Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Electronic Notes in Theoretical Computer Science 270 (2) (2011) 251–261

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Confluence Results for a Quantum Lambda Calculus with Measurements[1]

Ugo Dal Lago[2]

Dipartimento di Scienze dell'Informazione
Università di Bologna

Andrea Masini[3] Margherita Zorzi[4]

Dipartimento di Informatica
Università di Verona

Abstract

A strong confluence result for Q^*, a quantum λ-calculus with measurements, is proved. More precisely, confluence is shown to hold both for finite and infinite computations. The technique used in the confluence proof is syntactical but innovative. This makes Q^* different from similar quantum lambda calculi, which are either measurement-free or provided with a reduction strategy.

Keywords: Quantum computation, lambda calculus, confluence

1 Introduction

It is well known that the measurement-free evolution of a quantum system is deterministic. As a consequence it is to be expected that a good measurement-free quantum lambda calculus enjoys confluence. This is the case of Q, by the authors [4] and of the lambda calculus recently introduced by Arrighi and Dowek [1]. The situation becomes more complicated if we introduce a measurement operator. In fact measurements break the deterministic evolution of a quantum system.

[1] The authors are partially supported by PRIN project "CONCERTO" and FIRB grant RBIN04M8S8, "Intern. Inst. for Applicable Math."

[2] Email:dallago@cs.unibo.it

[3] Email:andrea.masini@univr.it

[4] Email:margherita.zorzi@univr.it

doi:10.1016/j.entcs.2011.01.035

An explicit measurement operator in the syntax allows an observation at an intermediate step of the computation: this feature is needed if we want, for example, to write algorithms such as Shor's factorization. In quantum calculi the intended meaning of a measurement is the observation of a (possibly superimposed) quantum bit, giving as output a classical bit; the two possible outcomes (i.e., the two possible values of the obtained classical bit) can be observed with two probabilities summing to 1. Since measurement forces a probabilistic evolution in the computation, it is not surprising that we need probabilistic instruments in order to investigate the main features of the language.

In this paper, we study an extension of Q obtained by endowing the language of terms with a suitable measurement operator and coherently extending the reduction relation, which becomes probabilistic for the reasons we have just explained. We investigate the resulting calculus, called Q^*, focusing, in particular, on confluence.

In Q^* and Q, states are formalized by *configurations*, i.e., triples in the form $[\mathcal{Q}, \mathcal{QV}, M]$, where M is a lambda term, $\mathcal{Q}$ is a quantum state, and $\mathcal{QV}$ is a set of names of quantum variables. So, control is classical (M is simply a term) while data is quantum ($\mathcal{Q}$ is an element of a finite-dimensional Hilbert space).

We are interested in the following question: what happens to properties such as confluence in presence of measurements? And moreover: is it possible to preserve confluence in the probabilistic setting induced by measurements? Apparently, the questions above cannot receive a positive answer: as we will see in section 3, it is possible to exhibit a configuration C such that there are two different reductions starting at C and ending in two essentially different configurations in normal form $[1, \emptyset, 0]$ and $[1, \emptyset, 1]$. In other words, confluence fails in its usual form. But the question now becomes: are the usual notions of computations and confluence adequate in this setting?

In Q^*, there are two distinct sources of divergence:

- On the one hand, a redex involving the measurement operator can be reduced in two different ways, i.e., divergence can come from *a single redex.*
- On the other hand, a term can contain more than one redex and Q^* is *not* endowed with a reduction strategy. As a consequence, some configurations can be reduced in different ways due to the presence of *distinct redexes* in a term.

We cannot hope to be confluent with respect to the first source of divergence, but we can anyway ask ourselves whether all reduction strategies are somehow equivalent. More precisely, we say that Q^* is *confluent* if for every configuration C and for every configuration in normal form D, there is a fixed real number p such that the probability of observing D when reducing C is always p, independently of the reduction strategy.

This notion of confluence can be easily captured by analyzing rewriting on *mixed states* rather than rewriting on configurations. A mixed state is a probabilistic distribution on configurations whose support is finite. Rewriting on configurations naturally extend to rewriting on mixed states. Rewriting on mixed states is *not* a probabilistic relation, and confluence is the usual confluence coming from rewriting theory [13].

In this paper, we prove that Q^* is indeed confluent in this sense. Technically, confluence is proved in an innovative way. The key point is that we need a new definition of computation. The usual notion of computation as a sequence of configurations is not adequate here. A notion of *probabilistic computation* replaces it, as something more general than a linear sequence of configurations but less general than the reduction tree: a probabilistic computation is a (possibly) infinite tree, in which binary choice (a node can have at most two children) corresponds to the two possible outcomes of a measurement. This new notion of computation is needed, because proving confluence *directly* on mixed states is non-trivial. As by-products, we prove other results in the style of confluence.

Another important property of any quantum lambda calculus with measurements is the importance of infinite computations. In the case of standard lambda calculus, the study of infinite computations is strongly related to the study of infinite lambda terms. This is not the case of Q^* (and in general of quantum calculi with measurements). This phenomenon forced us to extend the study of confluence to the case of infinite probabilistic computations. The proposed analysis is not standard and is based on new techniques.

Up to our knowledge, the only paper about confluence in a quantum setting is [6]. The authors claim to have studied confluence for an extension of Van Tonder's quantum lambda calculus λ_q [14] obtained by endowing λ_q with explicit qubits and a family of measurement operators. The main result consists in showing that confluence and the consistency of the operational semantics hold in the extended calculus, here called λ_M, *provided* the same holds in λ_q. This could be a promising result, weaker but similar to the one presented in this paper. In our opinion, however, [6] has some problems, which prevent us from properly evaluate it:

- The significance of the main result is not completely clear. The crucial point is that λ_M is *not* an extension of λ_q, the main difference being the absence of a strategy (see, for example, the proof of Theorem 3.13 in [6], where the authors assume that reduction can happen under the scope of a λ-abstraction or when the argument to a β-redex is not a value), whereas the reduction relation on λ_q is completely deterministic, since λ_q is a call-by-value-calculus [14]. Moreover, the syntax of λ_M seems to be more restrictive than the one of λ_q in some respects, e.g. in λ_M "since measurement is linear, promotion to non-linear expression is disallowed" ([6], page 12, line 16) but in λ_q every term can be promoted (see [14], Figure 10).
- Some crucial definitions about the syntax of λ_M are ambiguous in [6]. The rules in Figure 2 allow to prove the well-formedness of syntactical objects which are not terms. Take, for example, the superposition rule: its conclusion is a syntactic object which cannot be derived from the rules in Figure 1, since e.g. $!|0\rangle$ is not a qubit constant. Moreover, it seems that syntactic objects like $t_1 \otimes t_2$ (where t_1 and t_2 are not qubit constants) cannot be terms. The authors themselves observed that in Note 1. Now, look at the definition of alice in Algorithm 2: what is $((Hr) \otimes w)$? It cannot be a term. Actually, we believe that λ_M as described in [6] cannot express the teleportation scheme.

The rest of this paper is structured as follow:

- in Section 2 the quantum λ-calculus Q^* is introduced;
- in Section 3 we introduce the confluence problem in an informal way;
- in Section 4 we give the definition of a *probabilistic computation*;
- in Section 5 a strong confluence result on probabilistic computations is given;
- in Section 6 *mixed states* and *mixed computations* are introduced, and we give a confluence theorem for mixed computations.

An extended version with all proofs is available [3].

2 A Brief Introduction to Q^*

In [4] we have introduced a measurement–free, untyped quantum λ–calculus, called Q, based on the *quantum data and classical control* paradigm (see e.g. [10,11]). In this paper we generalize Q by extending the class of terms with a measurement operator, obtaining Q^*. Space limitations prevent us from being exhaustive, and when needed, we will make reference to our paper [4] and to the literature.

Q^* is based on the notion of a *configuration*, namely a triple $[\mathcal{Q}, \mathcal{QV}, M]$ where $\mathcal{Q}$ is a *quantum register* [5] , $\mathcal{QV}$ is a finite set of names, called *quantum variables*, and M is an untyped *term* based on the linear lambda-calculus defined by Wadler [15] and Simpson [12]. **Conf** denotes the set of all such configurations.

Quantum registers are systems of n qubits, that, mathematically speaking, are normalized vectors of finite dimensional Hilbert spaces. In particular, a quantum register $\mathcal{Q}$ of a configuration $[\mathcal{Q}, \mathcal{QV}, M]$, is a normalized vector of the Hilbert space $\ell^2(\{0,1\}^{\mathcal{QV}})$, denoted here with $\mathcal{H}(\mathcal{QV})$. [6] Roughly speaking, the reader not familiar with Hilbert spaces could think that quantum variables are pointers to qubits in the quantum register.

There are three kinds of operations on quantum registers: *(i)* the *new* operation, responsible of the creation of qubits; *(ii) unitary operators*: each unitary operator $\mathbf{U}_{\langle\langle q_1,\ldots,q_n\rangle\rangle}$ corresponds to a pure quantum operation acting on qubits with names $q_1, \ldots, q_n$ (mathematically, a unitary transform on the Hilbert space $\mathcal{H}(\{q_1, \ldots, q_n\})$, see [4]); *(iii) one qubit measurement* operations $\mathcal{M}_{r,0}, \mathcal{M}_{r,1}$ responsible of the probabilistic reduction of the quantum state plus the destruction of the qubit referenced by r: given a quantum register $\mathcal{Q} \in \mathcal{H}(\mathcal{QV})$, and a quantum variable name $r \in \mathcal{QV}$, we allow the (destructive) measurement of the qubit with name r [7] .

The other main component of a configuration is a *term*. The set of terms is built from *(i)* a denumerable set of *classical variables*, ranged over by $x, x_0, \ldots$;

[5] the "empty" quantum register will be denoted with the scalar number 1.

[6] see [4] for a full discussion of $\mathcal{H}(\mathcal{QV})$ and [9] for a general treatment of $\ell^2(S)$ spaces.

[7] More precisely, for every quantum variable r we assume the existence of two linear measurement operators, $\mathcal{M}_{r,0}, \mathcal{M}_{r,1} : \mathcal{H}(\mathcal{QV}) \to \mathcal{H}(\mathcal{QV} - \{r\})$ enjoying the completeness condition ${\mathcal{M}_{r,0}}^\dagger\mathcal{M}_{r,0} + {\mathcal{M}_{r,1}}^\dagger\mathcal{M}_{r,1} = id_{\mathcal{H}(\mathcal{QV})}$ and such that, given a quantum register $\mathcal{Q} \in \mathcal{H}(\mathcal{QV})$, the measurement of the qubit with name r in $\mathcal{Q}$ gives the outcome c (with $c \in \{0,1\}$) with probability $p_c = \langle\mathcal{Q}|{\mathcal{M}_{r,c}}^\dagger\mathcal{M}_{r,c}|\mathcal{Q}\rangle$ and produces the new quantum register $\frac{\mathcal{M}_{r,c}\mathcal{Q}}{\sqrt{p_c}}$; see [8,7] for a detailed discussion of general pure measurements and [3] for formal definitions and detailed results about $\mathcal{M}_{r,0}$ and $\mathcal{M}_{r,1}$.

(ii) a denumerable set of *quantum variables*, ranged over by $r, r_0, \ldots$; *(iii)* a finite or at most denumerable set of names corresponding to *unitary operators*; *(iv)* the *boolean constants* $0, 1$ and *(v)* the operators `new` and `meas`. An *environment* Γ is a (possibly empty) finite set in the form $\Lambda, !\Delta$, where Λ is a (possibly empty) set of classical and quantum variables, and $!\Delta$ denote a (possibly empty) set of patterns $!x_1, \ldots, !x_n$. We impose that in an environment, each classical variable x occurs at most once (either as $!x$ or as x). A *judgement* is an expression $\Gamma \vdash M$, where Γ is an environment and M is a term. We say that a judgement is *well-formed* if it is derivable by means of the *well-forming rules* in Figure 1.

$$\dfrac{}{!\Delta \vdash C}\,\mathsf{const} \qquad \dfrac{}{!\Delta, r \vdash r}\,\mathsf{qvar} \qquad \dfrac{}{!\Delta, x \vdash x}\,\mathsf{cvar} \qquad \dfrac{}{!\Delta, !x \vdash x}\,\mathsf{der}$$

$$\dfrac{!\Delta \vdash M}{!\Delta \vdash !M}\,\mathsf{prom} \qquad \dfrac{\Lambda_1, !\Delta \vdash M \quad \Lambda_2, !\Delta \vdash N}{\Lambda_1, \Lambda_2, !\Delta \vdash MN}\,\mathsf{app} \qquad \dfrac{\Lambda_1, !\Delta \vdash M_1 \cdots \Lambda_k, !\Delta \vdash M_k}{\Lambda_1, \ldots, \Lambda_k, !\Delta \vdash \langle M_1, \ldots, M_k \rangle}\,\mathsf{tens}$$

$$\dfrac{\Gamma \vdash M}{\Gamma \vdash \mathtt{new}(M)}\,\mathsf{new} \qquad \dfrac{\Gamma, x_1, \ldots, x_n \vdash M}{\Gamma \vdash \lambda\langle x_1, \ldots, x_n \rangle.M}\,\mathsf{lam}_1 \qquad \dfrac{\Gamma, x \vdash M}{\Gamma \vdash \lambda x.M}\,\mathsf{lam}_2 \qquad \dfrac{\Gamma, !x \vdash M}{\Gamma \vdash \lambda !x.M}\,\mathsf{lam}_3$$

$$\dfrac{\Gamma \vdash M}{\Gamma \vdash \mathtt{meas}(M)}\,\mathsf{meas} \qquad \dfrac{\Lambda \vdash N \quad !\Delta \vdash M_1 \quad !\Delta \vdash M_2}{\Lambda, !\Delta \vdash \ \mathtt{if}\ N\ \mathtt{then}\ M_1\ \mathtt{else}\ M_2}\,\mathsf{if}$$

Fig. 1. Well–Forming Rules

Let $\mathscr{L} = \{\mathsf{Uq}, \mathsf{new}, \mathsf{l}.\beta, \mathsf{q}.\beta, \mathsf{c}.\beta, \mathsf{l.cm}, \mathsf{r.cm}, \mathsf{if}_1, \mathsf{if}_0, \mathsf{meas}_r\}$. For every $\alpha \in \mathscr{L}$ and for every $p \in \mathbb{R}_{[0,1]}$, we define a relation $\to^p_\alpha \subseteq \mathbf{Conf} \times \mathbf{Conf}$ by the set of rewriting rules *contractions* in Figure 2, plus standard closure rules. The notation $C \to_\alpha D$ stands for $C \to^1_\alpha D$. In order to be consistent with the so-called non-cloning and non-erasing properties, we adopt surface reduction [12,4]: reduction is not allowed in the scope of any ! operator. Furthermore, as usual, we also forbid reduction in N and P in the term `if` M `then` N `else` P. Observe that contractions include two commutative rules $\mathsf{l.cm}$ and $\mathsf{r.cm}$ (see Figure 2): they come from Q, where they were essential to get *quantum standardization* [4]. We distinguish three particular subsets

$$[\mathcal{Q}, \mathcal{QV}, (\lambda x.M)N] \to^1_{\mathsf{l}.\beta} [\mathcal{Q}, \mathcal{QV}, M\{N/x\}] \qquad [\mathcal{Q}, \mathcal{QV}, (\lambda !x.M)!N] \to^1_{\mathsf{c}.\beta} [\mathcal{Q}, \mathcal{QV}, M\{N/x\}]$$

$$[\mathcal{Q}, \mathcal{QV}, (\lambda\langle x_1, \ldots, x_n \rangle.M)\langle r_1, \ldots, r_n \rangle] \to^1_{\mathsf{q}.\beta} [\mathcal{Q}, \mathcal{QV}, M\{r_1/x_1, \ldots, r_n/x_n\}]$$

$$[\mathcal{Q}, \mathcal{QV},\ \mathtt{if}\ 1\ \mathtt{then}\ M_1\ \mathtt{else}\ M_2] \to^1_{\mathsf{if}_1} [\mathcal{Q}, \mathcal{QV}, M_1]$$

$$[\mathcal{Q}, \mathcal{QV},\ \mathtt{if}\ 0\ \mathtt{then}\ M_1\ \mathtt{else}\ M_2] \to^1_{\mathsf{if}_1} [\mathcal{Q}, \mathcal{QV}, M_2]$$

$$[\mathcal{Q}, \mathcal{QV}, U\langle r_{i_1}, ..., r_{i_n} \rangle] \to^1_{\mathsf{Uq}} [\mathbf{U}_{\langle\langle r_{i_1}, ..., r_{i_n} \rangle\rangle}\mathcal{Q}, \mathcal{QV}, \langle r_{i_1}, ..., r_{i_n} \rangle]$$

$$[\mathcal{Q}, \mathcal{QV}, \mathtt{meas}(r)] \to^{\mathsf{meas}_r}_{p_c} [\mathcal{M}_{r,c}(\mathcal{Q}), \mathcal{QV} - \{r\}, !c] \qquad (c \in \{0, 1\} \text{ and } p_c \in \mathbb{R}_{[0,1]})$$

$$[\mathcal{Q}, \mathcal{QV}, \mathtt{new}(c)] \to^1_{\mathsf{new}} [\mathcal{Q} \otimes |r \mapsto c\rangle, \mathcal{QV} \cup \{r\}, r] \qquad (r \text{ is fresh})$$

$$[\mathcal{Q}, \mathcal{QV}, L((\lambda\pi.M)N)] \to^1_{\mathsf{l.cm}} [\mathcal{Q}, \mathcal{QV}, (\lambda\pi.LM)N]$$

$$[\mathcal{Q}, \mathcal{QV}, ((\lambda\pi.M)N)L] \to^1_{\mathsf{r.cm}} [\mathcal{Q}, \mathcal{QV}, (\lambda\pi.ML)N]$$

Fig. 2. Contractions.

of $\mathscr{L}$, namely $\mathscr{K} = \{\mathsf{l.cm}, \mathsf{r.cm}\}$, $\mathscr{N} = \mathscr{L} - (\mathscr{K} \cup \{\mathsf{meas}_r\})$ and $n\mathscr{M} = \mathscr{L} - \{\mathsf{meas}_r\}$. In the following, we write $M \to_\alpha N$ meaning that there are $\mathcal{Q}$, $\mathcal{QV}$, $\mathcal{R}$ and $\mathcal{RV}$ such that $[\mathcal{Q}, \mathcal{QV}, M] \to_\alpha [\mathcal{R}, \mathcal{RV}, N]$. Similarly for the notation $M \to_{\mathscr{S}} N$ where $\mathscr{S}$ is a subset of $\mathscr{L}$.

3 The Confluence Problem: an Informal Introduction

The confluence problem is central for any quantum λ-calculus with measurements, as stressed in the introduction.

Let us consider the following configuration:

$$C = [1, \emptyset, (\lambda!x.(\ \mathtt{if}\ x\ \mathtt{then}\ 0\ \mathtt{else}\ 1))(\mathsf{meas}(H(\mathsf{new}(0))))].$$

If we focus on reduction sequences, it is easy to check that there are two different reduction sequences starting with C, the first ending in the normal form $[1, \emptyset, 0]$ (with probability 1/2) and the second in the normal form $[1, \emptyset, 1]$ (with probability 1/2). But if we reason with mixed states, the situation changes: the mixed state $\{1 : C\}$ (i.e., the mixed state assigning probability 1 to C and 0 to any other configuration) rewrites *deterministically* to $\{1/2 : [1, \emptyset, 0], 1/2 : [1, \emptyset, 1]\}$ (where both $[1, \emptyset, 0]$ and $[1, \emptyset, 1]$ have probability 1/2). So, confluence seems to hold.

Confluence in Other Quantum Calculi.

Contrarily to the measurement-free case, the above notion of confluence is *not* an expected result for a quantum lambda calculus. Indeed, it does not hold in the quantum lambda calculus λ_{sv} proposed by Selinger and Valiron [11]. In λ_{sv}, it is possible to exhibit a configuration C that gives as outcome the distribution $\{1 : [1, \emptyset, 0]\}$ when reduced call-by-value and the distribution $\{1/2 : [1, \emptyset, 0], 1/2 : [1, \emptyset, 1]\}$ if reduced call-by-name. This is a *real* failure of confluence, which is there even if one uses probability distributions in place of configurations. The same phenomenon cannot happen in Q^* (as we will show in Section 5): this fundamental difference can be traced back to another one: the linear lambda calculus with surface reduction (on which Q^* is based) enjoys (a slight variation on) the so-called diamond property [12], while in usual, pure, lambda calculus (on which λ_{sv} is based) confluence only holds in a weaker sense.

Finite or infinite rewriting?

In Q^*, an infinite computation can tend to a configuration which is essentially different from the configurations in the computation itself. For example, a configuration $C = [1, \emptyset, M]$ can be built [8] such that:

- after a finite number of reduction steps C rewrites to a distribution in the form $\{\sum_{1<i\leq n} \frac{1}{2^i} : [1, \emptyset, 0], 1 - \sum_{1<i\leq n} \frac{1}{2^i} : D\}$
- only after infinitely many reduction steps the distribution $\{1 : [1, \emptyset, 0]\}$ is reached.

[8] $M \equiv (\mathsf{Y}!(\lambda!f.\lambda!x\ \mathtt{if}\ x\ \mathtt{then}\ 0\ \mathtt{else}\ f(\mathtt{meas}(H(\mathtt{new}(0))))))(\mathtt{meas}(H(\mathtt{new}(0))))$, where Y is a fix point operator.

Therefore finite probability distributions of finite configurations could be obtained by means of infinite rewriting. We believe that the study of confluence for infinite computations is important.

Related Work.
In the literature, probabilistic rewriting systems have been already analyzed. For example, Bournez and Kirchner [2] have introduced the notion of a probabilistic abstract rewriting system as a structure $A = (|A|, [\cdot \rightsquigarrow \cdot])$ where $|A|$ is a set and $[\cdot \rightsquigarrow \cdot]$ is a function from $|A|$ to $\mathbb{R}$ such that for every $a \in |A|$, $\sum_{b\in|A|}[a \rightsquigarrow b]$ is either 0 or 1. Then, they define a notion of *probabilistic confluence* for a PARS: such a structure is probabilistically locally confluent iff the probability to be locally confluent, in a classical sense, is different from 0. Unfortunately, Bournez and Kirchner's analysis does not apply to Q^*, since Q^* is *not* a PARS. Indeed, the quantity $\sum_{b\in|A|}[a \rightsquigarrow b]$ can in general be any natural number. Similar considerations hold for the probabilistic lambda calculus introduced by Di Pierro, Hankin and Wiklicky in [5].

4 A Probabilistic Notion of Computation

We represent computations as (possibly) infinite trees. In the following, a (possibly) infinite tree T will be an $(n+1)$-tuple $[R, T_1, \ldots, T_n]$, where $n \geq 0$, R is the *root* of T and $T_1, \ldots, T_n$ are its *immediate subtrees.*

Definition 4.1 A set of (possibly) infinite trees $\mathscr{S}$ is said to be a *set of probabilistic computations* if $P \in \mathscr{S}$ iff (exactly) one of the following three conditions holds:

1. $P = [C]$ and $C \in \mathbf{Conf}$.
2. $P = [C, R]$, where $C \in \mathbf{Conf}$, $R \in \mathscr{S}$ has root D and $C \rightarrow_{n\mathscr{M}} D$
3. $P = [(p, q, C), R, Q]$, where $C \in \mathbf{Conf}$, $R, Q \in \mathscr{S}$ have roots D and E, $C \rightarrow^{p}_{meas_r} D$, $C \rightarrow^{q}_{meas_r} E$ and $p, q \in \mathbb{R}_{[0,1]}$;

The set of all (respectively, the set of finite) probabilistic computations is the largest set $\mathscr{P}$ (respectively, the smallest set $\mathscr{F}$) of probabilistic computations with respect to set inclusion. $\mathscr{P}$ and $\mathscr{F}$ exist because of the Knapster-Tarski Theorem.

We will often say that the root of $P = [(p, q, C), R, Q]$ is simply C, slightly diverging from the above definition without any danger of ambiguity.

Definition 4.2 A probabilistic computation P is *maximal* if for every leaf C in P, $C \in \mathsf{NF}$. More formally, (sets of) maximal probabilistic computations can be defined as in Definition 4.1, where clause 1 must be restricted to $C \in \mathsf{NF}$.

We can give definitions and proofs over *finite* probabilistic computations (i.e., over $\mathscr{F}$) by ordinary induction. An example is the following definition. Notice that the same is not true for arbitrary probabilistic computations, since $\mathscr{P}$ is not a well-founded set.

Definition 4.3 Let $P \in \mathscr{P}$ be a probabilistic computation. A finite probabilistic computation $R \in \mathscr{F}$ is a *sub-computation* of P, written $R \sqsubseteq P$ iff one of the following conditions is satisfied:

- $R = [C]$ and the root of P is C.
- $R = [C, Q]$, $P = [C, S]$, and $Q \sqsubseteq S$.
- $R = [(p, q, C), Q, S]$, $P = [(p, q, C), U, V]$, $Q \sqsubseteq U$ and $S \sqsubseteq V$.

Let $\delta : \mathbf{Conf} \to \{0, 1\}$ be a function defined as follows: $\delta(C) = 0$ if the quantum register of C is 0, otherwise, $\delta(C) = 1$.

Quantitative Properties of Computations.

The outcomes of a probabilistic computation P are given by the configurations which appear as leaves of P. Starting from this observation, the following definitions formalize some quantitative properties of probabilistic computations.

For every *finite* probabilistic computation P and every $C \in \mathsf{NF}$ we define $\mathcal{P}(P, C) \in \mathbb{R}_{[0,1]}$ and $\mathcal{N}(P, C) \leq \aleph_0$ by induction on the structure of P:

- $\mathcal{P}([C], C) = \mathcal{N}([C], C) = 1$ and $\mathcal{P}([C], D) = \mathcal{N}([C], D) = 0$ whenever $C \neq D$.
- $\mathcal{P}([C, P], D) = \mathcal{P}(P, D)$ and $\mathcal{N}([C, P], D) = \mathcal{N}(P, D)$.
- $\mathcal{P}([(p, q, C), P, R], D) = p\mathcal{P}(P, D) + q\mathcal{P}(R, D)$ and $\mathcal{N}([(p, q, C), P, R], D) = \mathcal{N}(P, D) + \mathcal{N}(R, D)$.

Informally, $\mathcal{P}(P, C)$ is the probability of observing C as a leaf in P, and $\mathcal{N}(P, C)$ is the number of times C appears as a leaf in P.

The definitions above can be easily modified to get the probability of observing *any* configuration (in normal form) as a leaf in P, $\mathcal{P}(P)$, or the number of times *any* configuration appears as a leaf in P, $\mathcal{N}(P)$. Since $\mathbb{R}_{[0,1]}$ and $\mathbb{N} \cup \{\aleph_0\}$ are complete lattices (with respect to standard orderings), we extend the above notions to the case of *arbitrary* probabilistic computations, by taking the least upper bound over all finite sub-computations. If $P \in \mathscr{P}$ and $C \in \mathsf{NF}$, then $\mathcal{P}(P, C) = \sup_{R \sqsubseteq P} \mathcal{P}(R, C)$, $\mathcal{N}(P, C) = \sup_{R \sqsubseteq P} \mathcal{N}(R, C)$, $\mathcal{P}(P) = \sup_{R \sqsubseteq P} \mathcal{P}(R)$, $\mathcal{N}(P) = \sup_{R \sqsubseteq P} \mathcal{N}(R)$. The quantities above exists because $\mathbb{R}_{[0,1]}$ and $\mathbb{N} \cup \{\aleph_0\}$ are complete lattices.

5 A Strong Confluence Result

In this Section, we will give a strong confluence result in the following form: *any two maximal probabilistic computations P and R with the same root have exactly the same quantitative and qualitative behaviour*, that is to say, the following equations hold for every $C \in \mathsf{NF}$: $\mathcal{P}(P, C) = \mathcal{P}(R, C)$, $\mathcal{N}(P, C) = \mathcal{N}(R, C)$, $\mathcal{P}(P) = \mathcal{P}(R)$, and $\mathcal{N}(P) = \mathcal{N}(R)$.

Remark 5.1 Please notice that equalities like the ones above do *not* even hold for the ordinary lambda calculus. For example, the lambda term $(\lambda x.\lambda y.y)\Omega$ is the root of two (linear) maximal computations, the first having one leaf $\lambda y.y$ and the second having no leaves. This is the reason why the confluence result we prove here is dubbed as strong.

Before embarking in the proof of the equalities above, let us spend a few words to explain their consequences. The fact $\mathcal{P}(P, C) = \mathcal{P}(R, C)$ whenever P and R have the same root can be read as a confluence result: the probability of observing C is independent from the adopted strategy. On the other hand, $\mathcal{P}(P) = \mathcal{P}(R)$ means that the probability of converging is not affected by the underlying strategy. The corresponding results on $\mathcal{N}(\cdot,\cdot)$ and $\mathcal{N}(\cdot)$ can be read as saying that the number of (not necessarily distinct) leaves in any probabilistic computation with root C does not depend on the strategy.

Q^* enjoys a form of *quasi-one-step confluence*. As an example, if $C \to_{\mathscr{N}} D$ and $C \to_{\mathscr{N}} E$ then there is F with $D \to_{\mathscr{N}} F$ and $E \to_{\mathscr{N}} F$. If, on the other hand, $C \to_{\mathscr{N}} D$ and $C \to_{\mathscr{K}} E$ then either $E \to_{\mathscr{N}} D$ or there is F as above. As another interesting example, if $C \to_{\mathscr{N}} D$ and $C \to_{\mathsf{meas}_r} E$ then there is F as above. The only problematic case is when $C \to_{\mathsf{meas}_r} D$ and $C \to_{\mathsf{meas}_r} E$, which cannot be solved. Lack of space prevents us from formally stating and proving quasi-one-step confluence, which can anyway be found in [3]. Quasi-one-step confluence is an essential ingredient towards strong confluence. Unfortunately, quasi-one-step confluence does not translate into an equivalent result on mixed states, because of commutative reduction rules. As a consequence, it is more convenient to first study confluence at the level of probabilistic computations.

We define the *weight* $\mathsf{W}(P)$ and the *branch degree* $\mathsf{B}(P)$ of every *finite* probabilistic computation P by induction on the structure of P:

- $\mathsf{W}([C]) = 0$ and $\mathsf{B}([C]) = 1$.
- $\mathsf{B}([C, P]) = \mathsf{B}(P)$. Moreover, let D be the root of P. If $C \to_{\mathscr{K}} D$, then $\mathsf{W}([C, P]) = \mathsf{W}(P)$, otherwise $\mathsf{W}([C, P]) = \mathsf{B}(P) + \mathsf{W}(P)$.
- $\mathsf{B}([(p, C), P, R]) = \mathsf{B}(P)+\mathsf{B}(R)$, while $\mathsf{W}([(p, C), P, R]) = \mathsf{B}(P)+\mathsf{B}(R)+\mathsf{W}(P)+\mathsf{W}(R)$.

Please observe that $\mathsf{B}(P) \geq 1$ for every P.

Now we propose a probabilistic variation on the classical *strip lemma* of the λ-calculus. It will have a crucial rôle in the proof of strong confluence (Theorem 5.4).

Lemma 5.2 (Probabilistic Strip Lemma) *Let P be a finite probabilistic computation with root C and positive weight $\mathsf{W}(P)$.*

- *If $C \to_{\mathscr{N}} D$, then there is R with root D such that $\mathsf{W}(R) < \mathsf{W}(P)$, $\mathsf{B}(R) \leq \mathsf{B}(P)$ and for every $E \in \mathsf{NF}$, it holds that $\mathcal{P}(R, E) \geq \mathcal{P}(P, E)$, $\mathcal{N}(R, E) \geq \mathcal{N}(P, E)$, $\mathcal{P}(R) \geq \mathcal{P}(P)$ and $\mathcal{N}(R) \geq \mathcal{N}(P)$.*
- *If $C \to_{\mathscr{K}} D$, then there is R with root D such that $\mathsf{W}(R) \leq \mathsf{W}(P)$, $\mathsf{B}(R) \leq \mathsf{B}(P)$ and for every $E \in \mathsf{NF}$, it holds that $\mathcal{P}(R, E) \geq \mathcal{P}(P, E)$, $\mathcal{N}(R, E) \geq \mathcal{N}(P, E)$, $\mathcal{P}(R) \geq \mathcal{P}(P)$ and $\mathcal{N}(R) \geq \mathcal{N}(P)$.*
- *If $C \to^{q}_{\mathsf{meas}_r} D$ and $C \to^{p}_{\mathsf{meas}_r} E$, then there are R and Q with roots D and E such that $\mathsf{W}(R) < \mathsf{W}(P)$, $\mathsf{W}(Q) < \mathsf{W}(P)$, $\mathsf{B}(R) \leq \mathsf{B}(P)$, $\mathsf{B}(Q) \leq \mathsf{B}(P)$ and for every $E \in \mathsf{NF}$, it holds that $q\mathcal{P}(R, E) + p\mathcal{P}(Q, E) \geq \mathcal{P}(P, E)$, $\mathcal{N}(R, E) + \mathcal{N}(Q, E) \geq \mathcal{N}(P, E)$, $q\mathcal{P}(R) + p\mathcal{P}(Q) \geq \mathcal{P}(P)$ and $\mathcal{N}(R) + \mathcal{N}(Q) \geq \mathcal{N}(P)$.*

Proof. By induction on the structure of P. The proof can be found in [3]. □

The following Proposition follows from the probabilistic strip lemma. It can be read as a simulation result: if P and R are maximal and have the same root, then P can simulate R (and vice versa).

Proposition 5.3 *For every maximal probabilistic computation P and for every finite probabilistic computation R such that P and R have the same root, there is a finite sub-computation Q of P such that for every $C \in \mathsf{NF}$, $\mathcal{P}(Q, C) \geq \mathcal{P}(R, C)$ and $\mathcal{N}(Q, C) \geq \mathcal{N}(R, C)$. Moreover, $\mathcal{P}(Q) \geq \mathcal{P}(R)$ and $\mathcal{N}(Q) \geq \mathcal{N}(R)$.*

Proof. The proof goes by induction on $(\mathsf{W}(R), n_R)$, ordered lexicographically. Details can be found in [3]. □

The main theorem is the following:

Theorem 5.4 (Strong Confluence) *For every maximal probabilistic computation P, for every maximal probabilistic computation R such that P and R have the same root, and for every $C \in \mathsf{NF}$, $\mathcal{P}(P, C) = \mathcal{P}(R, C)$ and $\mathcal{N}(P, C) = \mathcal{N}(R, C)$. Moreover, $\mathcal{P}(P) = \mathcal{P}(R)$ and $\mathcal{N}(P) = \mathcal{N}(R)$.*

Proof. See [3]. □

6 Computing with Mixed States

Definition 6.1 A mixed state is a function $\mathscr{M} : \mathbf{Conf} \to \mathbb{R}_{[0,1]}$ such that there is a finite set $S \subseteq \mathbf{Conf}$ with $\mathscr{M}(C) = 0$ except when $C \in S$ and, moreover, $\sum_{C \in S} \mathscr{M}(C) = 1$. **Mix** is the set of mixed states.

In this paper, a mixed state $\mathscr{M}$ will be denoted with the linear notation $\{p_1 : C_1, \ldots, p_k : C_k\}$ or as $\{p_i : C_i\}_{1 \leq i \leq k}$, where p_i is the probability $\mathscr{M}(C_i)$ associated to the configuration C_i, and where $\{C_1, \ldots, C_k\}$ is the set S of the above definition.

Definition 6.2 The reduction relation $\Longmapsto$ between mixed states is defined in the following way: $\{p_1 : C_1, \ldots, p_m : C_m\} \Longmapsto \mathscr{M}$ *iff* there exist m mixed states $\mathscr{M}_1 = \{q_1^i : D_1^i\}_{1 \leq i \in n_1}, \ldots, \mathscr{M}_m = \{q_m^i : D_m^i\}_{1 \leq i \leq n_m}$ such that:

1. For every $i \in [1, m]$, it holds that $1 \leq n_i \leq 2$;
2. If $n_i = 1$, then either C_i is in normal form and $C_i = D_i^1$ or $C_i \to_{n\mathscr{M}} D_i^1$;
3. If $n_i = 2$, then $C_i \to_{\mathsf{meas}_r}^{p} D_i^1$, $C_i \to_{\mathsf{meas}_r}^{q} D_i^2$, $p, q \in \mathbb{R}_{[0,1]}$, and $q_i^1 = p, q_k^2 = q$;
4. $\forall D \in \mathbf{Conf}.\ \mathscr{M}(D) = \sum_{i=1}^{m} p_i \cdot \mathscr{M}_i(D)$.

Given the reduction relation $\Longmapsto$, the corresponding notion of computation (that we call *mixed computation*, in order to emphasize that mixed states play the role of configurations) is completely standard.

Given a mixed state $\mathscr{M}$ and a configuration $C \in \mathsf{NF}$, the *probability* of observing C in $\mathscr{M}$ is defined as $\mathscr{M}(C)$ and is denoted as $\mathcal{P}(\mathscr{M}, C)$. Observe that if $\mathscr{M} \Longmapsto \mathscr{M}'$ and $C \in \mathsf{NF}$, then $\mathcal{P}(\mathscr{M}, C) \leq \mathcal{P}(\mathscr{M}', C)$. If $\{\mathscr{M}_i\}_{i<\varphi}$ is a mixed computation, then

$$\sup_{i<\varphi} \mathcal{P}(\mathscr{M}_i, C)$$

always exists, and is denoted as $\mathcal{P}(\{\mathscr{M}_i\}_{i<\varphi}, C)$.

Please notice that a maximal mixed computation is always infinite. Indeed, if $\mathscr{M} = \{p_i : C_i\}_{1\leq i\leq n}$ and for every $i \in [1, n], C_i \in \mathsf{NF}$, then $\mathscr{M} \Longmapsto \mathscr{M}$.

Proposition 6.3 *Let $\{\mathscr{M}_i\}_{i<\omega}$ be a maximal mixed computation and let $C_1, \ldots, C_n$ be the configurations on which $\mathscr{M}_0$ evaluates to a positive real. Then there are maximal probabilistic computations $P_1, \ldots, P_n$ with roots $C_1, \ldots, C_n$ such that $\sup_{j<\varphi} \mathscr{M}_j(D) = \sum_{i=1}^{n} (\mathscr{M}_0(C_i)\mathcal{P}(P_i, D))$ for every D.*

Proof. See [3]. □

Theorem 6.4 *For any two maximal mixed computations $\{\mathscr{M}_i\}_{i<\omega}$ and $\{\mathscr{M}'_i\}_{i<\omega}$ such that $\mathscr{M}_0 = \mathscr{M}'_0$, the following condition holds: for every $C \in \mathsf{NF}$, $\mathcal{P}(\{\mathscr{M}_i\}_{i<\omega}, C) = \mathcal{P}(\{\mathscr{M}'_i\}_{i<\omega}, C)$*

Proof. A consequence of Proposition 6.3. □

References

[1] P. Arrighi and G. Dowek. Linear-algebraic lambda-calculus: higher-order, encodings, and confluence. In A. Voronkov, editor, *Rewriting techniques and applications*, volume 5117 of *Lecture Notes in Comput. Sci.*, pages 17–31. Springer, 2008.

[2] O. Bournez and C. Kirchner. Probabilistic rewrite strategies. Applications to ELAN. In *Rewriting techniques and applications*, volume 2378 of *Lecture Notes in Comput. Sci.*, pages 252–266. Springer, Berlin, 2002.

[3] U. Dal Lago, A. Masini, and M. Zorzi. Confluence results for a quantum lambda calculus with measurements. 2009. Available from http://arxiv.org/abs/0905.4567.

[4] U. Dal Lago, A. Masini, and M. Zorzi. On a measurement-free quantum lambda calculus with classical control. *Mathematical Structures in Computer Science*, 19(2):297–335, April 2009.

[5] A. Di Pierro, C. Hankin, and H. Wiklicky. *J. Logic Comput.*, 15(2):159–179, 2005.

[6] A. Díaz-Caro, P. Arrighi, M. Gadella, and J. Grattage. Measurements and confluence in quantum lambda calculi with explicit qubits. Available from http://arxiv.org/abs/0806.2447v2, 2008.

[7] P. Kaye, R. Laflamme, and M. Mosca. *An introduction to quantum computing.* Oxford University Press, Oxford, 2007.

[8] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information.* Cambridge University Press, Cambridge, 2000.

[9] S. Roman. *Advanced linear algebra*, volume 135 of *Graduate Texts in Mathematics.* Springer, New York, third edition, 2008.

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

[11] P. Selinger and B. Valiron. A lambda calculus for quantum computation with classical control. *Math. Structures Comput. Sci.*, 16(3):527–552, 2006.

[12] A. Simpson. Reduction in a linear lambda-calculus with applications to operational semantics. In *Term rewriting and applications*, volume 3467 of *Lecture Notes in Comput. Sci.*, pages 219–234. Springer, Berlin, 2005.

[13] Terese. *Term rewriting systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, Cambridge, 2003.

[14] A. van Tonder. A lambda calculus for quantum computation. *SIAM J. Comput.*, 33(5):1109–1135 (electronic), 2004.

[15] P. Wadler. A syntax for linear logic. In *Mathematical foundations of programming semantics (New Orleans, LA, 1993)*, volume 802 of *Lecture Notes in Comput. Sci.*, pages 513–529. Springer, Berlin, 1994.