c \in \{x, \mathtt{skip}, \mathtt{split}^A, \mathtt{new}, \mathtt{meas}, \mathtt{U}\}}{\Delta \rhd [q,l,c]\ \hat{\mathcal{R}}\ [q,l,c]} \qquad \dfrac{\Delta, x : A \rhd [q,l,M]\ \ \mathcal{R}\ \ [r,j,N]}{\Delta \rhd [q,l,(\lambda x^A.M)]\ \hat{\mathcal{R}}\ [r,j,(\lambda x^A.N)]}$$

$$\dfrac{!\Delta, \Delta' \rhd [q,l,M]\ \ \mathcal{R}\ \ [r,j,N] \qquad !\Delta, \Delta'' \rhd [q,i,L]\ \ \mathcal{R}\ \ [r,m,P]}{!\Delta, \Delta', \Delta'' \rhd [q, l \uplus i, ML]\ \hat{\mathcal{R}}\ [r, j \uplus m, NP]}$$

$$\dfrac{!\Delta, \Delta' \rhd [q,l,M]\ \ \mathcal{R}\ \ [r,j,N] \qquad !\Delta, \Delta'' \rhd [q,i,L]\ \ \mathcal{R}\ \ [r,m,P]}{!\Delta, \Delta', \Delta'' \rhd [q, l \uplus i, M \otimes L]\ \hat{\mathcal{R}}\ [q, j \uplus m, N \otimes P]}$$

$$\dfrac{!\Delta, \Delta' \rhd [q,l,M]\ \ \mathcal{R}\ \ [r,j,N]}{!\Delta, \Delta' \rhd [q,l,\mathtt{in}_l\ M]\ \hat{\mathcal{R}}\ [r,j,\mathtt{in}_l\ N]} \qquad \dfrac{!\Delta, \Delta' \rhd [q,l,M]\ \ \mathcal{R}\ \ [r,j,N]}{!\Delta, \Delta' \rhd [q,l,\mathtt{in}_r\ M]\ \hat{\mathcal{R}}\ [r,j,\mathtt{in}_r\ N]}$$

Figure 7 Compatible Refinement Rules — Selection.

$\hat{\mathcal{R}}$ is defined by some natural rules, a selection of which is in Figure 7. A relation $\mathcal{R}$ is a *precongruence* iff it contains its own compatible refinement, that is $\hat{\mathcal{R}} \subseteq \mathcal{R}$. Let a *congruence* be an equivalence relation that is a precongruence.

Let $\mathcal{R}$ be a typed relation on quantum closures. The typed relation $\mathcal{R}^H$ is defined by the rules in Figure 8. Note that if $\mathcal{R}$ is reflexive then $\mathcal{R} \subseteq \mathcal{R}^H$, and $\mathcal{R}^H$ is a precongruence. Therefore, in order to show that $\mathcal{R}$ is a precongruence (or congruence if $\mathcal{R}$ is also symmetric), it suffices to establish $\mathcal{R}^H \subseteq \mathcal{R}$ because we then have the coincidence of $\mathcal{R}$ with $\mathcal{R}^H$. In order to show $(\sim_s)^H \subseteq \sim_s$, we need the following two technical lemmas.

▶ **Lemma 14.** *If $\emptyset \rhd [q,l,M] \preceq^H [r,j,N]$ then $[\![[q,l,M]]\!]\ (\preceq^H)^\dagger\ [\![[r,j,N]]\!]$.* ◀

▶ **Lemma 15.** *If $\emptyset \rhd [q,l,V] \preceq^H [r,j,W]$ then we have that $[q,l,V] \xrightarrow{a} \mu$ implies $[r,j,W] \xrightarrow{a} \nu$ and $\mu\ (\preceq^H)^\dagger\ \nu$.* ◀

Consequently, we can establish the coincidence of $\preceq$ with $\preceq^H$, from which it is easy to show that $\sim_s$ is a congruence. Similar arguments apply to $\sim_d$.

▶ **Theorem 16.** *Both $\sim_s$ and $\sim_d$ are included in $\simeq$.* ◀

4 Completeness of Distribution-Based Bisimilarity

In this section we show that distribution-based bisimilarity is complete for linear contextual equivalence. The basic idea is to first characterise bisimilarity by a very simple testing framework. Let $\mathcal{T}$ be the set of tests of the two forms: ω and $a \cdot \mathtt{t}$, where ω is used to indicate success and a ranges over the set of all possible labels in the transition rules in Figure 4. In other words, the testing language is given by the grammar: $\mathtt{t}\ ::=\ \omega\ \big|\ a \cdot \mathtt{t}$.

Below we define the function Pr that calculates the probability of passing a test for a distribution of states in a reactive pLTS.

$$\begin{array}{rcl} Pr(\mu, \omega) & = & |\mu| \\ Pr(\mu, a \cdot \mathtt{t}) & = & Pr(\rho, \mathtt{t}) \text{ where } \mu \xrightarrow{a} \rho \end{array}$$

If μ is a point distribution $\overline{s}$, we will write $Pr(s, \mathtt{t})$ for $Pr(\mu, \mathtt{t})$. We define a testing equivalence $=^{\mathcal{T}}$ by letting $\mu =^{\mathcal{T}} \nu$ iff $\forall \mathtt{t} \in \mathcal{T} : Pr(\mu, \mathtt{t}) = Pr(\nu, \mathtt{t})$. It turns out that the tests in $\mathcal{T}$ are sufficient to characterise $\sim_d$ as far as reactive pLTSs are concerned.

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\frac{\Delta, x : A \rhd [q,l,M]\ \mathcal{R}^H\ [r,j,N] \quad \Delta \rhd [r,j,\lambda x^A.N]\ \ \mathcal{R}\ \ [p,i,L]}{\Delta \rhd [q,l,\lambda x^A.M]\ \mathcal{R}^H\ [p,i,L]}$$

$$\frac{\begin{array}{c} !\Delta,\Delta' \rhd [q,l,M]\ \mathcal{R}^H\ [r,j,N] \\ !\Delta,\Delta'' \rhd [q,i,L]\ \mathcal{R}^H\ [r,m,P] \\ !\Delta,\Delta',\Delta'' \rhd [r,j \uplus m, NP]\ \ \mathcal{R}\ \ [s,n,Q] \end{array}}{!\Delta,\Delta',\Delta'' \rhd [q,l \uplus i, ML]\ \mathcal{R}^H\ [s,n,Q]}$$

$$\frac{\begin{array}{c} !\Delta,\Delta' \rhd [q,l,M]\ \mathcal{R}^H\ [r,j,N] \\ !\Delta,\Delta'' \rhd [q,i,L]\ \mathcal{R}^H\ [r,m,P] \\ !\Delta,\Delta',\Delta'' \rhd [r,j \uplus m, N \otimes P]\ \ \mathcal{R}\ \ [s,n,Q] \end{array}}{!\Delta,\Delta',\Delta'' \rhd [q,l \uplus i, M \otimes L]\ \mathcal{R}^H\ [s,n,Q]}$$

$$\frac{!\Delta,\Delta' \rhd [q,l,M]\ \mathcal{R}^H\ [r,j,N] \quad !\Delta,\Delta' \rhd [r,j,\mathtt{in}_l\ N]\ \mathcal{R}\ [p,i,L]}{!\Delta,\Delta' \rhd [q,l,\mathtt{in}_l\ M]\ \mathcal{R}^H\ [p,i,L]}$$

$$\frac{!\Delta,\Delta' \rhd [q,l,M]\ \mathcal{R}^H\ [r,j,N] \quad !\Delta,\Delta' \rhd [r,j,\mathtt{in}_r\ N]\ \mathcal{R}\ [p,i,L]}{!\Delta,\Delta' \rhd [q,l,\mathtt{in}_r\ M]\ \mathcal{R}^H\ [p,i,L]}$$

Figure 8 Howe's Construction Rules — Selection.

▶ **Theorem 17.** *Let μ and ν be two distributions in a reactive pLTS. Then $\mu \sim_d \nu$ if and only if $\mu =^{\mathcal{T}} \nu$.* ◀

Following [2], we turn each test into a corresponding context. That is, for a given test $\mathtt{t}$ and a given type A, there exists a linear context $\mathcal{C}^A_{\mathtt{t}}$ such that for all terms M of type A, the success probability of $\mathtt{t}$ applied to any total quantum closure $[q,l,M]$ is exactly the convergence probability of $[q,l,\mathcal{C}^A_{\mathtt{t}}[M]]$.

▶ **Lemma 18.** *Let A be a type and $\mathtt{t}$ a test. There is a context $\mathcal{C}^A_{\mathtt{t}}$ such that $\emptyset \rhd \mathcal{C}^A_{\mathtt{t}}(\emptyset; A) : \mathsf{bit}$ and for every M with $\emptyset \rhd M : A$, we have $Pr([q,l,M],\mathtt{t}) = |[\![[q,l,\mathcal{C}^A_{\mathtt{t}}[M]]]\!]|$, where $[q,l,M]$ and $[q,l,\mathcal{C}^A_{\mathtt{t}}[M]]$ are quantum closures for any q and l.* ◀

As a consequence of the previous lemma, we can show that the distribution-based bisimilarity $\sim_d$ is complete with respect to the linear contextual equivalence $\simeq$.

▶ **Theorem 19** (Full Abstraction). *$\simeq$ coincides with $\sim_d$.* ◀

5 Concluding Remarks

We have presented two notions of bisimilarity for reasoning about equivalence of higher-order quantum programs in linear contexts, based on an appropriate labelled transition system for specifying the operational behaviour of programs. Both bisimilarities are sound with respect to the linear contextual equivalence, but only the distribution-based one turns out to be complete. Since linear resources are widely used in quantum computation, we believe that linear contextual equivalence will be a useful notion of behavioural equivalence for quantum programs. The coinductive proof techniques developed in the current work can help to reason about quantum programs.

In the future, it would be interesting to seek a denotational model fully abstract with respect to the linear contextual equivalence. As recently shown in [35], Fock spaces can be

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

useful to interpret quantum computation and they are close to the categorical semantics studied in [24], so it seems promising to start from there.

References

1 S. Abramsky. The lazy lambda calculus. In *Research Topics in Functional Programming*, pages 65–116. Addison-Wesley, 1999.

2 R. Crubillé and U. Dal Lago. On probabilistic applicative bisimulation and call-by-value λ-calculi. In *ESOP'14*, volume 8410 of *LNCS*, pages 209–228. Springer, 2014.

3 U. Dal Lago and A. Rioli. Applicative bisimulation and quantum λ-calculi (long version). *CoRR*, abs/1506.06661, 2015.

4 U. Dal Lago, D. Sangiorgi, and M. Alberti. On coinductive equivalences for higher-order probabilistic functional programs. In *POPL'14*, pages 297–308. ACM, 2014.

5 T. A. Davidson. *Formal verification techniques using quantum process calculus.* PhD thesis, University of Warwick, 2011.

6 Y. Deng. *Semantics of Probabilistic Processes: An Operational Approach.* Springer, 2014.

7 Y. Deng, R. van Glabbeek, M. Hennessy, and C. Morgan. Testing finitary probabilistic processes (extended abstract). In *CONCUR'09*, volume 5710 of *LNCS*, pages 274–288. Springer, 2009.

8 Y. Deng and Y. Zhang. Program equivalence in linear contexts. *Theoretical Computer Science*, 585:71–90, 2015.

9 J. Desharnais. *Labelled Markov Processes.* PhD thesis, McGill University, 1999.

10 Y. Feng, Y. Deng, and M. Ying. Symbolic bisimulation for quantum processes. *ACM Transactions on Computational Logic*, 15(2):1–32, 2014.

11 Y. Feng, R. Duan, Z. Ji, and M. Ying. Probabilistic bisimulations for quantum processes. *Information and Computation*, 205(11):1608–1639, 2007.

12 Y. Feng, R. Duan, and M. Ying. Bisimulation for quantum processes. *ACM Transactions on Programming Languages and Systems*, 34(4):17, 2012.

13 Y. Feng and L. Zhang. When equivalence and bisimulation join forces in probabilistic automata. In *FM'14*, volume 8442 of *LNCS*, pages 247–262. Springer, 2014.

14 A. M. Gordon. A tutorial on co-induction and functional programming. In *Glasgow Workshop on Functional Programming*, pages 78–95. Springer, 1995.

15 A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron. Quipper: a scalable quantum programming language. In *PLDI'13*, pages 333–342. ACM, 2013.

16 L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC'96*, pages 212–219. ACM, 1996.

17 L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79:325–328, 1997.

18 H. Hermanns, J. Krcál, and J. Kretínský. Probabilistic bisimulation: Naturally on distributions. In *CONCUR'14*, volume 8704 of *LNCS*, pages 249–265. Springer, 2014.

19 D. J. Howe. Proving congruence of bisimulation in functional programming languages. *Information and Computation*, 124(2):103–112, 1996.

20 E. Knill. Conventions for quantum pseudocode. Technical Report LAUR-96-2724, Los Alamos National Laboratory, 1996.

21 M. Lalire. Relations among quantum processes: bisimilarity and congruence. *Mathematical Structures in Computer Science*, 16(3):407–428, 2006.

22 K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

23 J. H. Morris. *Lambda Calculus Models of Programming Languages.* PhD thesis, MIT, 1969.

24 M. Pagani, P. Selinger, and B. Valiron. Applying quantitative semantics to higher-order quantum computing. In *POPL'14*, pages 647–658. ACM, 2014.

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

25 A. M. Pitts. Operationally-based theories of program equivalence. In *Semantics and Logics of Computation*, pages 241–298. Cambridge University Press, 1997.

26 A. M. Pitts. Howe's method for higher-order languages. In *Advanced Topics in Bisimulation and Coinduction*, pages 197–232. Cambridge University Press, 2011.

27 J. Sack and L. Zhang. A general framework for probabilistic characterizing formulae. In *VMCAI'12*, volume 7148 of *LNCS*, pages 396–411. Springer, 2012.

28 P. Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.

29 P. Selinger and B. Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 16(3):527–552, 2006.

30 P. Selinger and B. Valiron. A linear-non-linear model for a computational call-by-value lambda calculus (extended abstract). In *FOSSACS'08*, volume 4962 of *LNCS*, pages 81–96. Springer, 2008.

31 P. Selinger and B. Valiron. On a fully abstract model for a quantum linear functional language (extended abstract). *Electronic Notes in Theoretical Computer Science*, 210:123–137, 2008.

32 P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS'94*, pages 124–134. IEEE Computer Society, 1994.

33 F. van Breugel, M. W. Mislove, J. Ouaknine, and J. Worrell. Domain theory, testing and simulation for labelled markov processes. *Theoretical Computer Science*, 333(1-2):171–197, 2005.

34 D. Wecker and K. M. Svore. LIQUi|>: A software design architecture and domain-specific language for quantum computing. *CoRR*, abs/1402.4467, 2014.

35 M. Ying. Quantum recursion and second quantisation: Basic ideas and examples. *CoRR*, abs/1405.4443, 2014.

© Yuxin Deng, Yuan Feng, and Ugo Dal Lago;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany