# A Lambda Calculus for Density Matrices with Classical and Probabilistic Controls

Alejandro Díaz-Caro(✉)

Universidad Nacional de Quilmes & CONICET,
Roque Sáenz Peña 352, B1876BXD Bernal, Buenos Aires, Argentina
alejandro.diaz-caro@unq.edu.ar

**Abstract.** In this paper we present two flavors of a quantum extension to the lambda calculus. The first one, $\lambda_\rho$, follows the approach of classical control/quantum data, where the quantum data is represented by density matrices. We provide an interpretation for programs as density matrices and functions upon them. The second one, $\lambda_\rho^\circ$, takes advantage of the density matrices presentation in order to follow the mixed trace of programs in a kind of generalised density matrix. Such a control can be seen as a weaker form of the quantum control and data approach.

**Keywords:** Lambda calculus · Quantum computing · Density matrices · Classical control

## 1 Introduction

In the last decade several quantum extensions to lambda calculus have been investigated,e.g. [5,6,11,19,22,23,31]. In all of those approaches, the language chosen to represent the quantum state are vectors in a Hilbert space. However, an alternative formulation of quantum mechanics can be made using density matrices. Density matrices provide a way to describe a quantum system in which the state is not fully known. More precisely, density matrices describe quantum systems in a mixed state, that is, a statistical set of several quantum states. All the postulates of quantum mechanics can be described in such a formalism, and hence, also quantum computing can be done using density matrices.

The first postulate states that a quantum system can be fully described by a density matrix $\rho$, which is a positive operator with trace ($\mathsf{tr}$) one. If a system is in state $\rho_i$ with probability $p_i$, then the density matrix of the system is $\sum_i p_i\rho_i$. The second postulate states that the evolution of a quantum system $\rho$ is described with a unitary operator $U$ by $U\rho U^\dagger$, where $U^\dagger$ is the adjoint operator of $U$. The third postulate states that the measurement is described by a set of measurement operators $\{\pi_i\}_i$ with $\sum_i \pi_i^\dagger\pi_i = \mathsf{I}$, so that the output of the measurement is $i$, with probability $\mathsf{tr}(\pi_i^\dagger\pi_i\rho)$, leaving the sate of the system

A. Díaz-Caro—Supported by projects STIC-AmSud 16STIC05 FoQCoSS, PICT 2015-1208 and the Laboratoire International Associé "INFINIS".


B.-Y.E. Chang (Ed.): APLAS 2017, LNCS 10695, pp. 448–467, 2017.
https://doi.org/10.1007/978-3-319-71237-6_22

as $\frac{\pi_i \rho \pi_i^\dagger}{\mathsf{tr}(\pi_i^\dagger \pi_i \rho)}$. The fourth postulate states that from two systems $\rho$ and $\rho'$, the composed one can be described by the tensor product of those $\rho \otimes \rho'$.

Naturally, if we want to use the output of a measurement as a condition in the classical control, we need to know that output. However, density matrices can still be used as a way to compare processes before running them. For example the process of tossing a coin, and according to its result, applying Z or not to a balanced superposition, and the process of tossing a coin and not looking at its result, may look quite different in most quantum programming languages. Yet both processes output the same density matrix, and so they are indistinguishable.

In [20], Selinger introduced a language of quantum flow charts, and an interpretation of his language into a CPO of density matrices. After this paper, the language of density matrices has been widely used in quantum programming, e.g. [9,14,15,25,28]. Indeed, the book "Foundations of Quantum Programming" [27] is entirely written in the language of density matrices. Yet, as far as we know, no lambda calculus for density matrix have been proposed.

Apart from the distinction of languages by how they treat the quantum states (vectors in a Hilbert space or density matrices), we also can distinguish the languages on how the control is considered: either quantumly or classically. The idea of quantum data/classical control stated by Selinger in [20] induced a quantum lambda calculus in this paradigm [22]. Later, this calculus was the base to construct the programming language Quipper [16], an embedded, scalable, functional programming language for quantum computing. The concept of quantum data/classical control declares that quantum computers will run in a specialized device attached to a classical computer, and it is the classical computer which will instruct the quantum computer what operations to perform over which qubits, and then read the classical result after a measurement. It is a direct consequence from the observation that quantum circuits are classical (i.e. one cannot superpose circuits or measure them). Several studies have been done under this paradigm, e.g. [2,16,19,22,31].

Dually to the quantum data/classical control paradigm, there is what we can call the quantum data and control paradigm. The idea is to provide a computational definition of the notions of vector space and bilinear functions. In the realm of quantum walks, quantum control is not uncommon (e.g. [1,3]). Also, several high-level languages on quantum control have been proposed in the past (e.g. [2,8,29,30]), however, up to now, no complete lambda-calculus with quantum control have been proposed. We benefit, though, from the long line of works in this direction [4–7,13].

In this paper, we propose a quantum extension to the lambda calculus, $\lambda_\rho$, in the quantum data/classical control paradigm, where the quantum data is given by density matrices, as first suggested by Selinger's interpretation of quantum flow charts [20]. Then, we propose a modification of such a calculus, called $\lambda_\rho^\circ$, in which we generalise the density matrices to the classical control: That is, after a measurement, we take all the possible outcomes in a kind of generalised density matrix of arbitrary terms. The control does not become quantum, since it is not possible to superpose programs in the quantum sense. However, we consider the density matrix of the mixed state of programs arising from a measurement. Therefore, this can be considered as a kind of probabilistic control, or even another way, perhaps weaker, of quantum control.

*Outline of the Paper.* In Sect. 2 we introduce the typed calculus $\lambda_\rho$, which manipulates density matrices, and we give two interpretations of the calculus. One where the terms are interpreted into a generalisation of mixed states, and another where the terms are interpreted into density matrices. Then we prove some properties of those interpretations. In Sect. 3 we introduce a modification of $\lambda_\rho$, called $\lambda_\rho^\circ$, where the output of a measurement produce a sum with all the possible outputs. We then extend the interpretation of $\lambda_\rho$ to accommodate $\lambda_\rho^\circ$, and prove its basic properties. In Sect. 4 we prove the Subject Reduction (Theorem 4.4) and Progress (Theorem 4.7) properties for both calculi. In Sect. 5 we give two interesting examples, in both calculi. Finally, in Sect. 6, we conclude and discuss some future work. A long version of this paper, with detailed proofs in a 10-pages appendix, has been submitted to the arXiv [10].

## 2 Classical-Control Calculus with Probabilistic Rewriting

### 2.1 Definitions

The grammar of terms, given in Table 1, have been divided in three categories.

1. Standard lambda calculus terms: Variables from a set $\mathsf{Vars}$, abstractions and applications.
2. The four postulates of quantum mechanics, with the measurement postulate restricted to measurements in the computational basis[1]: $\rho^n$ to represent the density matrix of a quantum system. $U^n t$ to describe its evolution. $\pi^n t$ to measure it. $t \otimes t$ to describe the density matrix of a composite system (that is, a non entangled system composed of two subsystems).
3. Two constructions for the classical control: a pair $(b^m, \rho^n)$, where $b^m$ is the output of a measurement in the computational basis and $\rho^n$ is the resulting density matrix, and the conditional $\mathsf{letcase}$ construction reading the output of the measurement.

The rewrite system, given in Table 2, is described by the relation $\longrightarrow_p$, which is a probabilistic relation where $p$ is the probability of occurrence. If $U^m$ is applied to $\rho^n$, with $m \leq n$, we write $\overline{U^m}$ for $U^m \otimes I^{n-m}$. Similarly, we write $\overline{\pi^m}$ when we apply this measurement operator to $\rho^n$ for $\{\pi_0 \otimes I^{n-m}, \dots, \pi_{2^m-1} \otimes I^{n-m}\}$. If the unitary $U^m$ needs to be applied, for example, to the last $m$ qubits of $\rho^n$ instead of the first $m$, we will need to use the unitary transformation $\overline{I^{n-m} \otimes U^m}$ instead. And if it is applied to the qubits $k$ to $k+m$, then, we can use $\overline{I^{k-1} \otimes U^m}$.

[1] A generalisation to any arbitrary measurement can be considered in a future, however, for the sake of simplicity in the classical control, we consider only measurements in the computational basis, which is a common practice in quantum lambda calculi [11,17,19,21,22,31].

**Table 1.** Grammar of terms of $\lambda_\rho$.

$$
\begin{aligned}
t := {} & x \mid \lambda x.t \mid tt && \text{(Standard lambda calculus)}\\
& \mid \rho^n \mid U^n t \mid \pi^n t \mid t \otimes t && \text{(Quantum postulates)}\\
& \mid (b^m, \rho^n) \mid \mathsf{letcase}\ x = r\ \mathsf{in}\ \{t, \dots, t\} && \text{(Classical control)}
\end{aligned}
$$

where:

- $n, m \in \mathbb{N}$, $m \leq n$.
- $\rho^n$ is a density matrix of $n$-qubits, that is, a positive $2^n \times 2^n$-matrix with trace 1.
- $b^m \in \mathbb{N}$, $0 \leq b^m < 2^m$.
- $\{t, \dots, t\}$ contains $2^m$ terms.
- $U^n$ is a unitary operator of dimension $2^n \times 2^n$, that is, a $2^n \times 2^n$-matrix such that $(U^n)^\dagger = (U^n)^{-1}$.
- $\pi^n = \{\pi_0, \dots, \pi_{2^n-1}\}$, describes a quantum measurement in the computational basis, where each $\pi_i$ is a projector operator of dimension $2^n$ projecting to one vector of the canonical base.

This rewrite system assumes that after a measurement, the result is known. However, since we are working with density matrices we could also provide an alternative rewrite system where after a measurement, the system turns into a mixed state. We left this possibility for Sect. 3.

The type system, including the grammar of types and the derivation rules, is given in Table 3. The type system is affine, so variables can be used at most once, forbiding from cloning a density matrix.

*Example 2.1.* The teleportation algorithm, while it is better described by pure states, can be expressed in the following way:

Let $\beta_{00} = \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)$. Then, the following term expresses the teleportation algorithm.

$$\lambda x.\mathsf{letcase}\ y = \pi^2(\mathsf{H}^1(\mathsf{Cnot}^2(x \otimes \beta_{00})))\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\}$$

where $\mathsf{Z}_3 = I \otimes I \otimes \mathsf{Z}^1$ and $\mathsf{X}_3 = I \otimes I \otimes \mathsf{X}^1$.

The type derivation is as follows.

$$
\dfrac{\dfrac{\dfrac{}{y:3 \vdash y:3}\,\mathsf{ax} \quad \dfrac{\dfrac{}{y:3 \vdash y:3}\,\mathsf{ax}}{y:3 \vdash \mathsf{Z}_3 y : 3}\,\mathsf{u} \quad \dfrac{\dfrac{}{y:3 \vdash y:3}\,\mathsf{ax}}{y:3 \vdash \mathsf{X}_3 y : 3}\,\mathsf{u} \quad \dfrac{\dfrac{\dfrac{}{y:3 \vdash y:3}\,\mathsf{ax}}{y:3 \vdash \mathsf{X}_3 y : 3}\,\mathsf{u}}{y:3 \vdash \mathsf{Z}_3\mathsf{X}_3 y : 3}\,\mathsf{u} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x:1 \vdash x:1}\,\mathsf{ax} \quad \dfrac{}{\vdash \beta_{00} : 2}\,\mathsf{ax}_\rho}{x:1 \vdash x \otimes \beta_{00} : 3}\,\otimes}{x:1 \vdash \mathsf{Cnot}^2(x \otimes \beta_{00}) : 3}\,\mathsf{u}}{x:1 \vdash \mathsf{H}^1(\mathsf{Cnot}^2(x \otimes \beta_{00})) : 3}\,\mathsf{u}}{x:1 \vdash \pi^2(\mathsf{H}^1(\mathsf{Cnot}^2(x \otimes \beta_{00}))) : (2,3)}\,\mathsf{m}}{x:1 \vdash \mathsf{letcase}\ y = \pi^2(\mathsf{H}^1(\mathsf{Cnot}^2(x \otimes \beta_{00})))\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X} y, \mathsf{Z}_3\mathsf{X}_3 y\} : 3}\,\mathsf{lc}}{\vdash \lambda x.\mathsf{letcase}\ y = \pi^2(\mathsf{H}^1(\mathsf{Cnot}^2(x \otimes \beta_{00})))\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\} : 1 \multimap 3}\,\multimap_i
$$

**Table 2.** Rewrite system for $\lambda_\rho$.

$$(\lambda x.t)r \longrightarrow_1 t[r/x]$$

$$U^m\rho^n \longrightarrow_1 {\rho'}^n \qquad \text{with } {\rho'}^n = \overline{U^m}\rho^n\overline{U^m}^\dagger$$

$$\pi^m\rho^n \longrightarrow_{p_i} (i, \rho_i^n) \qquad \text{with } \begin{cases} p_i = \mathsf{tr}(\overline{\pi_i}^\dagger\overline{\pi_i}\rho^n) \\ \rho_i^n = \frac{\overline{\pi_i}\rho^n\overline{\pi_i}^\dagger}{p_i} \end{cases}$$

$$\rho \otimes \rho' \longrightarrow_1 \rho'' \qquad \text{with } \rho'' = \rho \otimes \rho'$$

$$\mathsf{letcase}\ x = (b^m, \rho^n)\ \mathsf{in}\ \{t_0, \dots, t_{2^m-1}\} \longrightarrow_1 t_{b^m}[\rho^n/x]$$

$$\frac{t \longrightarrow_p r}{\lambda x.t \longrightarrow_p \lambda x.r} \qquad \frac{t \longrightarrow_p r}{ts \longrightarrow_p rs} \qquad \frac{t \longrightarrow_p r}{st \longrightarrow_p sr} \qquad \frac{t \longrightarrow_p r}{U^n t \longrightarrow_p U^n r}$$

$$\frac{t \longrightarrow_p r}{\pi^n t \longrightarrow_p \pi^n r} \qquad \frac{t \longrightarrow_p r}{t \otimes s \longrightarrow_p r \otimes s} \qquad \frac{t \longrightarrow_p r}{s \otimes t \longrightarrow_p s \otimes r}$$

$$\frac{t \longrightarrow_p r}{\mathsf{letcase}\ x = t\ \mathsf{in}\ \{s_0, \dots, s_n\} \longrightarrow_p \mathsf{letcase}\ x = r\ \mathsf{in}\ \{s_0, \dots, s_n\}}$$

**Table 3.** Type system for $\lambda_\rho$.

$$A := n \mid (m, n) \mid A \multimap A$$

where $m \leq n \in \mathbb{N}$.

$$\frac{}{\Gamma, x : A \vdash x : A}\ \mathsf{ax} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \multimap B}\ \multimap_i \qquad \frac{\Gamma \vdash t : A \multimap B \quad \Delta \vdash r : A}{\Gamma, \Delta \vdash tr : B}\ \multimap_e$$

$$\frac{}{\Gamma \vdash \rho^n : n}\ \mathsf{ax}_\rho \qquad \frac{\Gamma \vdash t : n}{\Gamma \vdash U^m t : n}\ \mathsf{u} \qquad \frac{\Gamma \vdash t : n}{\Gamma \vdash \pi^m t : (m, n)}\ \mathsf{m} \qquad \frac{\Gamma \vdash t : n \quad \Delta \vdash r : m}{\Gamma, \Delta \vdash t \otimes r : n + m}\ \otimes$$

$$\frac{}{\Gamma \vdash (b^m, \rho^n) : (m, n)}\ \mathsf{ax}_{\mathsf{am}} \qquad \frac{x : n \vdash t_0 : A \quad \dots \quad x : n \vdash t_{2^m-1} : A \quad \Gamma \vdash r : (m, n)}{\Gamma \vdash \mathsf{letcase}\ x = r\ \mathsf{in}\ \{t_0, \dots, t_{2^m-1}\} : A}\ \mathsf{lc}$$

## 2.2 Interpretation

We give two interpretations for terms. One, noted by $(\!|\cdot|\!)$, is the interpretation of terms into density matrices and functions upon them, and the other, noted by $[\![\cdot]\!]$, is a more fine-grained interpretation, interpreting terms into a generalisation of mixed states. In particular, we want $[\![\pi^n\rho^n]\!] = \{(\mathsf{tr}(\pi_i^\dagger\pi_i\rho^n), \frac{\pi_i\rho^n\pi_i^\dagger}{\mathsf{tr}(\pi_i^\dagger\pi_i\rho^n)})\}_i$, while $(\!|\pi^n\rho|\!) = \sum_i \pi_i\rho^n\pi_i^\dagger$. However, since the letcase construction needs also to distinguish each possible result of a measurement, we will carry those results in the interpretation $[\![\cdot]\!]$, making it a set of triplets instead of a set of tuples.

Let $\mathbb{N}^\varepsilon = \mathbb{N}_0 \cup \{\varepsilon\}$, so terms are interpreted into sets of triplets $(p, b, e)$ with $p \in \mathbb{R}_+^{\leq 1}$, representing the probability, $b \in \mathbb{N}^\varepsilon$, representing the output of a measurement if it occurred, and $e \in [\![A]\!]$ for some type $A$ and an interpretation $[\![\cdot]\!]$ on types yet to define. In addition, we consider that the sets $\{\dots, (p, b, e), (q, b, e), \dots\}$ and $\{\dots, (p+q, b, e), \dots\}$ are equal. Finally, we define the weight function as $\mathsf{w}(\{(p_i, b_i, e_i)\}_i) = \sum_i p_i$. We are interested in sets $S$ such that $\mathsf{w}(S) = 1$.

The interpretation of types is given in Table 4. $\mathcal{D}_n$ is the set of density matrices of $n$-qubits, that is $\mathcal{D}_n = \{\rho \mid \rho \in \mathcal{M}^+_{2^n \times 2^n}$ such that $\mathsf{tr}(\rho) = 1\}$, where $\mathcal{M}^+_{2^n \times 2^n}$ is the set of positive matrices of size $2^n \times 2^n$. $P(b, A)$ is the following property: $[A = \vec{B} \multimap (m, n) \implies b \neq \varepsilon]$, where $\vec{A} \multimap B$ is any of $B$, $A \multimap B$, $A_1 \multimap A_2 \multimap B, \dots, A_1 \multimap \cdots \multimap A_n \multimap B$. We also establish the convention that $P(\{(p_i, b_i, e_i)\}_i, A) = \bigwedge_i P(b_i, A)$. Finally, we write $\mathsf{trd}(S) = \{e \mid (p, b, e) \in S\}$.

**Table 4.** Interpretation of types

$$
\begin{aligned}
[\![n]\!] &= \mathcal{D}_n \\
[\![(m, n)]\!] &= \mathcal{D}_n \\
[\![A \multimap B]\!] &= \{f \mid \forall e \in [\![A]\!], \forall b \in \mathbb{N}^\varepsilon \text{ s.t. } P(b, A), \\
&\qquad \mathsf{trd}(f(b, e)) \subseteq [\![B]\!], \mathsf{w}(f(b, e)) = 1 \text{ and } P(f(b, e), B)\}
\end{aligned}
$$

Let $E = \bigcup_{A \in \mathsf{Types}} [\![A]\!]$. We denote by $\theta$ to a valuation $\mathsf{Vars} \to \mathbb{N}^\varepsilon \times E$. Then, we define the interpretation of terms with respect to a given valuation $\theta$ in Table 5.

**Definition 2.2.** *$\theta \vDash \Gamma$ if and only if, for all $x : A \in \Gamma$, $\theta(x) = (b, e)$ with $e \in [\![A]\!]$, and $P(b, A)$.*

Lemma 2.3 states that a term with type $(m, n)$ (or an arrow type ending in $(m, n)$), will be the result of a measurement, and hence, its interpretation will carry the results $b_i \neq \varepsilon$.

**Lemma 2.3.** *Let $\Gamma \vdash t : \vec{A} \multimap (m, n)$, $\theta \vDash \Gamma$, and $[\![t]\!]_\theta$ be well-defined. Then, $[\![t]\!]_\theta = \{(p_i, b_i, e_i)\}_i$ with $b_i \neq \varepsilon$ and $e_i \in [\![\vec{A} \multimap (m, n)]\!]_\theta$.*

*Proof.* By induction on the type derivation. □

Lemma 2.4 states that the interpretation of a typed term is well-defined.

**Lemma 2.4.** *If $\Gamma \vdash t : A$ and $\theta \vDash \Gamma$, then $\mathsf{w}([\![t]\!]_\theta) = 1$, and $\mathsf{trd}([\![t]\!]_\theta) \subseteq [\![A]\!]$.*

*Proof.* By induction on $t$. □

**Table 5.** Interpretation of terms

$$
\begin{aligned}
[\![x]\!]_\theta &= \{(1,b,e)\} \text{ where } \theta(x)=(b,e)\\
[\![\lambda x.t]\!]_\theta &= \{(1,\varepsilon,(\mathsf{b},\mathsf{e})\mapsto [\![t]\!]_{\theta,x=(\mathsf{b},\mathsf{e})})\}\\
[\![tr]\!]_\theta &= \{(p_iq_jh_{ijk},b''_{ijk},g_{ijk})\ |[\![r]\!]_\theta=\{(p_i,b_i,e_i)\}_i,\\
&\qquad [\![t]\!]_\theta=\{(q_j,b'_j,f_j)\}_j \text{ and}\\
&\qquad f_j(b_i,e_i)=\{(h_{ijk},b''_{ijk},g_{ijk})\}_k\}\\
[\![\rho^n]\!]_\theta &= \{(1,\varepsilon,\rho^n)\}\\
[\![U^n t]\!]_\theta &= \{(p_i,\varepsilon,\overline{U^n}\rho_i\overline{U^n}^\dagger)\mid [\![t]\!]_\theta=\{(p_i,b_i,\rho_i)\}_i\}\\
[\![\pi^m t]\!]_\theta &= \{(p_j\mathsf{tr}(\overline{\pi_i}^\dagger\overline{\pi_i}\rho_j),i,\frac{\overline{\pi_i}\rho_j\overline{\pi_i}^\dagger}{\mathsf{tr}(\overline{\pi_i}^\dagger\overline{\pi_i}\rho_j)})\mid [\![t]\!]_\theta=\{(p_j,b_j,\rho_j)\}_j\}\\
[\![t\otimes r]\!]_\theta &= \{(p_iq_j,\varepsilon,\rho_i\otimes\rho'_j)\ |[\![t]\!]_\theta=\{(p_i,b_i,\rho_i)\}_i \text{ and}\\
&\qquad [\![r]\!]_\theta=\{(q_j,b'_j,\rho'_j)\}_j\}\\
[\![(b^m,\rho^n)]\!]_\theta &= \{(1,b^m,\rho^n)\}\\
[\![\mathsf{letcase}\ x=r\ \mathsf{in}\ \{t_0,\dots,t_{2^m-1}\}]\!]_\theta &= \{(p_iq_{ij},b'_{ij},e_{ij})\ |\\
&\qquad [\![r]\!]_\theta=\{(p_i,b_i,\rho_i)\}_i \text{ and}\\
&\qquad [\![t_{b_i}]\!]_{\theta,x=(\varepsilon,\rho_i)}=\{(q_{ij},b'_{ij},e_{ij})\}_j\}
\end{aligned}
$$

Since the interpretation $[\![\cdot]\!]$ of a term is morally a mixed state, the interpretation $(\!|\cdot|\!)$, which should be the density matrix of such a state, is naturally defined using the interpretation $[\![\cdot]\!]$.

**Definition 2.5.** *Let $e\in[\![A]\!]$ for some $A$, $\theta$ a valuation, and $t$ be a term such that $[\![t]\!]_\theta = \{(p_i,b_i,e_i)\}_i$. We state the convention that $(\mathsf{b},\mathsf{e})\mapsto\sum_i p_ie_i = \sum_i p_i((\mathsf{b},\mathsf{e})\mapsto e_i)$. We define $[e]$ and $(\!|t|\!)_\theta$ by mutual recursion as follows:*

$$
\begin{aligned}
[\rho] &= \rho\\
\left[(\mathsf{b},\mathsf{e})\mapsto [\![t]\!]_{\theta,x=(\mathsf{b},\mathsf{e})}\right] &= (\mathsf{b},\mathsf{e})\mapsto (\!|t|\!)_{\theta,x=(\mathsf{b},\mathsf{e})}\\
(\!|t|\!)_\theta &= \sum_i p_i\,[e_i]
\end{aligned}
$$

**Lemma 2.6. (Substitution).** *Let $[\![r]\!]_\theta=\{(p_i,b_i,e_i)\}_i$, then*

$$(\!|t[r/x]|\!)_\theta = \sum_i p_i(\!|t|\!)_{\theta,x=(b_i,e_i)}$$

*Proof.* By induction on $t$. However, we enforce the hypothesis by also showing that if $[\![t]\!]_{\theta,x=(b_i,e_i)}=\{(q_{ij},b'_{ij},\rho_{ij})\}_j$, then $[\![t[r/x]]\!]_\theta=\{(p_iq_{ij},b'_{ij},\rho_{ij})\}_{ij}$. We use five auxiliary results (cf. appendix in [10] for more details). □

Theorem 2.7 shows how the interpretation $(\!|\cdot|\!)$ of a term relates to all its reducts.

**Theorem 2.7.** *If* $\Gamma \vdash t : A$, $\theta \vDash \Gamma$ *and* $t \longrightarrow_{p_i} r_i$, *with* $\sum_i p_i = 1$, *then* $(\!|t|\!)_\theta = \sum_i p_i (\!|r_i|\!)_\theta$.

*Proof.* By induction on the relation $\longrightarrow_p$. □

## 3 Probabilistic-Control Calculus with No-Probabilistic Rewriting

### 3.1 Definitions

In the previous sections we have presented an extension to lambda calculus to handle density matrices. The calculus could have been done using just vectors, because the output of a measurement is not given by the density matrix of the produced mixed state, instead each possible output is given with its probability. In this section, we give an alternative presentation, named $\lambda_\rho^\circ$, where we can make the most of the density matrices setting.

In Table 6 we give a modified grammar of terms for $\lambda_\rho^\circ$ in order to allow for linear combination of terms. We follow the grammar of the algebraic lambda-calculi [6,7,24].

**Table 6.** Grammar of terms of $\lambda_\rho^\circ$.

| | |
|---|---|
| $t := x \mid \lambda x.t \mid tt$ | (Standard lambda calculus) |
| $\mid \rho^n \mid U^n t \mid \pi^n t \mid t \otimes t$ | (Quantum postulates) |
| $\mid \sum_{i=1}^{n} p_i t_i \mid \mathsf{letcase}^\circ\ x = r\ \mathsf{in}\ \{t, \dots, t\}$ | (Probabilistic control) |

where $p_i \in (0, 1]$, $\sum_{i=1}^{n} p_i = 1$, and $\sum$ is considered modulo associativity and commutativity (cf. for example [6]).

The new rewrite system is given by the non-probabilistic relation $\rightsquigarrow$, described in Table 7. The measurement does not reduce, unless it is the parameter of a $\mathsf{letcase}^\circ$. Therefore, if only a measurement is needed, we can encode it as:

$$\mathsf{letcase}^\circ\ x = \pi^m \rho^n\ \mathsf{in}\ \{x, \dots, x\} \rightsquigarrow \sum_i p_i \rho_i^n \rightsquigarrow \rho'$$

where $\rho' = \sum_i \overline{\pi_i} \rho^n \overline{\pi_i}^\dagger$. The rationale is that in this version of the calculus, we can never look at the result of a measurement. It will always produce the density matrix of a mixed-state. As a consequence, the $\mathsf{letcase}^\circ$ constructor rewrites to a sum of terms.

The type system for $\lambda_\rho^\circ$, including the grammar of types and the derivation rules, is given in Table 8. The only difference with the type system of $\lambda_\rho$

**Table 7.** Rewrite system of $\lambda_\rho^\circ$.

$$(\lambda x.t)r \rightsquigarrow t[r/x]$$

$$\mathsf{letcase}^\circ\ x = \pi^m \rho^n \ \mathsf{in}\ \{t_0, \dots, t_{2^m} - 1\} \rightsquigarrow \sum_i p_i t_i[\rho_i^n/x] \quad \text{with} \begin{cases} \rho_i^n = \frac{\overline{\pi_i}\rho^n\overline{\pi_i}^\dagger}{p_i} \\ p_i = \mathsf{tr}(\overline{\pi_i}^\dagger\overline{\pi_i}\rho^n) \end{cases}$$

$$U^m\rho^n \rightsquigarrow \rho'^n \qquad \text{with } \overline{U^m}\rho^n\overline{U^m}^\dagger = \rho'^n$$

$$\rho\otimes\rho' \rightsquigarrow \rho'' \qquad \text{with } \rho'' = \rho\otimes\rho'$$

$$\sum_i p_i\rho_i \rightsquigarrow \rho' \qquad \text{with } \rho' = \sum_i p_i\rho_i$$

$$\sum_i p_i t \rightsquigarrow t$$

$$(\sum_i p_i t_i)r \rightsquigarrow \sum_i p_i(t_i r)$$

$$\frac{t\rightsquigarrow r}{\lambda x.t \rightsquigarrow \lambda x.r} \qquad \frac{t\rightsquigarrow r}{ts\rightsquigarrow rs} \qquad \frac{t\rightsquigarrow r}{st\rightsquigarrow sr} \qquad \frac{t\rightsquigarrow r}{U^n t\rightsquigarrow U^n r}$$

$$\frac{t\rightsquigarrow r}{\pi^n t\rightsquigarrow \pi^n r} \qquad \frac{t\rightsquigarrow r}{t\otimes s\rightsquigarrow r\otimes s} \qquad \frac{t\rightsquigarrow r}{s\otimes t\rightsquigarrow s\otimes r}$$

$$\frac{t_j \rightsquigarrow r_j}{\sum_{i=1}^n p_i t_i \rightsquigarrow \sum_{i=1}^n p_i r_i}\ {\scriptstyle (\forall i\neq j, t_i = r_j)}$$

$$\frac{t\rightsquigarrow r}{\mathsf{letcase}^\circ\ x = t\ \mathsf{in}\ \{s_0,\dots,s_{2^m-1}\} \rightsquigarrow \mathsf{letcase}^\circ\ x = r\ \mathsf{in}\ \{s_0,\dots,s_{2^m-1}\}}$$

(cf. Table 3), is that rule $\mathsf{ax_{am}}$ is no longer needed, since $(b^m, \rho^n)$ is not in the grammar of $\lambda_\rho^\circ$, and there is a new rule $(+)$ typing the generalised mixed states. We use the symbol $\Vdash$ for $\lambda_\rho^\circ$ to distinguish it from $\vdash$ used in $\lambda_\rho$.

*Example 3.1.* The teleportation algorithm expressed in $\lambda_\rho$ in Example 2.1, is analogous for $\lambda_\rho^\circ$, only changing the term $\mathsf{letcase}$ by $\mathsf{letcase}^\circ$. Also, the type derivation is analogous. The difference is in the reduction. Let $\rho$ be the density matrix of a given quantum state (mixed or pure). Let

$$\rho_0^3 = \rho\otimes\beta_{00}, \qquad \rho_1^3 = (\mathsf{Cnot}\otimes I)\rho_0^3, \qquad \text{and} \qquad \rho_2^3 = (\mathsf{H}\otimes I\otimes I)\rho_1^3$$

The trace of the teleportation of $\rho$ in $\lambda_\rho$ is the following:

$$\begin{aligned}
&(\lambda x.\mathsf{letcase}\ y = \pi^2(\mathsf{H}^1(\mathsf{Cnot}^2(x\otimes\beta_{00})))\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\})\rho \\
&\longrightarrow_1 \mathsf{letcase}\ y = \pi^2(\mathsf{H}^1(\mathsf{Cnot}^2(\rho\otimes\beta_{00})))\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\} \\
&\longrightarrow_1 \mathsf{letcase}\ y = \pi^2(\mathsf{H}^1(\mathsf{Cnot}^2\rho_0^3))\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\} \\
&\longrightarrow_1 \mathsf{letcase}\ y = \pi^2(\mathsf{H}^1\rho_1^3)\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\} \\
&\longrightarrow_1 \mathsf{letcase}\ y = \pi^2\rho_2^3\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\}
\end{aligned} \tag{1}$$

**Table 8.** Type system for $\lambda_\rho^\circ$.

$$A := n \mid (m,n) \mid A \multimap A$$

where $m \leq n \in \mathbb{N}$.

$$\frac{}{\Gamma, x:A \Vdash x:A}\ \mathsf{ax} \qquad \frac{\Gamma, x:A \Vdash t:B}{\Gamma \Vdash \lambda x.t : A \multimap B}\ \multimap_i \qquad \frac{\Gamma \Vdash t : A \multimap B \quad \Delta \Vdash r : A}{\Gamma, \Delta \Vdash tr : B}\ \multimap_e$$

$$\frac{}{\Gamma \Vdash \rho^n : n}\ \mathsf{ax}_\rho \qquad \frac{\Gamma \Vdash t : n}{\Gamma \Vdash U^m t : n}\ \mathsf{u} \qquad \frac{\Gamma \Vdash t : n}{\Gamma \Vdash \pi^m t : (m,n)}\ \mathsf{m} \qquad \frac{\Gamma \Vdash t : n \quad \Delta \Vdash r : m}{\Gamma, \Delta \Vdash t \otimes r : n+m}\ \otimes$$

$$\frac{x:n \Vdash t_0 : A \quad \dots \quad x:n \Vdash t_{2^m-1} : A \quad \Gamma \Vdash r : (m,n)}{\Gamma \Vdash \mathsf{letcase}^\circ\ x = r\ \mathsf{in}\ \{t_0, \dots, t_{2^m-1}\} : A}\ \mathsf{lc}$$

$$\frac{\Gamma \Vdash t_1 : A \quad \dots \quad \Gamma \Vdash t_n : A \quad \sum_{i=1}^n p_i = 1}{\Gamma \Vdash \sum_{i=1}^n p_i t_i : A}\ +$$

From (1), there are four possible reductions. For $i = 0,1,2,3$, let $p_i = \mathsf{tr}(\overline{\pi_i}^\dagger \overline{\pi_i} \rho_2^3)$ and $\rho_{3i}^3 = \frac{\overline{\pi_i}\rho_2^3\overline{\pi_i}^\dagger}{p_i}$. Then,

- (1)$\longrightarrow_{p_0}$ $\mathsf{letcase}\ y = (0, \rho_{30}^3)\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\} \longrightarrow_1 \rho_{30}^3 = \rho$.
- (1)$\longrightarrow_{p_1}$ $\mathsf{letcase}\ y = (1, \rho_{31}^3)\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\} \longrightarrow_1 \mathsf{Z}_3\rho_{31}^3 \longrightarrow_1 \rho$.
- (1)$\longrightarrow_{p_2}$ $\mathsf{letcase}\ y = (2, \rho_{32}^3)\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\} \longrightarrow_1 \mathsf{X}_3\rho_{32}^3 \longrightarrow_1 \rho$.
- (1)$\longrightarrow_{p_3}$ $\mathsf{letcase}\ y = (3, \rho_{33}^3)\ \mathsf{in}\ \{y, \mathsf{Z}_3 y, \mathsf{X}_3 y, \mathsf{Z}_3\mathsf{X}_3 y\} \longrightarrow_1 \mathsf{Z}_3\mathsf{X}_3\rho_{33}^3 \longrightarrow_1 \rho$.

On the other hand, the trace of the same term, in $\lambda_\rho^\circ$, would be analogous until (1), just using $\rightsquigarrow$ instead of $\longrightarrow_1$. Then:

$$(1) \rightsquigarrow p_0\rho + p_1\mathsf{Z}_3\rho_{31}^3 + p_2\mathsf{X}_3\rho_{32}^3 + p_3\mathsf{Z}_3\mathsf{X}_3\rho_{33}^3 \rightsquigarrow^* \sum_{i=0}^3 p_i\rho_{30}^3 \rightsquigarrow (\sum_{i=0}^3 p_i)\rho \rightsquigarrow \rho$$

### 3.2 Interpretation

The interpretation of $\lambda_\rho$ given in Sect. 2.2 considers already all the traces. Hence, the interpretation of $\lambda_\rho^\circ$ can be obtained from a small modification of it. We only need to drop the interpretation of the term that no longer exists, $(b^m, \rho^n)$, and add an interpretation for the new term $\sum_i p_i t_i$ as follows:

$$[\![\sum_i p_i t_i]\!]_\theta = \{(p_i q_{ij}, b_{ij}, e_{ij}) \mid [\![t_i]\!]_\theta = \{(q_{ij}, b_{ij}, e_{ij})\}_j\}$$

The interpretation of $\mathsf{letcase}^\circ$ is the same as the interpretation of $\mathsf{letcase}$.

Then, we can prove a theorem (Theorem 3.4) for $\lambda_\rho^\circ$ analogous to Theorem 2.7.

We need the following auxiliary Lemmas.

**Lemma 3.2.** *If $\Gamma \Vdash t : A$ and $\theta \vDash \Gamma$, then* $(\!|\sum_i p_i t_i|\!)_\theta = \sum_i p_i (\!|t_i|\!)_\theta$

*Proof.* Let $[\![t_i]\!]_\theta = \{(q_{ij}, b_{ij}, e_{ij})\}_j$. Then, we have $(\!|\sum_i p_i t_i|\!)_\theta = \sum_{ij} p_i q_{ij} e_{ij} = \sum_i p_i \sum_j q_{ij} e_{ij} = \sum_i p_i (\!|t_i|\!)_\theta$. □

**Lemma 3.3.** *Let* $[\![r]\!]_\theta = \{(p_i, b_i, e_i)\}_i$, *then* $(\!|t[r/x]|\!)_\theta = \sum_i p_i (\!|t|\!)_{\theta, x=(b_i,e_i)}$.

*Proof.* The proof of the analogous Lemma 2.6 in $\lambda_\rho$ follows by induction on $t$. Since the definition of $[\![\cdot]\!]$ is the same for $\lambda_\rho$ than for $\lambda_\rho^\circ$, we only need to check the only term of $\lambda_\rho^\circ$ which is not a term of $\lambda_\rho$: $\sum_j q_j t_j$. Using Lemma 3.2, and the induction hypothesis, we have $(\!|(\sum_j q_j t_j)[r/x]|\!)_\theta = (\!|\sum_j q_j (t_j[r/x])|\!)_\theta = \sum_j q_j (\!|t_j[r/x]|\!)_\theta = \sum_j q_j \sum_i p_i (\!|t_j|\!)_{\theta, x=(b_i,e_i)} = \sum_i p_i (\!|\sum_j q_j t_j|\!)_{\theta, x=(b_i,e_i)}$. □

**Theorem 3.4.** *If $\Gamma \Vdash t : A$, $\theta \vDash \Gamma$ and $t \rightsquigarrow r$, then* $(\!|t|\!)_\theta = (\!|r|\!)_\theta$.

*Proof.* By induction on the relation $\rightsquigarrow$. Rules $(\lambda x.t)r \rightsquigarrow t[r/x]$, $U^m \rho^n \rightsquigarrow \rho'$ and $\rho \otimes \rho' \rightsquigarrow \rho''$ are also valid rules for relation $\longrightarrow_1$, and hence the proof of these cases are the same than in Theorem 2.7. □

## 4 Subject Reduction and Progress

In this section we state and prove the subject reduction and progress properties on both, $\lambda_\rho$ and $\lambda_\rho^\circ$ (Theorems 4.4 and 4.7 respectively).

**Lemma 4.1 (Weakening)**

- *If $\Gamma \vdash t : A$ and $x \notin FV(t)$, then $\Gamma, x : B \vdash t : A$.*
- *If $\Gamma \Vdash t : A$ and $x \notin FV(t)$, then $\Gamma, x : B \Vdash t : A$.*

*Proof.* By a straightforward induction on the derivation of $\Gamma \vdash t : A$ and on $\Gamma \Vdash t : A$. □

**Lemma 4.2 (Strengthening)**

- *If $\Gamma, x : A \vdash t : B$ and $x \notin FV(t)$, then $\Gamma \vdash t : B$.*
- *If $\Gamma, x : A \Vdash t : B$ and $x \notin FV(t)$, then $\Gamma \Vdash t : B$.*

*Proof.* By a straightforward induction on the derivation of $\Gamma, x : A \vdash t : B$ and $\Gamma, x : A \Vdash t : B$. □

**Lemma 4.3 (Substitution)**

- *If $\Gamma, x : A \vdash t : B$ and $\Delta \vdash r : A$ then $\Gamma, \Delta \vdash t[r/x] : B$.*
- *If $\Gamma, x : A \Vdash t : B$ and $\Delta \Vdash r : A$ then $\Gamma, \Delta \Vdash t[r/x] : B$.*

*Proof.* By induction on $t$. □

**Theorem 4.4 (Subject reduction)**

- *If* $\Gamma \vdash t : A$*, and* $t \longrightarrow_p r$*, then* $\Gamma \vdash r : A$*.*
- *If* $\Gamma \Vdash t : A$*, and* $t \rightsquigarrow r$*, then* $\Gamma \Vdash r : A$*.*

*Proof.* By induction on the relations $\longrightarrow_p$ and $\rightsquigarrow$. ☐

**Definition 4.5 (Values)**

- *A value in* $\lambda_\rho$ *is a term* $v$ *defined by the following grammar:*

$$w := x \mid \lambda x.v \mid w \otimes w$$
$$v := w \mid \rho^n \mid (b^m, \rho^n).$$

- *A value in* $\lambda_\rho^\circ$ *(or value*$^\circ$*) is a term* $v$ *defined by the following grammar:*

$$w := x \mid \lambda x.v \mid w \otimes w \mid \sum_i p_i w_i \textit{ with } w_i \neq w_j \textit{ if } i \neq j$$
$$v := w \mid \rho^n$$

**Lemma 4.6**

1. *If* $v$ *is a value, then there is no* $t$ *such that* $v \longrightarrow_p t$ *for any* $p$*.*
2. *If* $v$ *is a value*$^\circ$*, then there is no* $t$ *such that* $v \rightsquigarrow t$*.*

*Proof.* By induction on $v$ in both cases. ☐

**Theorem 4.7 (Progress)**

1. *If* $\vdash t : A$*, then either* $t$ *is a value or there exist* $n$*,* $p_1, \ldots, p_n$*, and* $r_1, \ldots, r_n$ *such that* $t \longrightarrow_{p_i} r_i$*.*
2. *If* $\Vdash t : A$ *and* $A \neq (m, n)$*, then either* $t$ *is a value*$^\circ$ *or there exists* $r$ *such that* $t \rightsquigarrow r$*.*

*Proof.* We relax the hypotheses and prove the theorem for open terms as well. That is:

1. If $\Gamma \vdash t : A$, then either $t$ is a value, there exist $n$, $p_1, \ldots, p_n$, and $r_1, \ldots, r_n$ such that $t \longrightarrow_{p_i} r_i$, or $t$ contains a free variable, and $t$ does not rewrite.
2. If $\Gamma \Vdash t : A$, then either $t$ is a value$^\circ$, there exists $r$ such that $t \rightsquigarrow r$, or $t$ contains a free variable, and $t$ does not rewrite.

In both cases, we proceed by induction on the type derivation. ☐

## 5 Examples

*Example 5.1.* Consider the following experiment: Measure some $\rho$ and then toss a coin to decide whether to return the result of the measurement, or to give the result of tossing a new coin.

**The experiment in $\lambda_\rho$.** This experiment can be implemented in $\lambda_\rho$ as follows:

$$(\mathsf{letcase}\ y = \pi^1|+\rangle\langle+|\ \mathsf{in}\ \{\lambda x.x, \lambda x.\mathsf{letcase}\ w = \pi^1|+\rangle\langle+|\ \mathsf{in}\ \{w,w\}\}) \\ (\mathsf{letcase}\ z = \pi^1\rho\ \mathsf{in}\ \{z,z\})$$

*Trace:* We give one possible probabilistic trace. Notice that, by using different strategies, we would get different derivation trees. We will not prove confluence in this setting (cf. [12] for a full discussion on the notion of confluence of probabilistic rewrite systems), but we conjecture that such a property is meet.

We use the following notations:

$$\begin{aligned}
s &= \pi^1|+\rangle\langle+| \\
t_0 &= \lambda x.x \\
t_1 &= \lambda x.\mathsf{letcase}\ w = s\ \mathsf{in}\ \{w,w\} \\
\rho &= \frac{3}{4}|0\rangle\langle 0| + \frac{\sqrt{3}}{4}|0\rangle\langle 1| + \frac{\sqrt{3}}{4}|1\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1| \\
r_1 &= \mathsf{letcase}\ y = s\ \mathsf{in}\ \{t_0, t_1\} \\
r_2 &= \mathsf{letcase}\ z = \pi^1\rho\ \mathsf{in}\ \{z,z\} \\
\mathsf{l}_x &= \mathsf{letcase}\ y = (x, |x\rangle\langle x|)\ \mathsf{in}\ \{y,y\}\ \text{with } x = 0,1 \\
r_1^x &= \mathsf{letcase}\ y = (x, |x\rangle\langle x|)\ \mathsf{in}\ \{t_0,t_1\}\ \text{with } x = 0,1
\end{aligned}$$

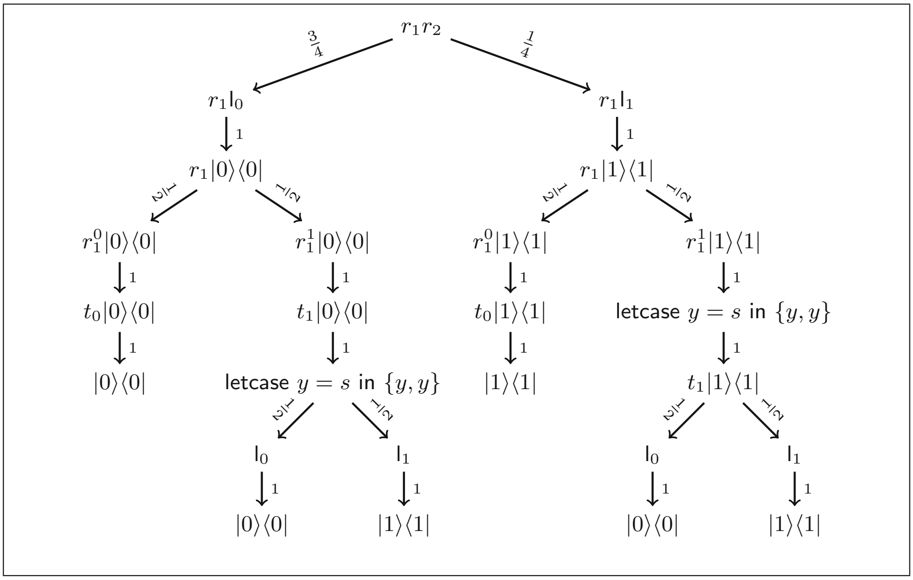
Using this notation, the probabilistic trace is given by the tree in Table 9. Therefore, with probability $\frac{5}{8}$ we get $|0\rangle\langle 0|$, and with probability $\frac{3}{8}$ we get $|1\rangle\langle 1|$. Thus, the density matrix of this mixed state is $\frac{5}{8}|0\rangle\langle 0| + \frac{3}{8}|1\rangle\langle 1|$.

*Typing:*

$$\dfrac{\dfrac{\dfrac{}{y:1,x:1,w:1\vdash w:1}\,\mathsf{ax} \quad \dfrac{}{y:1,x:1,w:1\vdash w:1}\,\mathsf{ax} \quad \dfrac{\dfrac{}{\vdash |+\rangle\langle+|:1}\,\mathsf{ax}_\rho}{\vdash \pi^1|+\rangle\langle+|:(1,1)}\,\mathsf{m}}{y:1,x:1\vdash \mathsf{letcase}\ w=\pi^1|+\rangle\langle+|\ \mathsf{in}\ \{w,w\}:1}\,\mathsf{lc}}{y:1\vdash \lambda x.\mathsf{letcase}\ w=\pi^1|+\rangle\langle+|\ \mathsf{in}\ \{w,w\}:1\multimap 1}\,\multimap_i \qquad (2)$$

$$\dfrac{\dfrac{\dfrac{}{y:1,x:1\vdash x:1}\,\mathsf{ax}}{y:1\vdash \lambda x.x:1\multimap 1}\,\multimap_i \quad \dfrac{\vdots}{y:1\vdash t_1:1\multimap 1}\,(2) \quad \dfrac{\dfrac{}{\vdash |+\rangle\langle+|:1}\,\mathsf{ax}_\rho}{\vdash \pi^1|+\rangle\langle+|:(1,1)}\,\mathsf{m}}{\vdash \mathsf{letcase}\ y=\pi^1|+\rangle\langle+|\ \mathsf{in}\ \{t_0,t_1\}:1\multimap 1}\,\mathsf{lc} \qquad (3)$$

**Table 9.** Trace of the $\lambda_\rho$ term implementing the experiment of Example 5.1.




$$
\dfrac{\dfrac{\vdots}{\vdash \mathsf{letcase}\ y = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{t_0, t_1\} : 1 \multimap 1}\ (3) \qquad \dfrac{\dfrac{}{z : 1 \vdash z : 1}\,\mathsf{ax} \qquad \dfrac{\dfrac{}{\vdash \rho : 1}\,\mathsf{ax}_\rho}{\vdash \pi^1\rho : (1,1)}\,\mathsf{m}}{\vdash \mathsf{letcase}\ z = \pi^1\rho\ \mathsf{in}\ \{z, z\} : 1}\,\mathsf{lc}}{\vdash (\mathsf{letcase}\ y = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{t_0, t_1\})(\mathsf{letcase}\ z = \pi^1\rho\ \mathsf{in}\ \{z, z\}) : 1}\,\multimap_e
$$

*Interpretation:*

$$[\![s]\!]_\emptyset = \{(\frac{1}{2}, 0, |0\rangle\langle 0|), (\frac{1}{2}, 1, |1\rangle\langle 1|)\}$$

$$[\![t_0]\!]_{y=(\varepsilon,|0\rangle\langle 0|)} = \{(1, \varepsilon, (\mathsf{b}, \mathsf{e}) \mapsto \{(1, \mathsf{b}, \mathsf{e})\})\}$$

$$[\![t_1]\!]_{y=(\varepsilon,|1\rangle\langle 1|)} = \{(1, \varepsilon, (\mathsf{b}, \mathsf{e}) \mapsto \{(\frac{1}{2}, \varepsilon, |0\rangle\langle 0|), (\frac{1}{2}, \varepsilon, |1\rangle\langle 1|)\})\}$$

$$[\![r_1]\!]_\emptyset = \{(\frac{1}{2}, \varepsilon, (\mathsf{b}, \mathsf{e}) \mapsto \{(\frac{1}{2}, \varepsilon, |0\rangle\langle 0|), (\frac{1}{2}, \varepsilon, |1\rangle\langle 1|)\}), (\frac{1}{2}, \varepsilon, (\mathsf{b}, \mathsf{e}) \mapsto \{(1, \mathsf{b}, \mathsf{e})\})\}$$

$$[\![\pi^1\rho]\!]_\emptyset = \{(\frac{3}{4}, 0, |0\rangle\langle 0|), (\frac{1}{4}, 1, |1\rangle\langle 1|)\}$$

$$[\![r_2]\!]_\emptyset = \{(\frac{3}{4}, \varepsilon, |0\rangle\langle 0|), (\frac{1}{4}, \varepsilon, |1\rangle\langle 1|)\}$$

Then,

$$[\![r_1 r_2]\!]_\emptyset = \{(\frac{3}{16}, \varepsilon, |0\rangle\langle 0|), (\frac{1}{16}, \varepsilon, |0\rangle\langle 0|), (\frac{3}{16}, \varepsilon, |1\rangle\langle 1|),$$
$$(\frac{1}{16}, \varepsilon, |1\rangle\langle 1|), (\frac{3}{8}, \varepsilon, |0\rangle\langle 0|), (\frac{1}{8}, \varepsilon, |1\rangle\langle 1|)\}$$

Hence,

$$\begin{aligned}(\!|r_1 r_2|\!)_\emptyset &= \frac{3}{16}|0\rangle\langle 0| + \frac{1}{16}|0\rangle\langle 0| + \frac{3}{16}|1\rangle\langle 1| + \frac{1}{16}|1\rangle\langle 1| + \frac{3}{8}|0\rangle\langle 0| + \frac{1}{8}|1\rangle\langle 1| \\ &= \frac{5}{8}|0\rangle\langle 0| + \frac{3}{8}|1\rangle\langle 1|\end{aligned}$$

**The experiment in $\lambda_\rho^\circ$.** In $\lambda_\rho^\circ$, the example becomes:

$$t := (\mathsf{letcase}^\circ\ y = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{\lambda x.x, \lambda x.\mathsf{letcase}^\circ\ w = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{w, w\}\})$$
$$(\mathsf{letcase}^\circ\ z = \pi^1\rho\ \mathsf{in}\ \{z, z\})$$

*Trace:* In this case the trace is not a tree, because the relation $\rightsquigarrow$ is not probabilistic. We use the same $\rho$ as before: $\frac{3}{4}|0\rangle\langle 0| + \frac{\sqrt{3}}{4}|1\rangle\langle 0| + \frac{\sqrt{3}}{4}|0\rangle\langle 1| + \frac{1}{4}|1\rangle\langle 1|$.

$$\begin{aligned}t &\rightsquigarrow (\mathsf{letcase}^\circ\ y = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{\lambda x.x, \lambda x.\mathsf{letcase}^\circ\ w = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{w, w\}\}) \\ &\qquad (\frac{3}{4}|0\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1|) \\ &\rightsquigarrow (\frac{1}{2}\lambda x.x + \frac{1}{2}\lambda x.\mathsf{letcase}^\circ\ w = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{w, w\})(\frac{3}{4}|0\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1|) \\ &\rightsquigarrow (\frac{1}{2}\lambda x.x + \frac{1}{2}(\lambda x.\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|))(\frac{3}{4}|0\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1|) \\ &\rightsquigarrow \frac{1}{2}((\lambda x.x)(\frac{3}{4}|0\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1|)) \\ &\qquad + \frac{1}{2}((\lambda x.\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|)(\frac{3}{4}|0\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1|)) \\ &\rightsquigarrow \frac{1}{2}((\lambda x.x)(\frac{3}{4}|0\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1|)) + \frac{1}{2}(\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|) \\ &\rightsquigarrow \frac{1}{2}(\frac{3}{4}|0\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1|) + \frac{1}{2}(\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|) \\ &\rightsquigarrow \frac{5}{8}|0\rangle\langle 0| + \frac{3}{8}|1\rangle\langle 1|\end{aligned}$$

*Typing and Interpretation:* Since $t$ does not contain sums, its typing is analogous to the term in $\lambda_\rho$, as well as the interpretation.

*Example 5.2.* In [18, p. 371] there is an example of the freedom in the operator-sum representation by showing two quantum operators, which are actually the same. One is the process of tossing a coin and, according to its results, applying $I$ or $Z$ to a given qubit The second is the process performing a projective measurement with unknown outcome to the same qubit. These operations can be encoded in $\lambda_\rho$ by:

$$O_1 = \lambda y.\mathsf{letcase}\ x = \pi^1|+\rangle\langle+|\ \mathsf{in}\ \{y, \mathsf{Z}y\}$$
$$O_2 = \lambda y.\mathsf{letcase}\ x = \pi^1 y\ \mathsf{in}\ \{x, x\}$$

with $\pi^1 = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$.

Let us apply those operators to the qubit $\rho = \frac{3}{4}|0\rangle\langle 0| + \frac{\sqrt{3}}{4}|0\rangle\langle 1| + \frac{\sqrt{3}}{4}|1\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1|$. We can check that the terms $O_1\rho$ and $O_2\rho$ have different interpretations $[\![\cdot]\!]$. Let $\rho^- = \mathsf{Z}\rho\mathsf{Z}^\dagger$, then

$$[\![(\lambda y.\mathsf{letcase}\ x = \pi^1|+\rangle\langle+|\ \mathsf{in}\ \{y, \mathsf{Z}y\})\rho]\!]_\emptyset = \{(\frac{1}{2}, \varepsilon, \rho), (\frac{1}{2}, \varepsilon, \rho^-)\}$$
$$[\![(\lambda y.\mathsf{letcase}\ x = \pi^1 y\ \mathsf{in}\ \{x, x\})\rho]\!]_\emptyset = \{(\frac{3}{4}, \varepsilon, |0\rangle\langle 0|), (\frac{1}{4}, \varepsilon, |1\rangle\langle 1|)\}$$

However, they have the same interpretation $(\!|\cdot|\!)$.

$$\begin{aligned}
&(\!|(\lambda y.\mathsf{letcase}\ x = \pi^1|+\rangle\langle+|\ \mathsf{in}\ \{y, \mathsf{Z}y\})\rho|\!)_\emptyset \\
&= \frac{1}{2}\rho + \frac{1}{2}\rho^- \\
&= \frac{3}{4}|0\rangle\langle 0| + \frac{1}{4}|1\rangle\langle 1| \\
&= (\!|(\lambda y.\mathsf{letcase}\ x = \pi^1 y\ \mathsf{in}\ \{x, x\})\rho|\!)_\emptyset
\end{aligned}$$

**Table 10.** Trace of the terms $O_1\rho$ from Example 5.2 in $\lambda_\rho$.

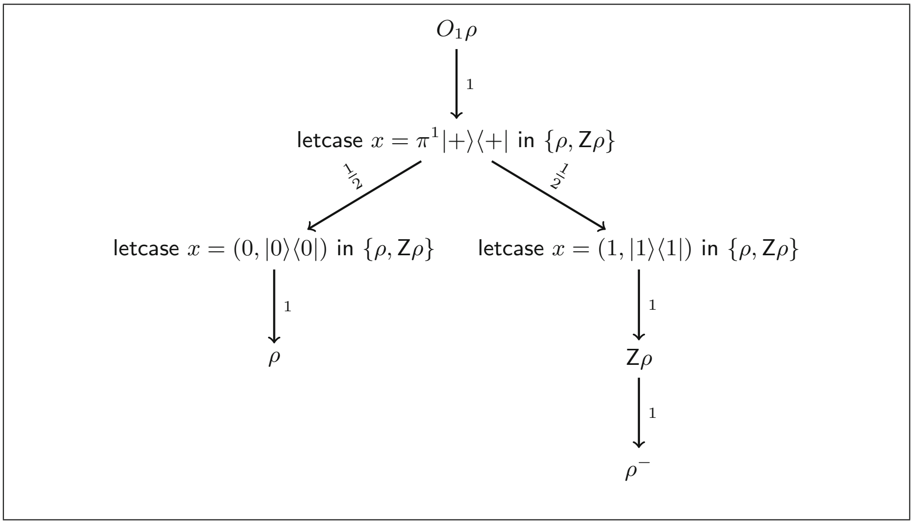
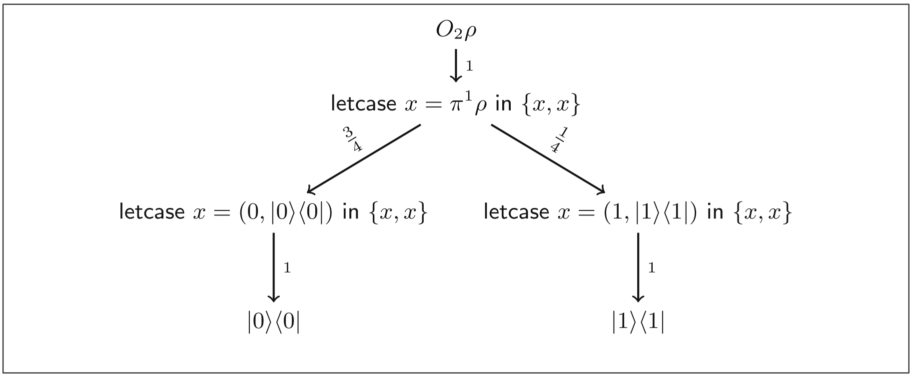


**Table 11.** Trace of the term $O_2\rho$ from Example 5.2 in $\lambda_\rho$.




The trace of $O_1\rho$ is given in Table 10, and the trace of $O_2\rho$ in Table 11. The first term produces $\rho$, with probability $\frac{1}{2}$, and $\rho^-$, with probability $\frac{1}{2}$, while the second term produces either $|0\rangle\langle 0|$ with probability $\frac{3}{4}$ or $|1\rangle\langle 1|$, with probability $\frac{1}{4}$.

However, if we encode the same terms in $\lambda_\rho^\circ$, we can get both programs to produce the same density matrix:

$$O_1^\circ = \lambda y.\mathsf{letcase}^\circ\ x = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{y, \mathsf{Z}y\}$$
$$O_2^\circ = \lambda y.\mathsf{letcase}^\circ\ x = \pi^1 y\ \mathsf{in}\ \{x, x\}$$

The traces of $O_1^\circ\rho$ and $O_2^\circ\rho$ are as follow:

$$\begin{aligned}
&O_1^\circ\rho\\
&= (\lambda y.\mathsf{letcase}^\circ\ x = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{y, \mathsf{Z}y\})\rho\\
&\rightsquigarrow \mathsf{letcase}^\circ\ x = \pi^1|+\rangle\langle +|\ \mathsf{in}\ \{\rho, \mathsf{Z}\rho\}\\
&\rightsquigarrow (\tfrac{1}{2}\rho) + (\tfrac{1}{2}\mathsf{Z}\rho)\\
&\rightsquigarrow (\tfrac{1}{2}\rho) + (\tfrac{1}{2}\rho^-)\\
&\rightsquigarrow \tfrac{3}{4}|0\rangle\langle 0| + \tfrac{1}{4}|1\rangle\langle 1|
\end{aligned}$$

$$\begin{aligned}
&O_2^\circ\rho\\
&= (\lambda y.\mathsf{letcase}^\circ\ x = \pi^1 y\ \mathsf{in}\ \{x, x\})\rho\\
&\rightsquigarrow \mathsf{letcase}^\circ\ x = \pi^1\rho\ \mathsf{in}\ \{x, x\})\\
&\rightsquigarrow (\tfrac{3}{4}|0\rangle\langle 0|) + (\tfrac{1}{4}|1\rangle\langle 1|)\\
&\rightsquigarrow \tfrac{3}{4}|0\rangle\langle 0| + \tfrac{1}{4}|1\rangle\langle 1|
\end{aligned}$$

## 6 Conclusions

In this paper we have presented the calculus $\lambda_\rho$, which is a quantum data/classical control extension to the lambda calculus where the data is manipulated by density matrices. The main importance of this calculus is its interpretation into density matrices, which can equate programs producing the same density matrices. Then, we have given a second calculus, $\lambda_\rho^\circ$, where the density matrices are generalised to accommodate arbitrary terms, and so, programs producing the same density matrices, rewrite to such a matrix, thus, coming closer to its interpretation. The control of $\lambda_\rho^\circ$ is not classical nor quantum, however it can be seen as a weaker version of the quantum control approach. It is indeed not classical control because a generalised density matrix of terms is allowed ($\sum_i p_i t_i$). It is not quantum control because superposition of programs are not allowed (indeed, the previous sum is not a quantum superposition since all the $p_i$ are positive and so no interference can occur). However, it is quantum in the sense that programs in a kind of generalised mixed-states are considered. We preferred to call it *probabilistic control*.

As depicted in Example 5.2, the calculus $\lambda_\rho^\circ$ allows to represent the same operator in different ways. Understanding when two operators are equivalent is important from a physical point of view: it gives insights on when two different physical processes produce the same dynamics. To the best of our knowledge, it is the first lambda calculus for density matrices.

**Future work and open questions.** As pointed out by Bădescu and Panangaden [8], one of the biggest issues with quantum control is that it does not accommodate well with traditional features from functional programming languages like recursion. Ying [26] went around this problem by introducing a recursion based on second quantisation. Density matrices are DCPOs with respect to the Löwner order. Is the form of weakened quantum control suggested in this paper monotone? Can it be extended with recursion? Could this lead to a concrete quantum programming language, like Quipper [16]?

All these open questions are promising new lines of research that we are willing to follow. In particular, we have four ongoing works trying to answer some of these questions:

– The most well studied quantum lambda calculus is, without doubt, Selinger-Valiron's $\lambda_q$ [22]. Hence, we are working on the mutual simulations between $\lambda_\rho$ and $\lambda_q$, and between $\lambda_\rho^\circ$ and a generalisation of $\lambda_q$ into mixed states.
– We are also working on a first prototype of an implementation of $\lambda_\rho^\circ$.
– We are studying extensions to both $\lambda_\rho$ and $\lambda_\rho^\circ$ with recursion and with polymorphism.
– Finally, we are studying a more sophisticated denotational semantics for both calculi than the one given in this paper. We hope such a semantics to be adequate and fully abstract.

**Acknowledgements.** We want to thank the anonymous reviewer for some important references and suggestions on future lines of work.

## References

1. Aharonov, D., Ambainis, A., Kempe, J., Vazirani, U.: Quantum walks on graphs. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC 2001, pp. 50–59. ACM (2001)
2. Altenkirch, T., Grattage, J.J.: A functional quantum programming language. In: Proceedings of LICS-2005, pp. 249–258. IEEE Computer Society (2005)
3. Ambainis, A., Bach, E., Nayak, A., Vishwanath, A., Watrous, J.: One-dimensional quantum walks. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC 2001, pp. 37–49. ACM (2001)

4. Arrighi, P., Díaz-Caro, A.: A System F accounting for scalars. Logical Methods Comput. Sci. **8**(1:11) (2012)
5. Arrighi, P., Díaz-Caro, A., Valiron, B.: The vectorial λ-calculus. Inf. Comput. **254**(1), 105–139 (2017)
6. Arrighi, P., Dowek, G.: Lineal: A linear-algebraic lambda-calculus. Logical Methods Comput. Sci. **13**(1:8) (2017)
7. Assaf, A., Díaz-Caro, A., Perdrix, S., Tasson, C., Valiron, B.: Call-by-value, call-by-name and the vectorial behaviour of the algebraic λ-calculus. Logical Methods Comput. Sci. **10**(4:8) (2014)
8. Bădescu, C., Panangaden, P.: Quantum alternation: prospects and problems. In: Heunen, C., Selinger, P., Vicary, J. (eds.) Proceedings of QPL-2015. Electronic Proceedings in Theoretical Computer Science, vol. 195, pp. 33–42 (2015)
9. D'Hondt, E., Panangaden, P.: Quantum weakest preconditions. Mathe. Struct. Comput. Sci. **16**, 429–451 (2006)
10. Díaz-Caro, A.: A lambda calculus for density matrices with classical and probabilistic controls. arXiv:1705.00097 (extended version with a 10-pages appendix with proofs) (2017)
11. Díaz-Caro, A., Dowek, G.: Typing quantum superpositions and measurements. To appear in LNCS 10687 (TPNC 2017)
12. Díaz-Caro, A., Martínez, G.: Confluence in probabilistic rewriting. LSFA, to appear in ENTCS (2017)
13. Díaz-Caro, A., Petit, B.: Linearity in the non-deterministic call-by-value setting. In: Ong, L., de Queiroz, R. (eds.) WoLLIC 2012. LNCS, vol. 7456, pp. 216–231. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32621-9_16
14. Feng, Y., Duan, R., Ying, M.: Bisimulation for quantum processes. ACM SIGPLAN Notices (POPL 2011) **46**(1), 523–534 (2011)
15. Feng, Y., Yu, N., Ying, M.: Model checking quantum markov chains. J. Comput. Syst. Sci. **79**(7), 1181–1198 (2013)
16. Green, A.S., Lumsdaine, P.L., Ross, N.J., Selinger, P., Valiron, B.: Quipper: a scalable quantum programming language. ACM SIGPLAN Notices (PLDI 2013) **48**(6), 333–342 (2013)
17. Dal Lago, U., Masini, A., Zorzi, M.: Confluence results for a quantum lambda calculus with measurements. Electron. Notes Theor. Comput. Sci. **270**(2), 251–261 (2011)
18. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
19. Pagani, M., Selinger, P., Valiron, B.: Applying quantitative semantics to higher-order quantum computing. ACM SIGPLAN Notices (POPL 2014) **49**(1), 647–658 (2014)
20. Selinger, P.: Towards a quantum programming language. Mathe. Struct. Comput. Sci. **14**(4), 527–586 (2004)
21. Selinger, P., Valiron, B.: Quantum lambda calculus. In: Gay, S., Mackie, I. (eds.) Semantic Techniques in Quantum Computation, Chap. 9, pp. 135–172. Cambridge University Press (2009)
22. Selinger, P., Valiron, B.: A lambda calculus for quantum computation with classical control. Mathe. Struct. Comput. Sci. **16**(3), 527–552 (2006)
23. van Tonder, A.: A lambda calculus for quantum computation. SIAM J. Comput. **33**, 1109–1135 (2004)
24. Vaux, L.: The algebraic lambda calculus. Mathe. Struct. Comput. Sci. **19**, 1029–1059 (2009)

25. Ying, M.: Floyd-hoare logic for quantum programs. ACM Trans. Program. Lang. Syst. **33**(6), 19:1–19:49 (2011)
26. Ying, M.: Quantum recursion and second quantisation. arXiv:1405.4443 (2014)
27. Ying, M.: Foundations of Quantum Programming. Elsevier (2016)
28. Ying, M., Ying, S., Wu, X.: Invariants of quantum programs: characterisations and generation. ACM SIGPLAN Notices (POPL 2017) **52**(1), 818–832 (2017)
29. Ying, M., Yu, N., Feng, Y.: Defining quantum control flow. arXiv:1209.4379 (2012)
30. Ying, M., Yu, N., Feng, Y.: Alternation in quantum programming: from superposition of data to superposition of programs. arXiv:1402.5172 (2014)
31. Zorzi, M.: On quantum lambda calculi: a foundational perspective. Mathe. Struct. Comput. Sci. **26**(7), 1107–1195 (2016)