Measurements and confluence in quantum lambda calculi with explicit qubits

Alejandro Díaz-Caro* Pablo Arrighi† Manuel Gadella‡
Jonathan Grattage†

1 Introduction

In the quest to develop a quantum programming language, quantum extensions of functional languages give a very promising route, hence the explosion of works on quantum lambda calculi and quantum functional languages [1, 4, 9, 10]. Roughly, one could say that the current language proposals can be split into two categories. In the first category the qubits are manipulated as pointers towards a quantum memory [7, 9], thus the syntax does not provide an explicit description of the qubits. It does, however, together with a linear type system, give a convenient and coherent way to handle operations on qubits. A drawback is that the semantics of quantum operations cannot be given intrinsically in the syntax, as this would require the actual state of the quantum memory. In the second category [1, 4, 10] the description of the qubits is part of the programming language, and no type system is required. An advantage here is that the entire semantics can be expressed simply as a rewrite system between terms of the language. This turns into a weakness when it comes to measurements, because the inherently probabilistic nature of measurement makes it difficult to express as a rewrite system. In fact, neither of the languages [4, 10] allow this feature.

The case of Altenkirch and Grattage's *QML* [1] is not so clear-cut, but it does illustrate this difficulty: *QML* includes measurements with an operational semantics given in terms of quantum circuits. However, the corresponding algebraic theory [2] stands only for a pure quantum subset of the language, with classical-control and measurement omitted.

Van Tonder's λ_q [10] is a higher-order untyped lambda calculus which includes quantum properties. This calculus carries a history track to keep the necessary information to revert reductions, as a proof that the global computation process is unitary. It is closely related to linear logic, with the syntax being a fragment of the one introduced by Wadler [11], extended with constants to represent quantum entities such as qubits and gates. Linearity concepts are used to distinguish definite terms from arbitrary superposition terms. These syntactic markers constitute the main difference with Arrighi and Dowek's *Lineal* [3, 4], which is more permissive. As mentioned previously, measurement is not included in these two proposals.

Here it is shown how to add measurement to a quantum lambda calculus with explicit qubits, in an elegant manner. This is done with full details for the λ_q-calculus, with a proof that confluence, and hence the consistency of the operational semantics, is preserved by this extension. The methods illustrated here are general, and applying these techniques to *QML* and *Lineal* is currently ongoing.

2 Motivations

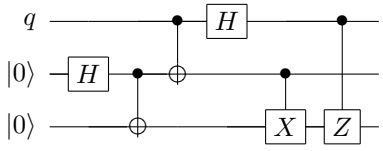
In contrast to measurement in classical mechanics, which gives the value of a given observable with an associated error, measurements in quantum mechanics have an intrinsically probabilistic character. That is, a quantum measurement can give, *a priori*, a certain number of results, each one

*Department of Computer Science, Universidad Nacional de Rosario, Argentina. diazcaro@fceia.unr.edu.ar
†LIG, Université de Grenoble and CNRS, France. {arrighi,grattage}@imag.fr
‡Department of Theoretical Physics, Universidad de Valladolid, Spain. gadella@fta.uva.es

with some finite probability. Moreover, the state of the system after the measurement is changed in an irreversible manner by the measurement. This unusual behaviour is of extreme relevance in quantum information processing. Firstly, measurement is a key property in many quantum information processing tasks, such as quantum cryptography, superdense coding, and in quantum search. Not having measurements can lead to misinterpretation of algorithms; consider for instance the quantum teleportation algorithm with deferred measurement [10], as defined as in Figure 1.

Secondly, understanding measurement is essential in order to avoid misinterpreting quantum computation as a whole (e.g. why quantum computation does not lead straightforwardly to an exponential jump in complexity). This paper takes the view that in order to grasp the resources and limitations of quantum computation, measurement needs be formalised in an elegant manner. Note that the projective measurement discussed in this paper is not the only possibility for a quantum measurement, but it is the simplest. In addition, any quantum measurement can be reproduced by the action of a unitary mapping and a projective measurement.

$$
\begin{array}{l}
\mathsf{teleport}\ q \rightarrow\ \textbf{let}\ (e_1, e_2) = \mathsf{epr}\ \textbf{in} \\
\qquad\qquad\qquad\ \textbf{let}\ (q', y') = \mathsf{alice}\ (q, e_1)\ \textbf{in} \\
\qquad\qquad\qquad\ \mathsf{bob}\ (q', y', e_2) \\
\text{where} \\
\mathsf{alice}\ (q, e_1) \rightarrow\ \textbf{let}\ (q', y') = \mathit{cnot}\ (q, e_1)\ \textbf{in} \\
\qquad\qquad\qquad\ ((H\ q'), y') \\
\mathsf{bob}\ (q', y', e_2) \rightarrow\ \textbf{let}\ (y'', e_2') = cX\ (y', e_2)\ \textbf{in} \\
\qquad\qquad\qquad\qquad\ \textbf{let}\ (q'', e_2'') = cZ\ (q', e_2')\ \textbf{in} \\
\qquad\qquad\qquad\qquad\ (q'', y'', e_2'') \\
\mathsf{epr} \equiv \mathit{cnot}\ ((H\ 0), 0)
\end{array}
$$



Circuit for the quantum teleportation algorithm with deferred measurement

Figure 1: Teleportation algorithm in non-extended λ_q. Here it is unclear if Alice and Bob can be physically separated, as all the channels used are quantum channels. An obvious question arises: why use this algorithm if there is a quantum channel between Alice and Bob? Measuring the final state will result, as expected, with the original logical qubit being moved to Bob. The problem is not one of correctness, but of interpretation.

To account for measurements, the grammar of λ_q must be extended with a family of measurement operators M_I, which measure the qubits indicated by the set I, and the syntax of the qubits has to be made precise, following the style of *Lineal* [4] and *QML* [1]. These are minor changes to the existing grammar of λ_q [10], which are fully developed in the full paper, but here the terms will be taken as self-explanatory. As already explained, measurement is an inherently probabilistic operation. Following Di Pierro *et al.* [5], where a probabilistic rewrite system is defined over a λ-calculus, the operational semantics for measurement is defined in Figure 2. With this rule the teleportation algorithm can be rewritten as shown in Figure 3, to restore the correct interpretation.

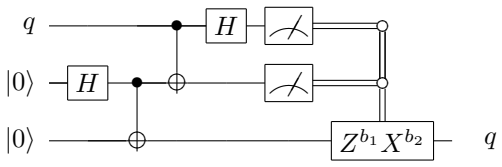
$$
\text{M:}\ \frac{q = \sum_{i=0}^{2^m-1} \alpha_i q_i}{\mathcal{H}; (M_I\ q) \rightarrow_{p_j} \sum_{i \in C_j} \frac{\alpha_i}{\sqrt{p_j}} q_i}\ \ |I| \leq m
$$

where:

- $q_i = !q_{1i} \otimes !q_{2i} \otimes \cdots \otimes !q_{mi}$ with $!q_{ki} = !\,|0\rangle$ or $!\,|1\rangle$ for $k = 1 \ldots m$.
- $C_j = \{i \mid bin(i, j, I), i = 0, \ldots, 2^m - 1\}$ for $j = 0, \ldots, 2^{|I|} - 1$; where, if i_k (resp. j_k) is the k^{th} bit in the binary representation of i (j), then $bin(i, j, I)$ is *true* iff $i_{k_h} = j_h,\ \forall h = 1, \ldots, |I|,\ k_h \in I, k_h < k_{h+\gamma}\ \forall \gamma > 0$
- $p_j = \sum_{i \in C_j} |\alpha_i|^2$.

Figure 2: Operational semantic for measurement. The index on the arrow is the probability of this transition, and I gives the index of qubits to be measured.

$\mathsf{teleport}\ q \to_1$ **let** $x \otimes y = \mathsf{epr}$ **in**
let $b_1 \otimes b_2 = M_{\{1,2\}}\ \mathsf{alice}\ (q, x)$ **in**
$\mathsf{bob}\ (b_1, b_2, y)$
where
$\mathsf{alice}\ (q, x) \to_1$ **let** $r \otimes w = cnot\ q \otimes x$ **in**
$((H\ r) \otimes w)$
$\mathsf{bob}\ (b_1, b_2, y) \to_1 (\mathsf{zed}\ b_1)\ (\mathsf{ex}\ b_2)\ y$
$\mathsf{epr} \equiv cnot\ ((H\ !\,|0\rangle \otimes !\,|0\rangle)$



Circuit for the original quantum teleportation algorithm

Figure 3: Teleportation algorithm in extended λ_q. The functions ex and zed are not shown here. $(\mathsf{ex}\ b_2)$ is X^{b_2} and $(\mathsf{zed}\ b_1)$ is Z^{b_1}.

When defining a language, its grammar must be provided (how to construct terms), and its semantics (how these terms compute). The semantics can be denotational (terms are mapped to elements of a semantic domain, each corresponding to what is computed by the term) or operational (terms are mapped into other terms, each transition corresponding to a computational step). Clearly, when providing semantics for a language, it must be proved that the semantics is unambiguous and consistent. For example, the semantics will usually induce an equational theory upon terms (via equality in the semantics domain or by equating two terms if one reduces to the other), and it is important that this theory should not equate all terms.

In λ_q a consistent equational theory is given. However, adding measurement does not correspond to a simple system for equational reasoning. It is not possible to proceed by replacing terms by equal terms according to any equational theory, as measurement is a probabilistic operation; each reduction instance could produce different terms that are impossible to reconcile in the system. In the presence of an operational semantics, a usual method of proving the consistency result is to provide a proof of confluence. This property states that the order in which the transition rules are applied does not matter to the end result, removing any ambiguity. In the following section it is shown how such a study of confluence can still be carried through, even in the presence of probabilities.

3 Confluence

Whilst the above-mentioned probabilistic reductions are an elegant and synthetic way to present the operational semantics, the study of the confluence is not immediate in this setting. For confluence, it is necessary to prove that if any term t can reduce to u and to v, then there exists a w such that $u \to w \wedge v \to w$. However, in a probabilistic calculus it could be that $t \to_p u$ and $t \to_q v$, where p and q represent the probability of the respective reduction occurring, and there is no w that both u and v could reduce to. For example, given $M_{\{1\}}$, a measurement operator in the computational basis, it follows that $M_{\{1\}}\ (\alpha\,|0\rangle + \beta\,|1\rangle) \to_{|\alpha|^2} |0\rangle$ and $M_{\{1\}}\ (\alpha\,|0\rangle + \beta\,|1\rangle) \to_{|\beta|^2} |1\rangle$. However, there is no w such that $|0\rangle \to_p w$ and $|1\rangle \to_q w$.

A naïve way to deal with this would be to assume that if there is some normal form that can be reached with a certain probability, then by following any path it must to be possible to reach the same normal form with the same probability. However, this definition is not rigorous, and it is not applicable to terms without a normal form. Hence, it does not allow a formal proof of confluence.

Probabilistic transitions need to abstracted out, in order to allow only one possible normal form for each term and to deal with terms without normal form. With this aim, the following definition gives a notion of confluence for probabilistic calculi:

Definition 1 *A term ensemble* $\{< t_i, p_i >\}$ *is defined as a collection of terms* t_i*, each with an associated probability, such that the terms* t_i *are the same symbol-by-symbol, while allowing qubit-constants to differ, with* $\sum_i p_i = 1$.

Example 1 *Consider* $\{< (\lambda x.!\,|0\rangle), \frac{1}{2} >, < (\lambda x.!\,|0\rangle), \frac{1}{4} >, < (\lambda x.!\,|1\rangle), \frac{1}{4} >\}$. *Note that the term*

$(\lambda x.!\,|0\rangle)$ *appears twice. By summing the probabilities of the equivalent terms, this ensemble is equivalent to* $\{<(\lambda x.!\,|0\rangle),\frac{3}{4}>,<(\lambda x.!\,|1\rangle),\frac{1}{4}>\}$.

A deterministic transition rule between term ensembles can be defined:

Definition 2 *If X is a probabilistic rewrite system over terms, let $Det(X)$ be the non-probabilistic rewrite system over term ensembles written $\overset{X}{\twoheadrightarrow}$ and defined as*

$$\{<t_i,p_i>\} \overset{X}{\twoheadrightarrow} \{<t'_j,p'_j>\}, \quad \textit{if and only if } \ (t_{i_1} \rightarrow_{q_1} t'_j) \wedge \cdots \wedge (t_{i_n} \rightarrow_{q_n} t'_j) \textit{ and } p'_j = \sum_{k=1}^{n} p_{i_k} q_k$$

where all the reductions between single terms are produced by following the same rule in X.

In fact, the appropriate steps so that $\{<t,p>,<t,q>\}$ is identified with $\{<t,p+q>\}$ need to be taken, but these technical details are left for the full paper.

Example 2 *Given the probabilistic rewrite system $T=\{(M)\}$,*

$$\{<M_{\{1\}}\ (\frac{1}{\sqrt{2}}!\,|0\rangle + \frac{1}{\sqrt{2}}!\,|1\rangle),1>\} \overset{T}{\twoheadrightarrow} \{<!\,|0\rangle\,,\frac{1}{2}>,<!\,|1\rangle\,,\frac{1}{2}>\}.$$

Using these concepts, (strong) confluence for a probabilistic rewrite system can be expressed as follows:

Definition 3 *Let R be a probabilistic rewrite system. R is said to be confluent if for each term ensemble τ such that $\tau \overset{R^*}{\twoheadrightarrow} \mu \wedge \tau \overset{R^*}{\twoheadrightarrow} \nu$, there exists a term ensemble ω such that $\mu \overset{R^*}{\twoheadrightarrow} \omega \wedge \nu \overset{R^*}{\twoheadrightarrow} \omega$. R is said to be strongly confluent if for each term ensemble τ such that $\tau \overset{R}{\twoheadrightarrow} \mu \wedge \tau \overset{R}{\twoheadrightarrow} \nu$, there exists a term ensemble ω such that $\mu \overset{R}{\twoheadrightarrow} \omega \wedge \nu \overset{R}{\twoheadrightarrow} \omega$.*

Note that strong confluence of R implies the confluence of R, and also that the confluence of R implies the strong confluence of R^*. It is possible to extend the Hindley-Rosen lemma [6, 8] to these notions of confluence, as follows:

Proposition 1 *Let R and U be strongly confluent probabilistic rewrite systems. If R and U strongly commute, that is if for each term ensemble τ such that $\tau \overset{R}{\twoheadrightarrow} \mu \wedge \tau \overset{U}{\twoheadrightarrow} \nu$, there exists a term ensemble ω such that $\mu \overset{U}{\twoheadrightarrow} \omega \wedge \nu \overset{R}{\twoheadrightarrow} \omega$, then $R \cup U$ is strongly confluent.*

The strong confluence of the additional rule $T=\{(M)\}$ is formally expressed and proved by the following theorem:

Theorem 1 *The probabilistic reduction rules system $T=\{(M)\}$ is strongly confluent.*

Proof. (Outline) It needs to be shown that measurements can be applied in any order, and this is achieved using structural induction over τ. The most interesting case is the following, where $\tau=\{<(t_{i1}\ t_{i2}),p_i>\}$, in which several subcases must be considered. The most interesting and relevant of these subcases are outlined below:

- Let $\mu=\{<(t_{i1}\ u_i),p'_i>\}$ and $\nu=\{<(t_{i1}\ v_i),p''_i>\}$, where $t_{i2} \rightarrow_{q_i} u_i \wedge p'_i = q_i p_i$ and $t_{i2} \rightarrow_{r_i} v_i \wedge p''_i = r_i p_i$. Notice that the index i is retained, as different terms may rewrite to the same u_i, v_i, as t_{i1} itself can be different. By the induction hypothesis, there exists $\omega=\{<w_j,p'''_j>\}$ such that $\{<u_i,p'_i>\} \overset{T}{\twoheadrightarrow} \omega$ and $\{<v_i,p''_i>\} \overset{T}{\twoheadrightarrow} \omega$. There are several interesting details to this case, as the different w_j have to be split in order to attach each t_{i1}. To do this, a concept of equivalence between term ensembles is required to prove that if $\mu \overset{T}{\twoheadrightarrow} \omega_1$ and $\nu \overset{T}{\twoheadrightarrow} \omega_2$, such that $\omega_1 \equiv \omega_2$, then it holds to have one ω. This proof is provided in the full paper. To simplify this presentation, we suppose as an induction hypothesis that there exists an $\omega_1=\{<w_i,p'''_i>\}$ such that $\{<u_i,p'_i>\} \overset{T}{\twoheadrightarrow} \omega_1$ and $\{<v_i,p''_i>\} \overset{T}{\twoheadrightarrow} \omega_1$, so we can take $\omega=\{<(t_{i1}\ w_i),p'''_i>\}$. Many cases follow this pattern, i.e. when the reduction is by congruence within a subterm, the induction hypothesis is used.

- Let $\tau = \{<(M_I\ q_i), p_i>\}$, $\mu = \{<q'_j, p'_j>\}$ where $(M_I\ q_i) \xrightarrow{T}_{r_j} q'_j$ and $p'_j = r_j p_i$ then there exists no ν different to μ. The fact that measurements only apply to qubit-terms is the basis of this proof.

□

Theorem 2 *The probabilistic rewrite systems* $S = \{(APP_1),\ (APP_2), (\beta), (!\beta_1),\ (!\beta_2), (U), (Id)\}$ *of the original* λ_q*, and* $T = \{(M)\}$ *strongly commute.*

In fact, the previous theorem ensures that adding measurement preserves the confluence of the calculus. Since S and T strongly commute, so too do S^* and T. Also, given that S is confluent, then S^* is strongly confluent. As a consequence, $S^* \cup T$ is strongly confluent, which entails the confluence of $S \cup T$.

Proof. (Outline) By structural induction in a similar case-by-case study.

Rather than detailing this case study, some key examples are presented:

- *Cloning* of arguments: $(\lambda x.(x\ x))\ (M_{\{1\}}\ (\frac{1}{\sqrt{2}}!\,|0\rangle + \frac{1}{\sqrt{2}}!\,|1\rangle))$

  The problem here is that if copying a measurement is allowed, this may give different results for each measurement. However, by measuring first and then applying the abstraction, both measurements are the same. In λ_q, these kinds of terms are disallowed by the well-formedness rules; a linear argument can appear only once in the body of a function.

- *Copying* of arguments: $(\lambda!x.(x\ x))\ (M_{\{1\}}\ (\frac{1}{\sqrt{2}}!\,|0\rangle + \frac{1}{\sqrt{2}}!\,|1\rangle))$

  When the argument is linear, there is no rule in the operational semantics of λ_q that allows the application of a non-linear abstraction to a linear term. Hence, $M_{\{1\}}$ must apply first, producing a non-linear output (either $!\,|0\rangle$ or $!\,|1\rangle$).

- *Promotion* of the argument: $(\lambda!x.(x\ x))\ !(M_{\{1\}}\ (\frac{1}{\sqrt{2}}!\,|0\rangle + \frac{1}{\sqrt{2}}!\,|1\rangle))$

  In this case copying the measurement operation twice is allowed, and this is the only applicable reduction strategy because of !-suspension.

□

4 Conclusions

In this paper we have extended the quantum lambda calculus λ_q, defined by van Tonder, with a family of measurement operations M_I, which measure the qubits indicated by the set I. By defining the notion of ensembles of terms, and extending the rewrite system to a deterministic system between term ensembles, a proof of confluence for this extended calculus is presented. The extended calculus is therefore confluent, and retains the simplicity of van Tonder's original calculus.

The proof of confluence follows a method which can be applied to other calculi that make use of probabilistic transition rules. For example, this method could be applied to both *Lineal* and to *QML*, and this is the subject of ongoing research.

The addition of a measurement operation to λ_q, which preserves confluence, is a significant development. This allows a more natural expression of quantum algorithms that intrinsically make use of measurement, such as quantum teleportation, superdense coding, and quantum search algorithms.

Acknowledgements

D.C. would like to thank Pablo E. Martínez López for useful comments and helpful suggestions on an early draft of this paper, and thank the QCG group at the Laboratoire d'Informatique de Grenoble for their hospitality. The authors also thank Simon Perdrix for fruitful discussions.

References

[1] T. Altenkirch and J. J. Grattage. A functional quantum programming language. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Computer Society, 2005.

[2] T. Altenkirch, J. J. Grattage, J. K. Vizzotto, and A. Sabry. An algebra of pure quantum programming. *Electronic Notes in Theoretical Computer Science*, 170:23–47, 2007.

[3] P. Arrighi and G. Dowek. A computational definition of the notion of vectorial space. *Electronic Notes in Theoretical Computer Science*, 117:249–261, 2005.

[4] P. Arrighi and G. Dowek. Linear-algebraic lambda-calculus: higher-order, encodings and confluence. Submitted to RTA'08. `arXiv:quant-ph/0612199`, 2006.

[5] A. Di Pierro, C. Hankin, and H. Wiklicky. Probabilistic λ-calculus and quantitative program analysis. *Journal of Logic and Computation*, 15(2):159–179, 2005.

[6] J. R. Hindley. *The Church-Rosser property and a result in combinatory logic*. PhD thesis, University of Newcastle-upon-Tyne, 1964.

[7] F. Prost. Taming non-compositionality using new binders. In *Proceedings of Unconventional Computation 2007 (UC'07)*, volume 4618 of *Lecture Notes in Computer Science*. Springer, 2007.

[8] B. K. Rosen. Tree-manipulating systems and church-rosser theorems. *Journal of the ACM*, 20(1):160–187, Jan. 1973.

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

[10] A. van Tonder. A lambda calculus for quantum computation. *SIAM Journal on Computing*, 33(5):1109–1135, 2004.

[11] P. Wadler. A syntax for linear logic. In *Proceedings of the 9th International Conference on Mathematical Foundations of Programming Semantics*, pages 513–529, London, UK, 1994. Springer-Verlag.