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

Benoît Valiron

Department of Mathematics and Statistics
University of Ottawa
Ottawa, Canada

Abstract

In this paper we discuss a semantics for a linear higher-order probabilistic lambda-calculus in the light of the semantics of completely positive maps for quantum computation. We analyse the set of representable elements in this category and describe some of its properties. We then show how one can use this to derive information on the capabilities of higher-order quantum computation compared to probabilistic computation. Finally, we derive a sound and complete semantics for a subset of the probabilistic language.

Keywords: semantics, higher-order, programming language, quantum computation, probabilistic computation.

1 Introduction

A fully-abstract semantics for a linear quantum lambda-calculus using the category **CPM** of completely positive maps is given in [6]. This denotation is not complete, and thus fails to distinguish between maps in the category that are images of a program and maps that are not.

A complete denotation for the first-order case is provided in [4]. Using the subcategory of superoperators, the denotation uses the notion of norm to determine the image of the language. However, the notion of norm fails to provide a suitable characterization at higher-order [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 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 [3].

[1] Email: bvali087@uottawa.ca

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

2 The Linear Quantum Lambda-Calculus

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

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

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

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

$$\begin{gathered} [\![\mathit{bit}]\!] = 1, 1 \qquad [\![\mathit{qbit}]\!] = 2 \\ [\![A \otimes B]\!] = [\![A \multimap B]\!] = [\![A]\!] \otimes [\![B]\!]. \end{gathered}$$

The booleans 0 and 1 are $[\![0]\!] = (1, 0) \in \mathbb{C}^2$ and $[\![1]\!] = (0, 1) \in \mathbb{C}^2$. The operations of creation and of measurements of a quantum bit are

$$\begin{aligned} [\![\mathit{new}]\!] : \quad [\![\mathit{bit}]\!] &\longrightarrow [\![\mathit{qbit}]\!] \\ (a, b) &\longmapsto \left(\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right), \\ [\![\mathit{meas}]\!] : \ [\![\mathit{qbit}]\!] &\longrightarrow [\![\mathit{bit}]\!] \\ \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) &\longmapsto (a, d). \end{aligned}$$

Example 2.1 It is possible to simulate a fair toss-coin using the language: $\mathit{meas}(H(\mathit{new}\, 0)) : \top \multimap \mathit{bit}$, where H corresponds to the Hadamard gate. The interpretation of this term is $(\frac{1}{2}, \frac{1}{2})$. Using a similar technique (and 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 get a probabilistic language. For this, we only need to remove the constants $\mathit{meas}, \mathit{new}$ and U. In order to retain the probabilistic effect of the measurement, we replace them with a set of constants $c(a)$, one for each pair of real numbers $0 \leqslant a \leqslant 1$. The meaning of the term $c(a)$ is "the boolean that is of value 1 with probability a and 0 with probability $(1 - a)$". The types are modified by removing the type qbit. We are left with the following terms and types:

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

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 n and m. The result of [6] extends to this case and the semantics is fully abstract.

4 Interpretation of the Bell's Inequalities

The Bell's 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 they can independently choose one of the following three bases $\{a, b, c\}$ for measuring their quantum bits:

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

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 **EPR** computing the EPR state:

$$\top \xrightarrow{(new\,1)\otimes(new\,0)} qbit \otimes qbit \xrightarrow{H\otimes id} qbit \otimes qbit \xrightarrow{N_c} qbit \otimes qbit \xrightarrow{id\otimes N} qbit \otimes qbit.$$

Note that we consider the tensor $\otimes$ to consist of all possible entangled pairs. In that sense, we can write the type of **EPR** as being $\top \to qbit \otimes qbit$.

Then Alice and Bob take each one qubit, choose a basis, and measure: the measurement they perform is then a function $\mathbf{f} : qbit \otimes trit \to bit$, where $trit = \top \oplus \top \oplus \top$. One can curry this function into $\mathbf{f'} : qbit \to (trit \multimap bit)$.

The algorithm is the composition

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

and produces a term of type $(trit \multimap bit) \otimes (trit \multimap bit)$. 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's inequalities precisely tell us that it is not the case.

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, for any classical type A (that is, not containing $qbit$).

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

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

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. What we mean by this is that the denotation of any valid typing judgement $\Delta \vdash M : A$ can be re-written as a probabilistic sum

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

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

The idea of the proof is the following: first,

$$[\![\Delta \vdash \mathit{if}\, c(a)\ \mathit{then}\ M\ \mathit{else}\ N : A]\!] = a[\![\Delta \vdash M : A]\!] + (1-a)[\![\Delta \vdash N : A]\!].$$

Thus for the result to be true, one needs to be able to send a term M containing a constant term $c(a)$ to a term of the form $\mathit{if}\, c(a) \mathit{then} M_1 \mathit{else} M_2$ with same denotation, where M_1 and M_2 satisfy some invariant. Since the language is linear, it is possible to write a rewriting system as in Table 1, where terms keep their denotation along the rewriting.

6 Interpretation as Polytopes

The deterministic terms (i.e. the ones with no occurrence of $c(a)$) 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 alpha-equivalence) such that $\Delta \vdash N_i : A$.*

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 *In the set $[\![A]\!]$, the subset of vectors representing linear probabilistic programs of type A is a convex* 0-1*-polytope (i.e. whose vertexes are of the form $(x_1, \ldots, x_n) \in [\![A]\!]$, where, for all i, 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 spawn 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 spawn by $(0,0)$, $(0,1)$, $(1,0$ and $(1,1)$.

(iii) The deterministic closed terms of type $(bit \multimap \top) \multimap \top$ are Ω, $\lambda f.(let * = f0\ in *)$ and $\lambda f.(let * = f1\ in *)$. That is, the polytope of admissible vectors is spawn by $(0,0)$, $(1,0)$ and $(0,1)$.

(iv) The Bell's algorithm described in Section 4 has for interpretation a vector in $[\![(trit \multimap bit) \otimes (trit \multimap bit)]\!] = \mathbb{C}^{3\times 2\times 3\times 2} = \mathbb{C}^{64}$. The Bell's inequalities carve out the polytope of representable elements in this space and state that the vector does not belong to it.

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 program

$$x : (bit \multimap \top) \multimap \top \vdash M : bit.$$

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 spans $\{0,1\}$ and where a and b span $\{*, \Omega\}$. The corresponding vertexes 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 $(1,0,0,1)$. The best one can do is to produce $\frac{1}{2}(1,0,0,1)$ by using a probabilistic distribution of the vertexes.

Thus, although polytopes provide a validating tool, allowing us to check, for example, that the Bell's experiment is not probabilistically representable, they are not enough to give a compositional semantics for the probabilistic language.

8 Toward a Full and Complete Semantics for the Probabilistic Linear Calculus

The polytopes constructed are convex sets containing the origin. Thus each one defines a norm. We extend this idea of norm following the work of [3], and define what we call linear Kripke logical relations: instead of considering the norm of a single vector, we consider the norm of a tuple of vectors.

8.1 A Toy Language

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

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

where $0 \leqslant \mu \leqslant 1$. We are going to build a full and complete semantics for this language.

8.2 Linear Kripke Logical Relations

Consider a small category of sets $\mathcal{C}$ with the product as a monoidal structure, containing the object $\top = \{\star\}$. We 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 verify 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

$$\max \left(\|(g_{f(j)}(x_i))_{j\otimes i \in w'\otimes v}\|_B^{w'\otimes v} \;\middle|\; \begin{array}{l} f : w' \to w \in \mathcal{C}, \\ (x_i)_{i\in v} \in [\![A]\!] \text{when} \|(x_i)_{i\in v}\|_A^v \leqslant 1 \end{array} \right).$$

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

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

8.3 Soundness

The soundness result states that the denotation of a closed term has indeed a norm smaller or equal to 1. The proof follows the idea developed in [3].

Definition 8.2 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 $\Delta = \{x_1 : A_1, \dots, x_n : A_n\}$, we write $\phi(\Delta)$ in place of $\phi(x_1) \otimes \cdots \otimes \phi(x_n)$.

Definition 8.3 We define the *extended interpretation* of a typing judgement as a tuple $[\![x_1 : A_1, \dots, x_n : A_n \vdash M : A]\!]_{\rho,\phi} = (x_i)_{i \in \phi(x_1) \otimes \cdots \otimes \phi(x_n)}$ where $x_i \in [\![A]\!]$ for all indices i, and where the domain of ρ and ϕ is the set $\{x_1, \dots, x_n\}$. We define this tuple as follows. First,

$$\begin{aligned} [\![\Delta \vdash \Omega^B : B]\!]_{\rho,\phi} &= (0)_{i \in \phi(\Delta)}, \\ [\![\vdash c(\mu) : bit]\!]_{\rho,\phi} &= ((1-\mu, \mu)), \\ [\![x : A \vdash x^A : A]\!]_{\rho,\phi} &= \rho(x^A). \end{aligned}$$

Then, if $[\![\Delta, x : A \vdash M : B]\!]_{\rho[x \mapsto (a)], \phi[x \mapsto \top]} = (c_i(a))_{i \in \phi(\Delta) \otimes \top}$, we have

$$[\![\Delta \vdash \lambda x^A.M : A \multimap B]\!]_{\rho,\phi} = (c_i)_{i \in \phi(\Delta)}.$$

If $[\![\Delta_1 \vdash M : A \multimap B]\!]_{\rho_1,\phi_1} = (g_i)_{i \in \phi_1(\Delta_1)}$ and $[\![\Delta_2 \vdash N : A]\!]_{\rho_2,\phi_2} = (x_j)_{j \in \phi_2(\Delta_2)}$, then

$$[\![\Delta_1, \Delta_2 \vdash MN : B]\!]_{\rho_1 \cup \rho_2, \phi_1 \cup \phi_2} = (g_i(x_j))_{i \otimes j \in \phi_1(\Delta_1) \otimes \phi_2(\Delta_2)}.$$

Finally, if

$$\begin{aligned} [\![\Delta_1 \vdash P : bit]\!]_{\rho_1,\phi_1} &= (a_i, b_i)_{i \in \phi_1(\Delta_1)}, \\ [\![\Delta_2 \vdash M : A]\!]_{\rho_2,\phi_2} &= (g_j)_{j \in \phi_2(\Delta_2)}, \\ [\![\Delta_2 \vdash N : A]\!]_{\rho_2,\phi_2} &= (h_j)_{j \in \phi_2(\Delta_2)}, \end{aligned}$$

we have

$$[\![\Delta_1, \Delta_2 \vdash \mathit{if}\, P \,\mathit{then}\, M \,\mathit{else}\, N : A]\!]_{\rho_1 \cup \rho_2, \phi_1 \cup \phi_2} = (b_i g_j + a_i h_j)_{i \otimes j \in \phi_1(\Delta_1) \otimes \phi_2(\Delta_2)}.$$

Lemma 8.4 *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 the extended interpretation $[\![\Delta \vdash M : A]\!]_{\bar{\rho},\bar{\phi}} = ([\![\Delta \vdash M : A]\!]_\rho)$.* □

Lemma 8.5 *Suppose $v \in |\mathcal{C}|$ and $(a_i)_{i \in v}$ is a tuple of elements of $[\![A]\!]$. Suppose that for all $i \in v$, $[\![\Delta, x : A \vdash M : B]\!]_{\rho \cup \{x \mapsto (a_i)\}, \phi \cup \{x \mapsto \top\}} = (b^i_j)_{j \in \phi(\Delta)}$. Then*

$$[\![\Delta, x : A \vdash M : B]\!]_{\rho \cup \{x \mapsto (a_i)_{i \in v}\}, \phi \cup \{x \mapsto v\}} = (b^i_j)_{j \otimes i \in \phi(\Delta) \otimes v}.$$

Proof. By structural induction on the typing judgement. □

Corollary 8.6 *If* $[\![\Delta \vdash \lambda x^A.M : A \multimap B]\!]_{\rho,\phi} = (g_j)_{j\in\phi(\Delta)}$ *and if* $(a_i)_{i\in v}$ *for some* $v \in |\mathcal{C}|$ *is such that* $a_i \in [\![A]\!]$ *for all* i*, then*

$$(g_j(a_i))_{j\otimes i\in\phi(\Delta)\otimes v} = [\![\Delta, x : A \vdash M : B]\!]_{\rho[x^A\mapsto(a_i)_{i\in v}],\phi[x^A\mapsto v]}.$$

Proof. Using the definition and Lemma 8.5. □

Lemma 8.7 (Soundness) *For all closed term* $M : A$*,* $\|[\![M]\!]\|_A \leqslant 1$*.*

Proof (Sketch) We prove by structural induction that for typing judgements $x_1 : A_1, \dots, x_n : A_n \vdash M : B$, for each extended environment (ϕ, ρ) such that $\|\rho(x_i)\|_{A_i}^{\phi(x_i)} \leqslant 1$, the norm

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

The difficult case is the λ-abstraction, and it is taken care of using Corollary 8.6 and Lemma 8.1. The desired result is obtained using Lemma 8.4. □

8.4 Completeness

The completeness result is obtained by choosing a carefully crafted category $\mathcal{C}$.

For each type A, denote with $\mathcal{B}_A$ the canonical basis of $[\![A]\!]$. When considering the set $\mathcal{B}_A$, we use the implicit order $(1, \dots, 0), \dots, (0, \dots, 1)$. We extend the order on $\mathcal{B}_{A_1} \times \cdots \times \mathcal{B}_{A_n}$ using the lexicographic convention.

Definition 8.8 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.

Definition 8.9 Let $\mathcal{U}_{A_1,\dots,A_n}$ be the set of tuples

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

Lemma 8.10 *The set* $\mathcal{U}_{A_1,\dots,A_n}$ *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 [6]. □

Lemma 8.11 *Consider the tuple* $(x_i)_{i\in\mathcal{B}_{C_1}\times\cdots\times\mathcal{B}_{C_k}}$*, where for all tuple* i*,* $x_i \in [\![A_1 \multimap \cdots \multimap A_n \multimap B]\!]$*. Then*

$$\forall \boldsymbol{a} \in \mathcal{B}_{A_1} \times \cdots \times \mathcal{B}_{A_n}, \quad x_i(a_1)\cdots(a_n) = (x_i)_{a_1\otimes\cdots\otimes a_n},$$

the coordinate of x_i *at* $a_1 \otimes \cdots \otimes a_n$*. Moreover,*

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

□

Definition 8.12 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 8.13 (Completeness) *Given $x \in [\![A]\!]$ such that $\|(x)\|_A \leqslant 1$, there exists a closed term M such that $[\![M]\!] = x$.* □

Theorem 8.14 *$x \in [\![A]\!]$ is representable if and only if for all norm with varying arity $(\| - \|_A^w)_{w\in|\mathcal{C}|,A\in Type}$, the norm $\|(x)\| \leqslant 1$.* □

9 Conclusion

We restricted the linear quantum lambda-calculus described in [6] to probabilistic effects, and we interpreted the Bell's inequalities in the context of higher-order computation. We then generalized the problem and sketched a method for answering such generalizations using convex polytopes.

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

10 Acknowledgements

I would like to thank Peter Selinger for fruitful discussions. In particular, Sections 4 and 7 are his ideas.

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*, in: P. Selinger, editor, *Preliminary proceedings of QPL'06*, 2006, pp. 103–115.