Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 270 (1) (2011) 121–128
www.elsevier.com/locate/entcs

On Quantum and Probabilistic Linear Lambda-calculi (Extended Abstract)

Benoît Valiron[1]

Laboratoire d'Informatique de Grenoble,
Université Joseph Fourier, Grenoble, France

Abstract

In this paper we give a fully complete model for a linear probabilistic lambda-calculus. The model is a Kripke semantics based on the category of stochastic relations. We sketch how this relates to quantum computation.

Keywords: Probabilistic computation, quantum computation, Bell inequalities, linear lambda-calculus, Kripke semantics, completely positive maps, multinorm.

1 Introduction

Selinger and Valiron [6] gave a fully-abstract semantics for a linear quantum lambda-calculus using the category **CPM** of completely positive maps. This semantics is not surjective, and thus fails to distinguish between maps in the category that are denotations of a program and maps that are not.

A surjective semantics for the first-order case was provided in [4]. The semantics uses a notion of norm to determine which completely positive maps are denotations of terms; namely, these are precisely the trace non-increasing ones. However, the notion of norm fails to provide a suitable characterization of higher-order functions [5].

In this paper, we restrict our study to the probabilistic fragment of the language described in [6]. We characterize the image of the language using the notion of polytopes, and sketch how one can use such a result to characterize the information-theoretic power of quantum computation over probabilistic computation. Finally we provide a full and complete denotational semantics for a fragment of the probabilistic linear lambda-calculus using an idea from Jung and Tiuryn [3].

[1] Email: benoit.valiron@imag.fr

doi:10.1016/j.entcs.2011.01.011

2 The linear quantum lambda-calculus

We recall the linear typed lambda calculus for quantum computation defined in [6]. The terms and the types are respectively the following:

$$\begin{aligned} M, N, P ::= {} & x \mid \lambda x.M \mid MN \mid \langle M, N\rangle \mid * \mid \Omega \mid \ let\ \ \langle x, y\rangle = M\ in\ N \mid \\ & let * = M\ in\ N \mid if P\ then\ M\ else\ N \mid 0 \mid 1 \mid \ meas\ \mid\ new\ \mid U, \\ A, B \quad ::= {} & A \otimes B \mid \top \mid A \multimap B \mid bit \mid qbit, \end{aligned}$$

where x ranges over term variables. We remind the reader that Ω corresponds to the diverging term and we refer to [6] for the definition of the typing judgements.

This language is interpreted in the category **CPM** of completely positive maps [4]. The types are interpreted in the following way.

$$[\![bit]\!] = 1, 1, \qquad [\![qbit]\!] = 2, \qquad [\![A \otimes B]\!] = [\![A \multimap B]\!] = [\![A]\!] \otimes [\![B]\!].$$

In this extended abstract we only recall the definition of the Booleans 0 and 1 and refer the reader to [6] for the interpretation of all the terms: $[\![0]\!] = (1, 0) \in \mathbb{C}^2$, $[\![1]\!] = (0, 1) \in \mathbb{C}^2$.

Using the language, it is possible to simulate a fair coin toss, for example with the program $meas(H(new\, 0)) : \top \multimap bit$, with H corresponding to the Hadamard gate. The interpretation of this term is $(\frac{1}{2}, \frac{1}{2})$. Using a similar technique (and possibly using the term Ω), it is possible to get a term simulating an unfair coin with denotation (a, b) where $a, b \geqslant 0$ and $a + b \leqslant 1$.

3 A probabilistic linear lambda calculus

We can modify the language of Section 2 to remove the quantum aspects and obtain a purely probabilistic language. For this, we only need to change the constants. We replace $meas$, new and U with a set of constants $c(a, b)$, one for each pair of real numbers (a, b) such that $a, b \geqslant 0$ and $a + b \leqslant 1$. The types are modified by removing the type $qbit$.

We are left with the following terms and types:

$$\begin{aligned} M, N, P ::= {} & x \mid \lambda x.M \mid MN \mid \langle M, N\rangle \mid * \mid \Omega \mid \ let\ \ \langle x, y\rangle = M\ in\ N \mid \\ & let * = M\ in\ N \mid if P\ then\ M\ else\ N \mid c(a, b) \mid 0 \mid 1, \\ A, B \quad ::= {} & A \otimes B \mid \top \mid A \multimap B \mid bit. \end{aligned}$$

In this context, the semantics in **CPM** of a term M is a completely positive map from vectors of $\mathbb{C}^m$ to vectors of $\mathbb{C}^n$ for some natural numbers m and n. Note that a completely positive map in this setting is the same as a (partial) stochastic map from $[0, 1]^m$ to $[0, 1]^n$. The result presented in [6] specializes to this case and the semantics is fully abstract.

4 Interpretation of the Bell inequalities

The Bell experiment [1] is the following. Consider a quantum machine that maximally entangles two quantum bits A and B

$$|\phi_{AB}\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle)$$

and sends qubit A to Alice and qubit B to Bob. Suppose that Alice and Bob can independently choose one of the following three bases $\{a, b, c\}$ for measuring their quantum bits:

$$|0_a\rangle = |0\rangle, \qquad |0_b\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle, \qquad |0_c\rangle = \frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle,$$
$$|1_a\rangle = |1\rangle \qquad |1_b\rangle = \frac{\sqrt{3}}{2}|0\rangle - \frac{1}{2}|1\rangle, \qquad |1_c\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle.$$

The question is to compute the probability of obtaining the same output when measuring A and B with respect to two different bases.

One can interpret this experiment in the context of higher-order quantum computation. First, the machine entangles two quantum bits: It produces a map $\mathbf{EPR} : \top \to \mathit{qbit} \otimes \mathit{qbit}$. Then Alice and Bob each takes one qubit, chooses a basis, and measures: the measurement they perform is then a function $\mathbf{f} : \mathit{qbit} \otimes \mathit{trit} \to \mathit{bit}$, where $\mathit{trit} = \top \oplus \top \oplus \top$. One can curry this function to $\mathbf{f}' : \mathit{qbit} \to (\mathit{trit} \multimap \mathit{bit})$

The algorithm can be described by the composition

$$\top \xrightarrow{\mathbf{EPR}} \mathit{qbit} \otimes \mathit{qbit} \xrightarrow{\mathbf{f}' \otimes \mathbf{f}'} (\mathit{trit} \multimap \mathit{bit}) \otimes (\mathit{trit} \multimap \mathit{bit}).$$

The algorithm produces a term of type $(\mathit{trit} \multimap \mathit{bit}) \otimes (\mathit{trit} \multimap \mathit{bit})$. This type is classical, and one can wonder whether the denotation of this term is the denotation of a term in the probabilistic linear calculus. The Bell inequalities precisely show that it is not the case. In other words, the terms expressible in the probabilistic linear lambda calculus correspond precisely to what is known to physicists as local hidden variable theories.

Consider a classical type A (that is, not containing qbit). In the remainder of this paper, we shall develop a methodology for determining the vectors of $[\![A]\!]$ that are representable by a term in the probabilistic linear calculus. This can be seen as a generalization of the Bell inequalities to any higher-order type.

5 Factorization of the probabilistic calculus

As in [2], a program written in the probabilistic linear lambda-calculus of Section 3 can be "factored" into a probabilistic sum of deterministic programs. That is, the denotation of any valid typing judgement $\Delta \rhd M : A$ can be re-written as a probabilistic sum

$$[\![\Delta \rhd M : A]\!] = \sum_i \alpha_i [\![\Delta \rhd N_i : A]\!]$$

$$\begin{aligned}
&[\![c(a,b)]\!] &&= [\![if\, c(a,b)\ then\ 1\ else\ 0]\!]\\
&[\![M(if\, c(a,b)\ then\ N_1\ else\ N_2)]\!] &&= [\![if\, c(a,b)\ then\ MN_1\ else\ MN_2]\!]\\
&[\![\langle M, if\, c(a,b)\ then\ N_1\ else\ N_2\rangle]\!] &&= [\![if\, c(a,b)\ then\ \langle M, N_1\rangle\ else\ \langle M, N_2\rangle]\!]\\
&[\![\lambda x.if\, c(a,b)\ then\ N_1\ else\ N_2]\!] &&= [\![if\, c(a,b)\ then\ \lambda x.N_1\ else\ \lambda x.N_2]\!]\\
&[\![if\,(if\, c(a,b)\ then\ M\ else\ N)\ then\ P\ else\ Q]\!]\\
&\qquad\qquad = [\![if\, c(a,b)\ then\ if\, M\ then\ P\ else\ Q\ else\ if\, N\ then\ P\ else\ Q]\!]
\end{aligned}$$

Table 1
Rewriting rules for the *if*-term.

where each $\alpha_i \geqslant 0$ and $\sum_i \alpha_i \leqslant 1$, and where N_i does not contain any constant term $c(a,b)$.

The idea of the proof is the following: first, note that

$$[\![\Delta \rhd if\, c(a,b)\ then\ M\ else\ N : A]\!] = b \cdot [\![\Delta \rhd M : A]\!] + a \cdot [\![\Delta \rhd N : A]\!].$$

Thus, for the result to be true, one needs to be able to send a term M containing a constant term $c(a,b)$ to a term of the form $if\, c(a,b)\ then\ M_1\ else\ M_2$ with same denotation, where M_1 and M_2 satisfy some invariant. Since the language is linear, this is possible. As a proof start, consider the equalities in Table 1 with orientation left-to-right.

6 Interpretation as polytopes

The deterministic terms (i.e. the ones with no occurrence of $c(a,b)$) share a special property:

Proposition 6.1 *Given a typing context Δ and a type A, there is a finite number of deterministic terms N_i (up to beta-equivalence) such that $\Delta \rhd N_i : A$.*

This proposition comes from the fact that the language is linear: one can enumerate all the possible (CUT-free) typing derivations. One can go one step further:

Proposition 6.2 *Any deterministic closed term of type A has a denotation of the form $(x_1^i, \ldots, x_n^i) \in [\![A]\!]$, where for all j, x_j^i is either 0 or 1.*

It allows us to state the following theorem:

Theorem 6.3 *The set of vectors in the set $[\![A]\!]$ that are the image of some linear probabilistic program through the denotation is a convex 0-1-polytope (i.e. whose vertices are of the form $(x_1, \ldots, x_n) \in [\![A]\!]$, where each x_i is either 0 or 1).*

A few examples are given below:

(i) The deterministic closed terms of type $[\![bit]\!]$ are Ω, 0 and 1. That is, the polytope of admissible vectors is spanned by $(0,0)$, $(1,0)$ and $(0,1)$.

(ii) The deterministic closed terms of type $[\![bit \multimap \top]\!]$ are Ω, $\lambda x.if\, x\ then * else\ \Omega$, $\lambda x.if\, x\ then\ \Omega\ else *$ and $\lambda x.if\, x\ then * else *$. That is, the polytope of admissible vectors is spanned by $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$.

(iii) The deterministic closed terms of type $[\![(bit \multimap \top) \multimap \top]\!]$ are of the form Ω, $\lambda f.\ let * = f0\ in *$ and $\lambda f.\ let * = f0\ in *$. That is, the polytope of admissible vectors is spanned by $(0,0)$, $(1,0)$ and $(0,1)$.

7 Polytopes are not compositional

The naive idea to develop a full and complete semantics for the probabilistic linear lambda-calculus is the following. Consider a category whose objects are polytopes and whose maps are CPM maps sending polytopes into polytopes. Then interpret the language in this category.

It turns out that this does not work. Indeed, if it were the case, the identity map $\mathbb{C}^2 \to \mathbb{C}^2$ should be represented by a judgement $x : (bit \multimap \top) \multimap \top \rhd M : bit$. However, the only deterministic closed terms corresponding to $((bit \multimap \top) \multimap \top) \multimap bit$ are of the form $\lambda f.\ let * = f(\lambda x.if\, x\ then\ a\ else\ b)\ in\ c$ where c ranges over $\{0,1\}$ and where a and b over $\{*, \Omega\}$. The corresponding vertices are

$$(0,0,0,0),\ (1,0,0,0),\ (1,0,1,0),\ (0,0,1,0),\ (0,1,0,1),\ (0,1,0,0),\ (0,0,0,1).$$

The identity map corresponds to the vector $(1,0,0,1)$, which is not in the polytope spanned by these vectors. The closest approximation is $\frac{1}{2}(1,0,0,1)$.

Thus, although polytopes provide a sound semantics, they are not sufficient to characterize the definable maps of the probabilistic language.

8 Toward a full and complete semantics for the probabilistic linear calculus

The polytopes we constructed are closed convex sets P containing the origin. Thus they define a norm via

$$\|v\| = \min\{\lambda \mid v \in \lambda P\}.$$

We are going to extend the idea of norm to the work of Jung and Tiuryn [3], and define what we will call linear Kripke logical relations: instead of considering norms of single vectors, we consider the norm of tuples of vectors.

8.1 A toy language

Consider the following reduced version of the probabilistic linear lambda-calculus:

$$\begin{aligned} &Term\ \ M, N ::= x^A \mid \Omega^A \mid c(\mu) \mid \lambda x^A.M \mid MN \mid if P\ then\ M\ else\ N,\\ &Type\ \ \ A, B\ \ ::= bit \mid A \multimap B, \end{aligned}$$

where $0 \leqslant \mu \leqslant 1$.

8.2 Linear Kripke logical relations

Consider a small category of sets $\mathcal{C}$ with the cartesian product as a monoidal structure, containing the object $\top = \{\star\}$. We are going to define a logical relation over each object w of $\mathcal{C}$, as a norm $\| - \|_A^w$ on tuples $(x_i)_{i\in w}$ where $x_i \in [\![A]\!]$ for all $i \in w$. The relation at ground type must satisfy the following compatibility property: if $f : v \to w$ is a map in $\mathcal{C}$, then

$$\|(x_i)_{i\in w}\|_{bit}^w \leqslant 1 \quad \to \quad \|(x_{f(i)})_{i\in v}\|_{bit}^v \leqslant 1.$$

Also, still at ground type *bit*, the norm should satisfy the norm axioms:

$$\begin{aligned}
\|(x_i + y_i)_i\|_{bit}^w &\leqslant \|(x_i)_i\|_{bit}^w + \|(y_i)_i\|_{bit}^w,\\
\|(x_i)_i\|_{bit}^w &\geqslant 0,\\
\|(\lambda x_i)_i\|_{bit}^w &= |\lambda|\|(x_i)_i\|_{bit}^w,\\
\|(x_i)_i\|_{bit}^w &= 0 \quad \text{iff} \quad \forall i \quad x_i = 0.
\end{aligned}$$

Finally, one should have $\|((a,b))\|_{bit}^\top = |a| + |b|$ We extend the relation for higher types as follows: $\|(g_j)_{j\in w}\|_{A\multimap B}^w$ is defined as the maximum of $\|(g_{f(j)}(x_i))_{j\otimes i\in w'\otimes v}\|_B^{w'\otimes v}$, where $f : w' \to w \in \mathcal{C}$, $(x_i)_{i\in v} \in [\![A]\!]$ such that $\|(x_i)_{i\in v}\|_A^v \leqslant 1$.

We interpret $(g_{f(j)}(x_i))_{j\otimes i\in w'\otimes v}$ as the tuple $(h(k))_{k\in w'\otimes v}$ where h is the map

$$w' \otimes v \xrightarrow{f\otimes id} w \otimes v \xrightarrow{g\otimes x} [\![A \multimap B]\!] \otimes [\![A]\!] \xrightarrow{\epsilon_{A,B}} [\![B]\!]$$

We will write $\|x\|_A$ for $\|(x)\|_A^\top$. We call $(\| - \|_A^w)_{w\in|\mathcal{C}|, A\in Type}$ a *norm with varying arity.*

Lemma 8.1 *Given $w \in |\mathcal{C}|$ and a type A, the norm $\| - \|_A^w$ satisfies the compatibility property and the norm axioms.*

8.3 Soundness

Lemma 8.2 (Soundness) *For all closed terms $M : A$, $\|[\![M]\!]\|_A \leqslant 1$.*

The proof follows the idea developed in [3]. We define the notion of *extended environment* as a pair (ϕ, ρ) of partial maps, where $\phi : Var \to |\mathcal{C}|$ and where ρ is a map that assigns to a variable x^A a tuple $(x_i)_{i\in\phi(x^A)}$, with $x_i \in [\![A]\!]$. If $\Delta = \{x_1 : A_1, \ldots, x_n : A_n\}$, we write $\phi(\Delta)$ in place of $\phi(x_1) \otimes \cdots \otimes \phi(x_n)$. We define the *extended interpretation* of a typing judgement as a tuple

$$[\![x_1 : A_1, \ldots, x_n : A_n \vdash M : A]\!]_{\rho,\phi} = (x_i)_{i\in\phi(x_1)\otimes\cdots\otimes\phi(x_n)},$$

where $x_i \in [\![A]\!]$ for all i, and where the domain of ρ and ϕ is the set $\{x_1, \ldots, x_n\}$.

The proof of Lemma 8.2 is done by structural induction. We first show that for all typing judgements $x_1 : A_1, \ldots, x_n : A_n \vdash M : B$, for all extended environments

(ϕ, ρ) such that $\|\rho(x_i)\|_{A_i}^{\phi(x_i)} \leqslant 1$, the inequality

$$\|[\![x_1 : A_1, \ldots, x_n : A_n \vdash M : B]\!]_{\rho,\phi}\|_B^{\phi(x_1)\otimes\cdots\otimes\phi(x_n)} \leqslant 1$$

is satisfied. We then conclude by using the following result.

Lemma 8.3 *Given an environment ρ, let $\bar{\phi}$ be the constant map of value $\{\star\}$ and let $\bar{\rho}$ be the map assigning $(\rho(x^A))$ to x^A. Then $[\![\Delta \vdash M : A]\!]_{\bar{\rho},\bar{\phi}} = ([\![\Delta \vdash M : A]\!]_\rho)$.*□

8.4 Completeness

The completeness result is obtained by choosing a carefully crafted category $\mathcal{C}$. For each type A, denote by $\mathcal{B}_A$ the canonical basis of $[\![A]\!]$. Define a category $\mathcal{C}$ as follows: objects are products $\mathcal{B}_{A_1} \times \cdots \times \mathcal{B}_{A_n}$, and arrows are the identities on objects.

Convention 8.4 When considering the set $\mathcal{B}_A$, we will use the implicit order $(1, \ldots, 0), \ldots, (0, \ldots, 1)$. We extend the order on $\mathcal{B}_{A_1} \times \cdots \times \mathcal{B}_{A_n}$ using the lexicographic convention.

Definition 8.5 Let $\mathcal{U}_{A_1,\ldots,A_n}$ be the set of tuples

$$\left\{ ([\![M]\!](a_1)\cdots(a_n))_{(a_1,\ldots,a_n)\in\mathcal{B}_{A_1}\times\cdots\times\mathcal{B}_{A_n}} \;\middle|\; \vdash M : A_1 \multimap \cdots \multimap A_n \multimap bit \right\}$$

Lemma 8.6 *The set $\mathcal{U}_{A_1,\ldots,A_n}$ is convex, and its interior contains the origin.*

Proof. The convexity is shown using the denotation of the *if* and of the Ω term. The fact that the origin lies in the interior comes from the correspondence between first-order and higher-order types. □

Lemma 8.7 *Consider the tuple $(x_i)_{i\in\mathcal{B}_{C_1}\times\cdots\times\mathcal{B}_{C_k}}$, where for all tuples i, $x_i \in [\![A_1 \multimap \cdots \multimap A_n \multimap B]\!]$. Then for all $\boldsymbol{a} \in \mathcal{B}_{A_1} \times \cdots \times \mathcal{B}_{A_n}$, the value $x_i(a_1)\cdots(a_n)$ is equal to $(x_i)_{a_1\otimes\cdots\otimes a_n}$, the coordinate of x_i at $a_1 \otimes \cdots \otimes a_n$. Moreover, its norm*

$$\|(x_i(a_1)\cdots(a_n))_{(a_1,\ldots,a_n)}\|_B^{\mathcal{B}_{C_1}\times\cdots\times\mathcal{B}_{C_k}\times\mathcal{B}_{A_1}\times\cdots\times\mathcal{B}_{A_n}} = \|(x_i)_i\|_{A_1\multimap\cdots\multimap A_n\multimap B}^{\mathcal{B}_{C_1}\times\cdots\times\mathcal{B}_{C_k}}$$

Definition 8.8 Define the norm with varying arity $(\|-\|_A^w)_{w\in|\mathcal{C}|,A\in Type}$ at ground types as follows: the unit ball of $\|-\|_{bit}^{\mathcal{B}_{A_1}\times\cdots\times\mathcal{B}_{A_n}}$ is $\mathcal{U}_{A_1,\ldots,A_n}$.

Lemma 8.9 (Completeness) *Given $x \in [\![A]\!]$ such that $\|(x)\|_A \leqslant 1$, there exists a closed term M such that $[\![M]\!] = x$.*

Theorem 8.10 *The element $x \in [\![A]\!]$ is representable by a probabilistic lambda term if and only if for all norms with varying arity $(\|-\|_A^w)_{w\in|\mathcal{C}|,A\in Type}$, the norm $\|(x)\| \leqslant 1$.*

9 Conclusion

We considered the probabilistic fragment of the linear quantum lambda-calculus described in [6]. We interpreted the Bell inequalities in this higher-order context.

We sketched a method for computing generalized Bell inequalities at higher types, using convex polytopes.

The polytope interpretation does not provide a compositional semantics. Finally, we drafted a compositional full and complete semantics for a fragment of the probabilistic linear lambda-calculus.

Acknowledgement

I would like to thanks Peter Selinger for suggesting the problem of characterizing the definable higher-order probabilistic functions, and for providing the counterexample in Section 7.

References

[1] Bell, J. S., *On the Einstein Podolsky Rosen paradox*, Physics **1** (1964), pp. 195–200.

[2] Danos, V. and R. S. Harmer, *Probabilistic game semantics*, ACM Transactional on Computational Logic **3** (2002), pp. 359–382.

[3] Jung, A. and J. Tiuryn, *A new characterization of lambda definability*, in: *Proceedings of TLCA'93*, Lecture Notes in Computer Science **664**, 1993, pp. 245–257.

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

[5] Selinger, P., *Towards a semantics for higher-order quantum computation*, in: P. Selinger, editor, *Proceedings of QPL'04*, TUCS General Publication No 33 (2004), pp. 127–143.

[6] Selinger, P. and B. Valiron, *On a fully abstract model for a quantum linear functional language (extended abstract)*, in: P. Selinger, editor, *Proceedings of QPL'06*, Electronic Notes in Theoretical Computer Science **210**, 2008, pp. 123–137.