Realizability in the Unitary Sphere

Alejandro Díaz-Caro, Mauricio Guillermo, Alexandre Miquel, Benoît Valiron

Realizability in the Unitary Sphere

Alejandro Díaz-Caro*†, Mauricio Guillermo‡, Alexandre Miquel‡, and Benoît Valiron§
*Universidad Nacional de Quilmes, Bernal, Buenos Aires, Argentina
†Instituto de Ciencias de la Computación (UBA-CONICET), Buenos Aires, Argentina
Email: adiazcaro@icc.fcen.uba.ar
‡Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
Email: {mguille,amiquel}@fing.edu.uy
§LRI, CentraleSupélec, Université Paris-Saclay, Orsay, France
Email: benoit.valiron@lri.fr

***Abstract*—In this paper we present a semantics for a linear algebraic lambda-calculus based on realizability. This semantics characterizes a notion of unitarity in the system, answering a long standing issue. We derive from the semantics a set of typing rules for a simply-typed linear algebraic lambda-calculus, and show how it extends both to classical and quantum lambda-calculi.**

I. Introduction

The linear-algebraic lambda calculus (Lineal) [1]–[3] is an extension of the lambda calculus where lambda terms are closed under linear combinations over a semiring K. For instance, if t and r are two lambda terms, then so is $\alpha.t+\beta.r$ with $\alpha, \beta \in K$. The original motivation of [1] for such a calculus was to set the basis for a future quantum calculus, where $\alpha.t + \beta.r$ could be seen as the generalization of the notion of quantum superposition to the realm of programs (in which case K is the field $\mathbb{C}$ of complex numbers).

In quantum computation, data is encoded in the state of a set of particles governed by the laws of quantum mechanics. The mathematical formalization postulates that quantum data is modeled as a unit vector in a Hilbert space. The quantum analogue to a Boolean value is the *quantum bit*, that is a linear combination of the form $\phi = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle$ and $|1\rangle$ respectively correspond to "true" and "false", and where $|\alpha|^2+|\beta|^2 = 1$. In other words, the state ϕ is a linear combination of the Boolean values "true" and "false", of l_2-norm equal to 1: it is a unit-vector in the Hilbert space $\mathbb{C}^2$.

A quantum memory consists in a list of registers holding quantum bits. The canonical model for interacting with a quantum memory is the QRAM model [4]. A fixed set of elementary operations are allowed on each quantum register. Mathematically, these operations are modeled with unitary maps on the corresponding Hilbert spaces, that is: linear maps preserving the l_2-norm and the orthogonality. These operations, akin to Boolean gates, are referred to as quantum gates, and they can be combined into linear sequences called quantum circuits. Quantum algorithms make use of a quantum memory to solve a particular classical problem. Such an algorithm therefore consists in particular in the description of a quantum circuit.

Several existing languages for describing quantum algorithms such as Quipper [5] and QWIRE [6] are purely functional and based on the lambda calculus. However, they only provide *classical control*: the quantum memory and the allowed operations are provided as black boxes. These languages are mainly circuit description languages using opaque high-level operations on circuits. They do not feature *quantum control*, in the sense that the operations on quantum data are not programmable.

A lambda calculus with linear combinations of terms made "quantum" would allow to program those "black boxes" explicitly, and provide an operational meaning to quantum control. However, when trying to identify quantum data with linear combinations of lambda terms, the problem arises from the norm condition on quantum superpositions. To be quantum-compatible, one cannot have *any* linear combination of programs. Indeed, programs should at the very least yield valid quantum superpositions, that is: linear combinations whose l_2-norm equals 1—a property which turns out to be very difficult to preserve along the reduction of programs.

So far, the several attempts at accommodating linear algebraic lambda calculi with the l_2-norm have failed. At one end of the spectrum, [7] stores lambda terms directly in the quantum memory, and encodes the reduction process as a purely quantum process. Van Tonder shows that this forces all lambda terms in superposition to be mostly equivalent. At the other end of the spectrum, the linear algebraic approaches pioneered by Arrighi and Dowek consider a constraint-free calculus and try to recover quantum-like behavior by adding ad-hoc term reductions [1] or type systems [8]–[10]. But if these approaches yield very expressive models of computations, none of them is managing to precisely characterize linear combinations of terms of unit l_2-norm, or equivalently, the unitarity of the representable maps.

This paper answers this question by presenting an algebraic

A. Díaz-Caro and B. Valiron have been partially supported by PICT 2015-1208, ECOS-Sud A17C03, and the French-Argentinian International Laboratory SINFIN. B. Valiron has been partially supported by the French National Research Agency (ANR) under the research project SoftQPRO ANR-17-CE25-0009-02, and by the DGE of the French Ministry of Industry under the research project PIA-GDN/QuantEx P163746-484124. M. Guillermo and A. Miquel have been partially supported by the Uruguayan National Research & Innovation Agency (ANII) under the research project "Realizability, Forcing and Quantum Computing", FCE_1_2014_1_104800.

lambda calculus together with a type system that enforces unitarity. For that, we use semantic techniques coming from *realizability* [11] to decide on the unitarity of terms.

Since its creation by Kleene as a semantics for Heyting arithmetic, realizability has evolved to become a versatile toolbox, that can be used both in logic and in functional programming. Roughly speaking, realizability can be seen as a generalization of the notion of typing where the relation between a term and its type is not defined from a given set of inference rules, but from the very operational semantics of the calculus, via a computational interpretation of types seen as specifications. Types are first defined as sets of terms verifying certain properties, and then, valid typing rules are derived from these properties rather than set up as axioms.

The main feature of our realizability model is that types are not interpreted as arbitrary sets of terms or values, but as subsets of the *unit sphere* of a particular *weak* vector space [3], whose vectors are *distributions* (i.e. weak linear combinations) of "pure" values. So that by construction, all functions that are correct w.r.t. this semantics preserve the ℓ_2-norm. As we shall see, this interpretation of types is not only compatible with the constructions of the simply typed lambda calculus (with sums and pairs), but it also allows us to distinguish pure data types (such as the type $\mathbb{B}$ of pure Booleans) from quantum data types (such as the type $\sharp\mathbb{B}$ of quantum Booleans). Thanks to these constraints, the type system we obtain naturally enforces that the realizers of the type $\sharp\mathbb{B} \to \sharp\mathbb{B}$ are precisely the functions representing unitary operators of $\mathbb{C}^2$.

This realizability model is therefore answering a hard problem [12]: it provides a unifying framework able to express not only *classical control*, with the presence of "pure" values, but also *quantum control*, with the possibility to interpret quantum data-types as (weak) linear combinations of classical ones.

A. Contributions

(1) We propose a realizability semantics based on a linear algebraic lambda calculus capturing a notion of unitarity through the use of a l_2-norm. As far as we know, such a construction is novel.

(2) The semantics provides a *unified* model for both classical and quantum control. Strictly containing the simply-typed lambda calculus, it does not only serve as a model for a quantum circuit-description language, but it also provides a natural interpretation of quantum control.

(3) In order to exemplify the expressiveness of the model, we show how a circuit-description language in the style of QWIRE [6] can be naturally interpreted in the model. Furthermore, we discuss how one can give within the model an *operational semantics* to a high-level operation on circuits usually provided as a black box in circuit-description languages: the control of a circuit.

B. Related Works

Despite its original motivations, [10] showed that Lineal can handle the l_1-norm. This can be used for example to represent probabilistic distributions of terms. Also, a simplification of Lineal, without scalars, can serve as a model for non-deterministic computations [13]. And, in general, if we consider the standard values of the lambda calculus as the basis, then linear combinations of those form a vector space, which can be characterized using types [9]. In [14] a similar distinction between classical bits ($\mathbb{B}$) and qbits ($\sharp\mathbb{B}$) has been also studied. However, without unitarity, it is impossible to obtain a calculus that could be compiled onto a quantum machine. Finally, a concrete categorical semantics for such a calculus has been recently given in [15].

An alternative approach for capturing unitarity (of data superpositions and functions) consists to change the language. Instead of starting with a lambda calculus, [16] defines and extends a reversible language to express quantum computation.

Lambda calculi with vectorial structures are not specific to quantum computation. Vaux [17] independently developed the algebraic lambda calculus (where linear combinations of terms are also terms), initially to study a fragment of the differential lambda calculus of [18]. Unlike its quantum-inspired cousin Lineal, the algebraic lambda calculus is morally call-by-name, and [19] shows the formal connection with Lineal.

Designing an (unconstrainted) algebraic lambda calculus (in call-by-name [17] or in call-by-value [1]) raises the problem of how to enforce the confluence of reduction. Indeed, if the semi-ring K is a ring, since $0 \cdot t = \vec{0}$, it is possible to design a term Y_t reducing both to t and the empty linear combination $\vec{0}$. A simple solution to recover consistency is to weaken the vectorial structure and remove the equality $0 \cdot t = \vec{0}$ [3]. The vector space of terms becomes a *weak* vector space. This approach is the one we shall follow in our construction.

This paper is concerned with modeling quantum higher-order programming languages. If the use of realizability techniques is novel, several other techniques have been used, based on positive matrices and categorical tools. For first-order quantum languages, [20] constructs a fully complete semantics based on superoperators. To model a strictly linear quantum lambda-calculus, [21] shows that the compact closed category CPM based on completely positive maps forms a fully abstract model. Another approach has been taken in [22], with the use of a presheaf model on top of the category of superoperators. To accomodate duplicable data, [23] extends CPM using techniques developed for quantitative models of linear logic. Finally, a categorical semantics of circuit-description languages has been recently designed using linear-non-linear models by [24], [25].

C. Outline

Section II presents the linear algebraic calculus and its weak vector space structure. Section III discusses the evaluation of term distributions. Section IV introduces the realizability semantics and the algebra of types spawning from it. At the end of this section, Theorem IV.12 and Corollary IV.13 express that the type of maps from quantum bits to quantum bits only contains unitary functions. Section V introduces a notion of typing judgment and derives a set of valid typing rules from the

semantics. Section V-B discusses the inclusion of the simply-typed lambda calculus in this unitary semantics. Finally, Section VI describes a small quantum circuit-description language and shows how it lives inside the unitary semantics.

II. Syntax of the calculus

This section presents the calculus upon which our realizability model will be designed. It is a lambda-calculus extended with linear combinations of lambda-terms, but with a subtelty: terms form a *weak vector space*.

A. Values, terms and distributions

The language is made up of four syntactic categories: *pure values*, *pure terms*, *value distributions* and *term distributions* (Table I). As usual, the expressions of the language are built from a fixed denumerable set of *variables*, written $\mathcal{X}$.

In this language, a *pure value* is either a variable x, a λ-abstraction $\lambda x\,.\,\vec{s}$ (whose body is an arbitrary term distribution $\vec{s}$), the void object $*$, a pair of pure values (v_1, v_2), or one the two variants $\mathtt{inl}(v)$ and $\mathtt{inr}(v)$ (where v is pure value). A *pure term* is either a pure value v or a destructor, that is: an application $s\,t$, a sequence $t;\vec{s}$ for destructing the void object in t[1], a let-construct $\mathtt{let}\ (x_1,x_2) = t\ \mathtt{in}\ \vec{s}$ for destructing a pair in t, or a match-construct $\mathtt{match}\ t\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\}$ (where $\vec{s}$, $\vec{s}_1$ and $\vec{s}_2$ are arbitrary term distributions). A *term distribution* is simply a formal $\mathbb{C}$-linear combination of pure terms, whereas a *value distribution* is a term distribution that is formed only from pure values. We also define Booleans using the following abbreviations: $\mathtt{tt} := \mathtt{inl}(*)$, $\mathtt{ff} := \mathtt{inr}(*)$, and, finally, $\mathtt{if}\ t\ \{\vec{s}_1 \mid \vec{s}_2\} := \mathtt{match}\ t\ \{\mathtt{inl}(x_1) \mapsto x_1; \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto x_2; \vec{s}_2\}$.

The notions of free and bound (occurrences of) variables are defined as expected, and in what follows, we shall consider pure values, pure terms, value distributions and term distributions up to α-conversion, silently renaming bound variables whenever needed. The set of all pure terms (resp. of all pure values) is written $\Lambda(\mathcal{X})$ (resp. $\mathrm{V}(\mathcal{X})$), whereas the set of all term distributions (resp. of all value distributions) is written $\vec{\Lambda}(\mathcal{X})$ (resp. $\vec{\mathrm{V}}(\mathcal{X})$). So that we have the inclusions:

$$\begin{array}{ccc} \Lambda(\mathcal{X}) & \subset & \vec{\Lambda}(\mathcal{X}) \\ \cup & & \cup \\ \mathrm{V}(\mathcal{X}) & \subset & \vec{\mathrm{V}}(\mathcal{X}) \end{array}$$

B. Distributions as weak linear combinations

Formally, the set $\vec{\Lambda}(\mathcal{X})$ of term distributions is equipped with a congruence $\equiv$ that is generated from the 7 rules of Table II. We assume that the congruence $\equiv$ is shallow, in the sense that it only goes through sums ($+$) and scalar multiplications ($\cdot$), and stops at the level of pure terms. So that $\vec{t}+(\vec{s}_1+\vec{s}_2) \equiv \vec{t}+(\vec{s}_2+\vec{s}_1)$ but $\lambda x\,.\,\vec{s}_1+\vec{s}_2 \not\equiv \lambda x\,.\,\vec{s}_2+\vec{s}_1$. (This important design choice will be justified in Section V-A, Remark V.5). We easily check that:

[1]Note the asymmetry: t is a pure term whereas $\vec{s}$ is a term distribution. As a matter of fact, the sequence $t;\vec{s}$ (that could also be written $\mathtt{let}\ * = t\ \mathtt{in}\ \vec{s}$) is the nullary version of the pair destructing let $\mathtt{let}\ (x_1,x_2) = t\ \mathtt{in}\ \vec{s}$.

Lemma II.1. *For all $\alpha \in \mathbb{C}$, we have $\alpha \cdot \vec{0} \equiv \vec{0}$.*

Proof. From $0\cdot\vec{0} \equiv 0\cdot\vec{0}+\vec{0} \equiv 0\cdot\vec{0}+1\cdot\vec{0} \equiv (0+1)\cdot\vec{0} = 1\cdot\vec{0} \equiv \vec{0}$, we get $\alpha \cdot \vec{0} \equiv \alpha \cdot (0 \cdot \vec{0}) \equiv (0\alpha) \cdot \vec{0} = 0 \cdot \vec{0} \equiv \vec{0}$. □

On the other hand, the relation $0 \cdot \vec{t} \equiv \vec{0}$ cannot be derived from the rules of Table II as we shall see below (Proposition II.6 and Example II.7). As a matter of fact, the congruence $\equiv$ implements the equational theory of a restricted form of linear combinations—which we shall call *distributions*—that is intimately related to the notion of *weak vector space* [3].

Definition II.2 (Weak vector space)**.** *A weak vector space (over a given field K) is a commutative monoid $(V, +, \vec{0})$ equipped with a scalar multiplication $(\cdot) : K \times V \to V$ such that for all $u, v \in V$, $\alpha, \beta \in K$, we have $1 \cdot u = u$, $\alpha \cdot (\beta \cdot u) = \alpha\beta \cdot u$, $(\alpha + \beta) \cdot u = \alpha \cdot u + \beta \cdot u$, and $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$.*

Remark II.3. *The notion of weak vector space differs from the traditional notion of vector space in that the underlying additive structure $(V, +, \vec{0})$ may be an arbitrary commutative monoid, whose elements do not necessarily have an an additive inverse. So that in a weak vector space, the vector $(-1)\cdot u$ is in general* not *the additive inverse of u, and the product $0\cdot u$ does not simplify to $\vec{0}$.*

Weak vector spaces naturally arise in functional analysis as the spaces of *unbounded operators*. Historically, the notion of unbounded operator was introduced by von Neumann to give a rigorous mathematical definition to the operators that are used in quantum mechanics. Given two (usual) vector spaces $\mathscr{E}$ and $\mathscr{F}$ (over the same field K), recall that an *unbounded operator* from $\mathscr{E}$ to $\mathscr{F}$ is a linear map $f : D(f) \to \mathscr{F}$ that is defined on a sub-vector space $D(f) \subseteq \mathscr{E}$, called the *domain* of f. The sum of two unbounded operators $f, g : \mathscr{E} \rightharpoonup \mathscr{F}$ is defined by: $D(f + g) := D(f) \cap D(g)$, $(f + g)(x) := f(x) + g(x)$ (for all $x \in D(f + g)$), whereas the product of an unbounded operator $f : \mathscr{E} \rightharpoonup \mathscr{F}$ by a scalar $\alpha \in K$ is defined by: $D(\alpha \cdot f) := D(f)$, $(\alpha \cdot f)(x) := \alpha \cdot f(x)$ (for all $x \in D(\alpha \cdot f)$).

Example II.4. *The space $Ł(\mathscr{E}, \mathscr{F})$ of all unbounded operators from $\mathscr{E}$ to $\mathscr{F}$ is a weak vector space, whose null vector is the (totally defined) null function.*

Indeed, we observe that an unbounded operator $f \in Ł(\mathscr{E}, \mathscr{F})$ has an additive inverse if and only f is total, that is: if and only if $D(f) = \mathscr{E}$—and in this case, the additive inverse of f is the operator $(-1) \cdot f$. In particular, it should be clear to the reader that $0 \cdot f\ (= \vec{0}_{\restriction D(f)}) \neq \vec{0}$ as soon as $D(f) \neq \mathscr{E}$.

We can now observe that, by construction:

Proposition II.5. *The space $\vec{\Lambda}(\mathcal{X})/\equiv$ of all term distributions (modulo the congruence $\equiv$) is the free weak $\mathbb{C}$-vector space generated by the set $\Lambda(\mathcal{X})$ of all pure terms*[2]*.* □

[2]The same way as the space of linear combinations over a given set X is the free vector space generated by X.

Pure values	v, w	$::=$	$x \mid \lambda x\,.\,\vec{s} \mid * \mid (v_1, v_2) \mid \mathtt{inl}(v) \mid \mathtt{inr}(v)$
Pure terms	s, t	$::=$	$v \mid s\,t \mid t; \vec{s} \mid \mathtt{let}\ (x_1, x_2) = t\ \mathtt{in}\ \vec{s} \mid \mathtt{match}\ t\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\}$
Value distributions	$\vec{v}, \vec{w}$	$::=$	$\vec{0} \mid v \mid \vec{v} + \vec{w} \mid \alpha \cdot \vec{v}$ $\quad (\alpha \in \mathbb{C})$
Term distributions	$\vec{s}, \vec{t}$	$::=$	$\vec{0} \mid t \mid \vec{s} + \vec{t} \mid \alpha \cdot \vec{t}$ $\quad (\alpha \in \mathbb{C})$

TABLE I
SYNTAX OF THE CALCULUS

$$\begin{array}{cccc} \vec{t}+\vec{0} \equiv \vec{t} & 1\cdot\vec{t} \equiv \vec{t} & \alpha\cdot(\beta\cdot\vec{t}) \equiv \alpha\beta\cdot\vec{t} \\ \vec{t}_1+\vec{t}_2 \equiv \vec{t}_2+\vec{t}_1 & & (\vec{t}_1+\vec{t}_2)+\vec{t}_3 \equiv \vec{t}_1+(\vec{t}_2+\vec{t}_3) \\ (\alpha+\beta)\cdot\vec{t} \equiv \alpha\cdot\vec{t}+\beta\cdot\vec{t} & & \alpha\cdot(\vec{t}_1+\vec{t}_2) \equiv \alpha\cdot\vec{t}_1+\alpha\cdot\vec{t}_2 \end{array}$$

TABLE II
CONGRUENCE RULES ON TERM DISTRIBUTIONS

Again, the notion of distribution (or weak linear combination) differs from the standard notion of linear combination in that the summands of the form $0 \cdot t$ cannot be erased, so that the distribution $t_1 + (-3) \cdot t_2$ is not equivalent to the distribution $t_1 + (-3) \cdot t_2 + 0 \cdot t_3$ (provided $t_3 \not\equiv t_1, t_2$). In particular, the distribution $(-1) \cdot t_1 + 3 \cdot t_2$ is not the additive inverse of $t_1 + (-3) \cdot t_2$, since $\big(t_1 + (-3) \cdot t_2\big) + \big((-1) \cdot t_1 + 3 \cdot t_2\big) \equiv 0 \cdot t_1 + 0 \cdot t_2 \not\equiv \vec{0}$. However, the equivalence of term distributions can be simply characterized as follows:

Proposition II.6 (Canonical form of a distribution)**.** *Each term distribution $\vec{t}$ can be written $\vec{t} \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i$, where $\alpha_1, \ldots, \alpha_n \in \mathbb{C}$ are arbitrary scalars (possibly equal to 0), and where $t_1, \ldots, t_n$ ($n \geq 0$) are pairwise distinct pure terms. This writing—which is called the* canonical form of $\vec{t}$*—is unique, up to a permutation of the summands $\alpha_i \cdot t_i$ ($i = 1..n$).* □

Example II.7. *Given distinct pure terms t_1 and t_2, we consider the term distributions $\vec{t} := 3 \cdot t_1$ and $\vec{t}' := 3 \cdot t_1 + 0 \cdot t_2$. We observe that the distributions $\vec{t}$ and $\vec{t}'$ (that are given in canonical form) do not have the same number of summands, hence they are not equivalent: $\vec{t} \not\equiv \vec{t}'$.*

Corollary II.8. *The congruence $\equiv$ is trivial on pure terms: $t \equiv t'$ iff $t = t'$, for all $t, t' \in \Lambda(\mathcal{X})$.* □

Thanks to Proposition II.6, we can associate to each term distribution $\vec{t} \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i$ (written in canonical form) its *domain* $\mathrm{dom}(\vec{t}) := \{t_1, \ldots, t_n\}$[3] and its *weight* $\varpi(\vec{t}) := \sum_{i=1}^{n} \alpha_i$. Note that the weight function $\varpi : \vec{\Lambda}(\mathcal{X})/{\equiv} \to \mathbb{C}$ is a linear function from the weak $\mathbb{C}$-vector space of term distributions to $\mathbb{C}$, whereas the domain function $\mathrm{dom} : \vec{\Lambda}(\mathcal{X})/{\equiv} \to \mathfrak{P}_{\mathrm{fin}}(\Lambda(\mathcal{X}))$ is a morphism of commutative monoids from $(\vec{\Lambda}(\mathcal{X})/{\equiv}, +, \vec{0})$ to $(\mathfrak{P}_{\mathrm{fin}}(\Lambda(\mathcal{X})), \cup, \varnothing)$, since we have[4]: $\mathrm{dom}(\vec{0}) = \varnothing$, $\mathrm{dom}(\vec{t}_1 + \vec{t}_2) = \mathrm{dom}(\vec{t}_1) \cup \mathrm{dom}(\vec{t}_2)$, $\mathrm{dom}(t) = \{t\}$ and $\mathrm{dom}(\alpha \cdot \vec{t}) = \mathrm{dom}(\vec{t})$ for all $t \in \Lambda(\mathcal{X})$, $\vec{t}_1, \vec{t}_2 \in \vec{\Lambda}(\mathcal{X})$ and $\alpha \in \mathbb{C}$.

Remark II.9. *In practice, one of the main difficulties of working with distributions is that addition is not regular, in the sense that the relation $\vec{t} + \vec{t}_1 \equiv \vec{t} + \vec{t}_2$ does not necessarily imply that $\vec{t}_1 \equiv \vec{t}_2$. However, for example if $\vec{t} = \alpha.s$, we can deduce that $\vec{t}_1 \equiv \vec{t}_2$ or $\vec{t}_1 \equiv \vec{t}_2 + 0 \cdot s$ or $\vec{t}_2 \equiv \vec{t}_1 + 0 \cdot s$.*

To simplify the notation, we shall adopt the following:

Convention II.10. *From now on, we consider term distributions modulo the congruence $\equiv$, and simply write $\vec{t} = \vec{t}'$ for $\vec{t} \equiv \vec{t}'$. This convention does not affect* inner*—or* raw*—distributions (which occur within a pure term, for instance in the body of an abstraction), that are still considered only up to α-conversion*[5]*. The same convention holds for value distributions.*

To sum up, we now consider that $\vec{s}_1 + \vec{s}_2 = \vec{s}_2 + \vec{s}_1$ (as a top-level distribution), but:

$$\begin{array}{r} \lambda x\,.\,\vec{s}_1 + \vec{s}_2 \neq \lambda x\,.\,\vec{s}_2 + \vec{s}_1 \\ t; (\vec{s}_1 + \vec{s}_2) \neq t; (\vec{s}_2 + \vec{s}_1) \\ \mathtt{let}\ (x, y) = t\ \mathtt{in}\ \vec{s}_1 + \vec{s}_2 \neq \mathtt{let}\ (x, y) = t\ \mathtt{in}\ \vec{s}_2 + \vec{s}_1 \\ \mathtt{match}\ t\ \{\mathtt{inl}(x) \mapsto \vec{s}_1 + \vec{s}_2 \mid \mathtt{inr}(y) \mapsto \vec{s}\} \\ \neq \mathtt{match}\ t\ \{\mathtt{inl}(x) \mapsto \vec{s}_2 + \vec{s}_1 \mid \mathtt{inr}(y) \mapsto \vec{s}\} \\ \mathtt{match}\ t\ \{\mathtt{inl}(x) \mapsto \vec{s} \mid \mathtt{inr}(y) \mapsto \vec{s}_1 + \vec{s}_2\} \\ \neq \mathtt{match}\ t\ \{\mathtt{inl}(x) \mapsto \vec{s} \mid \mathtt{inr}(y) \mapsto \vec{s}_2 + \vec{s}_1\} \end{array}$$

C. Extending syntactic constructs by linearity

Pure terms and term distributions are intended to be evaluated according to the *call-by-basis* strategy (Section III), that can be seen as the declination of the *call-by-value* strategy in a computing environment where all functions are *linear by construction*. Keeping this design choice in mind, it is natural to extend the syntactic constructs of the language by linearity, proceeding as follows: for all value distributions $\vec{v} = \sum_{i=1}^{n} \alpha_i \cdot v_i$ and $\vec{w} = \sum_{j=1}^{m} \beta_j \cdot w_j$, and for all term

[3]Note that the domain of a distribution $\vec{t} \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i$ gathers all pure terms t_i ($i = 1..n$), including those affected with a coefficient $\alpha_i = 0$. So that the domain of a distribution should not be mistaken with its support.

[4]Actually, the function $\mathrm{dom} : \vec{\Lambda}(\mathcal{X})/{\equiv} \to \mathfrak{P}_{\mathrm{fin}}(\Lambda(\mathcal{X}))$ is even *linear*, since the commutative (and idempotent) monoid $(\mathfrak{P}_{\mathrm{fin}}(\Lambda(\mathcal{X})), \cup, \varnothing)$ has a natural structure of weak $\mathbb{C}$-vector space whose (trivial) scalar multiplication is defined by $\alpha \cdot X = X$ for all $\alpha \in \mathbb{C}$ and $X \in \mathfrak{P}_{\mathrm{fin}}(\Lambda(\mathcal{X}))$.

[5]Intuitively, a distribution that appears in the body of an abstraction (or in the body of a let-construct, or in a branch of a match-construct) does not represent a real superposition, but it only represents *machine code* that will produce later a particular superposition, after some substitution has been performed.

distributions $\vec{s}_1, \vec{s}_2$, $\vec{t} = \sum_{k=1}^{p} \gamma_k \cdot t_k$ and $\vec{s} = \sum_{\ell=1}^{q} \delta_\ell \cdot s_\ell$ we have:

$$
\begin{aligned}
(\vec{v}, \vec{w}) &:= \textstyle\sum_{i=1}^{n} \sum_{j=1}^{k} \alpha_i \beta_j \cdot (v_i, w_j) \\
\mathtt{inl}(\vec{v}) &:= \textstyle\sum_{i=1}^{n} \alpha_i \cdot \mathtt{inl}(v_i) \\
\mathtt{inr}(\vec{v}) &:= \textstyle\sum_{i=1}^{n} \alpha_i \cdot \mathtt{inr}(v_i) \\
\vec{t}\,\vec{s} &:= \textstyle\sum_{k=1}^{p} \sum_{\ell=1}^{q} \gamma_k \delta_\ell \cdot t_k s_\ell \\
\vec{t}; \vec{s} &:= \textstyle\sum_{k=1}^{p} \gamma_k \cdot (t_k; \vec{s}) \\
\mathtt{let}\ (x,y) = \vec{t}\ \mathtt{in}\ \vec{s} &:= \textstyle\sum_{k=1}^{p} \gamma_k \cdot \big(\mathtt{let}\ (x,y) = t_k\ \mathtt{in}\ \vec{s}\big) \\
\mathtt{match}\ \vec{t}\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\} &:= \\
\textstyle\sum_{k=1}^{p} \gamma_k \cdot \big(\mathtt{match}\ t_k\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\}\big)
\end{aligned}
$$

The value distribution $(\vec{v}, \vec{w})$ will be sometimes written $\vec{v} \otimes \vec{w}$ as well.

D. Substitutions

Given a variable x and a pure value w, we define an operation of *pure substitution*, written $[x := w]$, that associates to each pure value v (resp. to each pure term t, to each raw value distribution $\vec{v}$, to each raw term distribution $\vec{t}$) a pure value $v[x := w]$ (resp. a pure term $t[x := w]$, a raw value distribution $\vec{v}[x := w]$, a raw term distribution $\vec{t}[x := w]$). The four operations $v[x := w]$, $t[x := w]$, $\vec{v}[x := w]$ and $\vec{t}[x := w]$ are defined by mutual recursion as expected.

Although the operation $\vec{t}[x := w]$ is primarily defined on *raw* term distributions (i.e. by recursion on the tree structure of $\vec{t}$, without taking into account the congruence $\equiv$), it is compatible with the congruence $\equiv$, in the sense that if $\vec{t} \equiv \vec{t}'$, then $\vec{t}[x := w] \equiv \vec{t}'[x := w]$ for all pure values w. In other words, the operation of pure substitution is compatible with Convention II.10. It is also clear that, by construction, the operation $\vec{t}[x := w]$ is linear w.r.t. $\vec{t}$, so that $\vec{t}[x := w]$ is $\sum_{i=1}^{n} \alpha_i \cdot t_i[x := w]$ for all term distributions $\vec{t} = \sum_{i=1}^{n} \alpha_i \cdot t_i$. (The same observations hold for the operation $\vec{v}[x := w]$).

Moreover, the operation of pure substitution behaves well with the linear extension of the syntactic constructs of the language (cf. Appendix D). And we have the expected substitution lemma: For all term distributions $\vec{t}$ and for all pure values v and w, provided $x \neq y$ and $x \notin FV(w)$), we have $\vec{t}[x := v][y := w] := \vec{t}[y := w][x := v[y := w]]$. We extend the notation to parallel substitution in the usual manner (cf. Remark A.14 in Appendix D).

From the operation of pure substitution $[x := w]$, we define an operation of *bilinear substitution* $\langle x := \vec{w} \rangle$ that is defined for all term distributions $\vec{t} = \sum_{i=1}^{n} \alpha_i \cdot t_i$ and for all value distributions $\vec{w} = \sum_{j=1}^{m} \beta_j \cdot w_j$, letting $\vec{t}\langle x := \vec{w} \rangle := \sum_{j=1}^{m} \beta_j \cdot \vec{t}[x := w_j] = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j \cdot t_i[x := w_j]$. By construction, the generalized operation of substitution $\vec{t}\langle x := \vec{w} \rangle$ is bilinear—which is consistent with the bilinearity of application (Section II-C). But beware! The bilinearity of the operation $\vec{t}\langle x := \vec{w} \rangle$ also makes its use often counter-intuitive, so that this notation should always be used with the greatest caution. Indeed, while $\mathtt{inl}(\vec{v})\langle x := \vec{w} \rangle = \mathtt{inl}(\vec{v}\langle x := \vec{w} \rangle)$, $(v_1, v_2)\langle x := \vec{w} \rangle \neq (v_1\langle x := \vec{w} \rangle, v_2\langle x := \vec{w} \rangle)$. Lemma A.10, in Appendix C gives the valid identities. In addition, bilinear substitution is not (completely) canceled when $x \notin FV(\vec{t})$, in which case $\vec{t}\langle x := \vec{w} \rangle = \varpi(\vec{w}) \cdot \vec{t} \neq \vec{t}$. where $\varpi(\vec{w}) := \sum_{j=1}^{m} \beta_j$ is the weight of $\vec{w}$ (cf Section II-B).

III. Evaluation

The set of term distributions is equipped with a relation of *evaluation* $\vec{t} \twoheadrightarrow \vec{t}'$ that is defined in three steps as follows.

A. Atomic evaluation

First we define an asymmetric relation of *atomic evaluation* $t \triangleright \vec{t}'$ (between a pure term t and a term distribution $\vec{t}'$) from the inference rules of Table III.

These rules basically implement a deterministic call-by-value strategy, where function arguments are evaluated from the right to the left. (The argument of an application is always evaluated before the function[6]). Also notice that no reduction is ever performed in the body of an abstraction, in the second argument of a sequence, in the body of a let-construct, or in a branch of a match-construct. Moreover, atomic evaluation is substitutive: If $t \triangleright \vec{t}'$, then $t[x := w] \triangleright \vec{t}'[x := w]$ for all pure values w.

B. One step evaluation

The relation of *one step evaluation* $\vec{t} \succ \vec{t}'$ is defined as follows:

Definition III.1 (One step evaluation)**.** *Given two term distributions $\vec{t}$ and $\vec{t}'$, we say that $\vec{t}$* evaluates in one step *to $\vec{t}'$ and write $\vec{t} \succ \vec{t}'$ when there exist a scalar $\alpha \in \mathbb{C}$, a pure term s and two term distributions $\vec{s}'$ and $\vec{r}$ such that $\vec{t} = \alpha \cdot s + \vec{r}$, $\vec{t}' = \alpha \cdot \vec{s}' + \vec{r}$, and $s \triangleright \vec{s}'$.*

Notice that the relation of one step evaluation is also substitutive. In addition, the strict determinism of the relation of atomic evaluation $t \triangleright \vec{t}'$ implies that the relation of one step evaluation fulfills the following weak diamond property:

Lemma III.2 (Weak diamond)**.** *If $\vec{t} \succ \vec{t}'_1$ and $\vec{t} \succ \vec{t}'_2$, then one of the following holds: either $\vec{t}'_1 = \vec{t}'_2$; either $\vec{t}'_1 \succ \vec{t}'_2$ or $\vec{t}'_2 \succ \vec{t}'_1$; either $\vec{t}'_1 \succ \vec{t}''$ and $\vec{t}'_2 \succ \vec{t}''$ for some $\vec{t}''$.* □

Remark III.3. *In the decomposition $\vec{t} = \alpha \cdot s + \vec{r}$ of Definition III.1, we allow that $s \in \mathrm{dom}(\vec{r})$. So that for instance, we have the following. Let $t := (\lambda x\,.\,x)\,y$. Then,*

$$t = 1 \cdot (\lambda x\,.\,x)\,y \succ y$$

$$t = \frac{1}{2} \cdot (\lambda x\,.\,x)\,y + \frac{1}{2} \cdot (\lambda x\,.\,x)\,y \succ \frac{1}{2} \cdot y + \frac{1}{2} \cdot (\lambda x\,.\,x)\,y$$

$$t = 7 \cdot (\lambda x\,.\,x)\,y + (-6) \cdot (\lambda x\,.\,x)\,y \succ 7 \cdot y + (-6) \cdot (\lambda x\,.\,x)\,y$$

Remark III.4. *Given a pure term t, we write $Y_t := (\lambda x\,.\,t + xx)(\lambda x\,.\,t + xx)$, so that we have $Y_t \triangleright t + Y_t$ by construction. Then we observe that for all $\alpha \in \mathbb{C}$, we have*

$$0 \cdot Y_t = \alpha \cdot Y_t + (-\alpha) \cdot Y_t \succ \alpha \cdot (t + Y_t) + (-\alpha) \cdot Y_t = \alpha \cdot t + 0 \cdot Y_t$$

This example does not jeopardize the confluence of evaluation, since we also have

$$\alpha \cdot t + 0 \cdot Y_t \succ \alpha \cdot t + ((-\alpha) \cdot t + 0 \cdot Y_t) = 0 \cdot t + 0 \cdot Y_t$$

[6]This design choice is completely arbitrary, and we could have proceeded the other way around.

$$\frac{}{(\lambda x\,.\,\vec{t})\,v \;\triangleright\; \vec{t}[x:=v]} \qquad \frac{}{*;\vec{s} \;\triangleright\; \vec{s}} \qquad \frac{}{\texttt{let}\ (x,y)=(v,w)\ \texttt{in}\ \vec{s} \;\triangleright\; \vec{s}[x:=v,y:=w]}$$

$$\frac{}{\texttt{match inl}(v)\ \{\texttt{inl}(x_1)\mapsto \vec{s}_1 \mid \texttt{inr}(x_2)\mapsto \vec{s}_2\} \;\triangleright\; \vec{s}_1[x_1:=v]} \qquad \frac{}{\texttt{match inr}(v)\ \{\texttt{inl}(x_1)\mapsto \vec{s}_1 \mid \texttt{inr}(x_2)\mapsto \vec{s}_2\} \;\triangleright\; \vec{s}_2[x_2:=v]}$$

$$\frac{t \;\triangleright\; \vec{t}'}{s\,t \;\triangleright\; s\,\vec{t}'} \qquad \frac{t \;\triangleright\; \vec{t}'}{t\,v \;\triangleright\; \vec{t}'\,v} \qquad \frac{t \;\triangleright\; \vec{t}'}{t;\vec{s} \;\triangleright\; \vec{t}';\vec{s}} \qquad \frac{t \;\triangleright\; \vec{t}'}{\texttt{let}\ (x,y)=t\ \texttt{in}\ \vec{s} \;\triangleright\; \texttt{let}\ (x,y)=\vec{t}'\ \texttt{in}\ \vec{s}}$$

$$\frac{t \;\triangleright\; \vec{t}'}{\texttt{match}\ t\ \{\texttt{inl}(x_1)\mapsto \vec{s}_1 \mid \texttt{inr}(x_2)\mapsto \vec{s}_2\} \;\triangleright\; \texttt{match}\ \vec{t}'\ \{\texttt{inl}(x_1)\mapsto \vec{s}_1 \mid \texttt{inr}(x_2)\mapsto \vec{s}_2\}}$$

TABLE III
INFERENCE RULES OF THE RELATION OF ATOMIC EVALUATION $t \triangleright \vec{t}'$

C. Evaluation

Finally, the relation of evaluation $\vec{t} \twoheadrightarrow \vec{t}'$ is defined as the reflexive-transitive closure of the relation of one step evaluation $\vec{t} \succ \vec{t}'$, that is: $(\twoheadrightarrow) := (\succ)^*$.

Proposition III.5 (Linearity of evaluation)**.** *The relation $\vec{t} \twoheadrightarrow \vec{t}'$ is linear, in the sense that:*

1) $\vec{0} \twoheadrightarrow \vec{0}$
2) *If* $\vec{t} \twoheadrightarrow \vec{t}'$, *then* $\alpha \cdot \vec{t} \twoheadrightarrow \alpha \cdot \vec{t}'$ *for all* $\alpha \in \mathbb{C}$.
3) *If* $\vec{t}_1 \twoheadrightarrow \vec{t}'_1$ *and* $\vec{t}_2 \twoheadrightarrow \vec{t}'_2$, *then* $\vec{t}_1 + \vec{t}_2 \twoheadrightarrow \vec{t}'_1 + \vec{t}'_2$. □

Example III.6. *In our calculus, the Hadamard operator* $H : \mathbb{C}^2 \to \mathbb{C}^2$*, whose matrix is given by* $\mathrm{Mat}(H) := \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$*, is computed by the term*

$$H \;:=\; \lambda x\,.\,\texttt{if}\ x\ \left\{\ \tfrac{1}{\sqrt{2}}\cdot\texttt{tt} + \tfrac{1}{\sqrt{2}}\cdot\texttt{ff}\ \middle|\ \tfrac{1}{\sqrt{2}}\cdot\texttt{tt} + (-\tfrac{1}{\sqrt{2}})\cdot\texttt{ff}\right\}.$$

Indeed, for all $\alpha, \beta \in \mathbb{C}$*, we have*

$$\begin{aligned} &H\,(\alpha\cdot\texttt{tt} + \beta\cdot\texttt{ff}) = \alpha\cdot H\,\texttt{tt} + \beta\cdot H\,\texttt{ff} \\ &\twoheadrightarrow \alpha\cdot\texttt{if tt}\ \left\{\ \tfrac{1}{\sqrt{2}}\cdot\texttt{tt} + \tfrac{1}{\sqrt{2}}\cdot\texttt{ff}\ \middle|\ \tfrac{1}{\sqrt{2}}\cdot\texttt{tt} + \left(-\tfrac{1}{\sqrt{2}}\right)\cdot\texttt{ff}\right\} + \\ &\quad \beta\cdot\texttt{if ff}\ \left\{\ \tfrac{1}{\sqrt{2}}\cdot\texttt{tt} + \tfrac{1}{\sqrt{2}}\cdot\texttt{ff}\ \middle|\ \tfrac{1}{\sqrt{2}}\cdot\texttt{tt} + \left(-\tfrac{1}{\sqrt{2}}\right)\cdot\texttt{ff}\right\} \\ &\twoheadrightarrow \alpha\cdot\left(\tfrac{1}{\sqrt{2}}\cdot\texttt{tt} + \tfrac{1}{\sqrt{2}}\cdot\texttt{ff}\right) \;+\; \beta\cdot\left(\tfrac{1}{\sqrt{2}}\cdot\texttt{tt} + \left(-\tfrac{1}{\sqrt{2}}\right)\cdot\texttt{ff}\right) \\ &= \tfrac{1}{\sqrt{2}}(\alpha+\beta)\cdot\texttt{tt} + \tfrac{1}{\sqrt{2}}(\alpha-\beta)\cdot\texttt{ff} \end{aligned}$$

Theorem III.7 (Confluence of evaluation)**.** *If* $\vec{t} \twoheadrightarrow \vec{t}_1$ *and* $\vec{t} \twoheadrightarrow \vec{t}_2$, *then there is a term distribution* $\vec{t''}$ *such that* $\vec{t}_1 \twoheadrightarrow \vec{t''}$ *and* $\vec{t}_2 \twoheadrightarrow \vec{t''}$.

Proof. Writing $(\succ^?)$ the reflexive closure of $(\succ)$, it is clear from Lemma III.2 that $(\succ^?)$ fulfills the diamond property. Therefore, $(\twoheadrightarrow) = (\succ)^* = (\succ^?)^+$ fulfills the diamond property. □

D. Normal forms

From what precedes, it is clear that the *normal forms* of the relation of evaluation $\vec{t} \twoheadrightarrow \vec{t}'$ are the term distributions of the form $\vec{t} = \sum_{i=1}^n \alpha_i \cdot t_i$ where $t_i \not\triangleright$ for each $i = 1..n$. In particular, all value distributions $\vec{v}$ are normal forms (but they are far from being the only normal forms in the calculus). From the property of confluence, it is also clear that when a term distribution $\vec{t}$ reaches a normal form $\vec{t}'$, then this normal form is unique.

In what follows, we shall be more particularly interested in the closed term distributions $\vec{t}$ that reach a (unique) closed value distribution $\vec{v}$ through the process of evaluation.

IV. A SEMANTIC TYPE SYSTEM

In this section, we present the type system associated with the (untyped) language presented in Section II as well as the corresponding realizability semantics.

A. Structuring the space of value distributions

In what follows, we write: Λ the set of all closed pure terms; $\vec{\Lambda}$ the space of all closed term distributions; V $(\subseteq \Lambda)$ the set of all closed pure values, which we shall call *basis vectors*; and $\vec{\mathrm{V}}$ $(\subseteq \vec{\Lambda})$ the space of all closed value distributions, which we shall call *vectors*.

The space $\vec{\mathrm{V}}$ formed by all closed value distributions (i.e. vectors) is equipped with the inner product $\langle \vec{v} \mid \vec{w} \rangle$ and the pseudo-ℓ_2-norm $\|\vec{v}\|$ that are defined by

$$\langle \vec{v} \mid \vec{w} \rangle := \sum_{i=1}^n \sum_{j=1}^m \overline{\alpha_i}\,\beta_j\,\delta_{v_i,w_j}$$
$$\|\vec{v}\| := \sqrt{\langle \vec{v} \mid \vec{v} \rangle} = \sqrt{\textstyle\sum_{i=1}^n |\alpha_i|^2}$$

where $\vec{v} = \sum_{i=1}^n \alpha_i \cdot v_i$ and $\vec{w} = \sum_{j=1}^m \beta_j \cdot w_j$ (both in canonical form), and where δ_{v_i,w_j} is the Kronecker delta such that it is 1 if $v_i = w_j$ and 0 otherwise. Let us observe that the inner product behaves well with term constructors, so that e.g. $\langle \texttt{inl}(\vec{v}_1) \mid \texttt{inl}(\vec{v}_2) \rangle = \langle \vec{v}_1 \mid \vec{v}_2 \rangle$, and that values built from distinct term constructors are orthogonal, so that e.g. $\langle \texttt{inl}(\vec{v}_1) \mid \texttt{inr}(\vec{w}_2) \rangle = 0$. We can also infer that for all $\vec{v}, \vec{w} \in \vec{\mathrm{V}}$, we have $\|\texttt{inl}(\vec{v})\| = \|\texttt{inr}(\vec{v})\| = \|\vec{v}\|$ and $\|(\vec{v},\vec{w})\| = \|\vec{v}\|\,\|\vec{w}\|$.

Most of the constructions we shall perform hereafter will take place in the *unit sphere* $\mathcal{S}_1 \subseteq \vec{\mathrm{V}}$, that is defined by $\mathcal{S}_1 := \{\vec{v} \in \vec{\mathrm{V}} \;:\; \|\vec{v}\| = 1\}$. It is clear that for all $\vec{v}, \vec{w} \in \mathcal{S}_1$, we have $\texttt{inl}(\vec{v}) \in \mathcal{S}_1$, $\texttt{inr}(\vec{w}) \in \mathcal{S}_1$ and $(\vec{v},\vec{w}) \in \mathcal{S}_1$.

Given a set of vectors $X \subseteq \vec{\mathrm{V}}$, we also write $\mathrm{span}(X)$ the *span* of X, defined by $\left\{\sum_{i=1}^n \alpha_i \cdot \vec{v}_i \;:\; n \geq 0,\ \alpha_1,\ldots,\alpha_n \in \mathbb{C},\ \vec{v}_1,\ldots,\vec{v}_n \in X\right\} \subseteq \vec{\mathrm{V}}$, and $\flat X$ the *basis* of X, defined by $\bigcup_{\vec{v}\in X} \mathrm{dom}(\vec{v}) \subseteq \mathrm{V}$.

Note that by construction, $\mathrm{span}(X)$ is the smallest (weak) sub-vector space of $\vec{\mathrm{V}}$ such that $X \subseteq \mathrm{span}(X)$, whereas $\flat X$ is the smallest set of basis vectors such that $X \subseteq \mathrm{span}(\flat X)$.

B. The notion of unitary type

Definition IV.1 (Unitary types)**.** *A* unitary type *(or a* type*, for short) is defined by a* notation *A, together with a set of unitary vectors $[\![A]\!] \subseteq \mathcal{S}_1$, called the* unitary semantics of *A.*

Definition IV.2 (Realizability predicate)**.** *To each type A we associate a* realizability predicate $\vec{t} \Vdash A$ *(where $\vec{t}$ ranges over $\vec{\Lambda}$) that is defined by $\vec{t} \Vdash A$ if and only if $\vec{t} \twoheadrightarrow \vec{v}$ for some $\vec{v} \in [\![A]\!]$. The set or* realizers *of A, written $\{\Vdash A\}$, is then defined by $\{\vec{t} \in \vec{\Lambda} \ : \ \vec{t} \Vdash A\}$, that is, $\{\vec{t} \in \vec{\Lambda} \ : \ \exists \vec{v} \in [\![A]\!], \ \vec{t} \twoheadrightarrow \vec{v}\}$.*

Lemma IV.3. *For all types A, we have $[\![A]\!] = \{\Vdash A\} \cap \vec{\mathrm{V}}$.* □

C. Judgments, inference rules and derivations

Definition IV.4 (Judgments)**.** *A* judgment *is a notation J expressing some assertion, together with a* criterion of validity*, that defines whether the judgment J is valid or not.*

For instance, given any two types A and B, we can consider the following two judgments:

- The judgment $A \leq B$ ('A is a subtype of B'), that is valid when $[\![A]\!] \subseteq [\![B]\!]$.
- The judgment $A \simeq B$ ('A is equivalent to B'), that is valid when $[\![A]\!] = [\![B]\!]$.

(In Section V-A below, we shall also introduce a *typing judgment* written $\Gamma \vdash \vec{t} : A$). From the definition of both judgments $A \leq B$ and $A \simeq B$, it is clear that the judgment $A \simeq B$ is valid if and only if both judgments $A \leq B$ and $B \leq A$ are valid. Moreover:

Lemma IV.5. *Given any two types A and B:*

1) *$A \leq B$ is valid if and only if $\{\Vdash A\} \subseteq \{\Vdash B\}$.*
2) *$A \simeq B$ is valid if and only if $\{\Vdash A\} = \{\Vdash B\}$.* □

More generally, we call an *inference rule* any pair formed by a finite set of judgments $J_1, \ldots, J_n$, called the *premises* of the rule, and a judgment J_0, called the *conclusion*:

$$\frac{J_1 \quad \cdots \quad J_n}{J_0}$$

We say that an inference rule $\frac{J_1 \ \cdots \ J_n}{J_0}$ is *valid* when the joint validity of the premises $J_1, \ldots, J_n$ implies the validity of the conclusion J_0. As usual, inference rules can be assembled into derivations, and we shall say that a derivation is *valid* when all the inference rules that are used to build this derivation are valid. It is clear that when all the premises of a valid derivation are valid, then so is its conclusion. In particular, when a judgment has a valid derivation without premises, then this judgment is valid.

D. A simple algebra of types

In this section, we design a simple algebra of unitary types whose notations (i.e. the syntax) are given in Table IV and whose unitary semantics are given in Table V.

The choice we make in this paper follows from the structure of the calculus: each set of standard constructor/destructor canonically yields a type constructor: this gives : $\mathbb{U}$, the *unit type*, that is inhabited by the sole vector $*$; $A + B$, the *simple sum* of A and B ; $A \times B$, the *simple product* of A and B; $A \to B$, the space of all pure functions mapping A to B.

The next natural choice of type constructor is derived from the existence of linear combinations of terms. First, $\flat A$ is the *basis* of A, that is: the minimal set of basis vectors that generate all vectors of type A by (weak) linear combinations. Note that in general, $\flat A$ is not a subtype of A. Then, $\sharp A$ is the *unitary span* of A, that is: the type of all unitary vectors that can be formed as a (weak) linear combination of vectors of type A. Note that A is always a subtype of $\sharp A$.

$$A, B \quad ::= \quad \mathbb{U} \mid \flat A \mid \sharp A \mid A + B \mid A \times B \mid A \to B \mid A \Rightarrow B$$

TABLE IV
SYNTAX OF UNITARY TYPES

$$\begin{aligned}
[\![\mathbb{U}]\!] &:= \{*\} \qquad [\![\flat A]\!] := \flat [\![A]\!] \qquad [\![\sharp A]\!] := \mathrm{span}([\![A]\!]) \cap \mathcal{S}_1 \\
[\![A + B]\!] &:= \{\mathtt{inl}(\vec{v}) : \vec{v} \in [\![A]\!]\} \cup \{\mathtt{inr}(\vec{w}) : \vec{w} \in [\![B]\!]\} \\
[\![A \times B]\!] &:= \{(\vec{v}, \vec{w}) : \vec{v} \in [\![A]\!], \ \vec{w} \in [\![B]\!]\} \\
[\![A \to B]\!] &:= \{\lambda x \,.\, \vec{t} \ : \ \forall \vec{v} \in [\![A]\!], \ \vec{t}\langle x := \vec{v}\rangle \Vdash B\} \\
[\![A \Rightarrow B]\!] &:= \{(\textstyle\sum_{i=1}^n \alpha_i \cdot \lambda x \,.\, \vec{t_i}) \in \mathcal{S}_1 : \forall \vec{v} \in [\![A]\!], \\
&\qquad\qquad (\textstyle\sum_{i=1}^n \alpha_i \cdot \vec{t_i}\langle x := \vec{v}\rangle) \Vdash B\}
\end{aligned}$$

TABLE V
UNITARY SEMANTICS OF TYPES

The last non-trivial type is $A \Rightarrow B$: the space of all unitary function distributions mapping A to B. As lambda-terms are not distributives over linear combinations, this type is distinct from $\sharp(A \to B)$ (see next remark for a discussion). However, by construction, $A \to B$ is always a subtype of $A \Rightarrow B$.

Finally, we provide some syntactic sugar: the type of Booleans, the direct sum and the tensor product are defined by $\mathbb{B} := \mathbb{U} + \mathbb{U}$, $A \oplus B := \sharp(A + B)$, and $A \otimes B := \sharp(A \times B)$.

The type $\sharp\mathbb{B} = \sharp(\mathbb{U} + \mathbb{U}) = \mathbb{U} \oplus \mathbb{U}$ will be called the type of *unitary Booleans*. Notice that its semantics is given by the definition $[\![\sharp\mathbb{B}]\!] = \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\}) \cap \mathcal{S}_1$, that is, the set $\{\alpha \cdot \mathtt{tt} : |\alpha| = 1\} \cup \{\beta \cdot \mathtt{ff} : |\beta| = 1\} \cup \{\alpha \cdot \mathtt{tt} + \beta \cdot \mathtt{ff} : |\alpha|^2 + |\beta|^2 = 1\}$.

Remarks IV.6.

1) *The type constructors $\flat$ and $\sharp$ are monotonic and idempotent: $\flat\flat A \simeq \flat A$ and $\sharp\sharp A \simeq \sharp A$.*
2) *We always have the inclusion $A \leq \sharp A$, but the inclusion $\flat A \leq A$ does not hold in general. For instance, given any type A, we easily check that $\frac{3}{5} \cdot \left(\lambda x \,.\, \frac{5}{6} \cdot x\right) + \frac{4}{5} \cdot \left(\lambda x \,.\, \frac{5}{8} \cdot x\right) \in [\![A \Rightarrow A]\!]$, so that $(\lambda x \,.\, \frac{5}{6} \cdot x), (\lambda x \,.\, \frac{5}{8} \cdot x) \in \flat[\![A \Rightarrow A]\!] = [\![\flat(A \Rightarrow A)]\!]$. On the other hand, it is also clear that $(\lambda x \,.\, \frac{5}{6} \cdot x), (\lambda x \,.\, \frac{5}{8} \cdot x) \notin [\![A \Rightarrow A]\!]$ (unless $[\![A]\!] = \varnothing$). Therefore, $\flat(A \Rightarrow A) \not\leq A \Rightarrow A$.*
3) *We have the equivalence $\flat\sharp A \simeq \flat A$, but only the inclusion $A \leq \sharp\flat A$. More generally, the type constructor $\flat$ commutes with $+$ and $\times$: $\flat(A + B) \simeq \flat A + \flat B$ and $\flat(A \times B) \simeq \flat A \times \flat B$ but the type constructor $\sharp$ does not, since we only have the inclusions $\sharp A + \sharp B \leq \sharp(A + B)$ and $\sharp A \times \sharp B \leq \sharp(A \times B)$*
4) *The inclusions $A \Rightarrow B \leq \sharp(A \Rightarrow B)$ and $\sharp(A \to B) \leq \sharp(A \Rightarrow B)$ are strict in general (unless the type $A \Rightarrow B$ is empty). As a matter of fact, the two types $\sharp(A \to B)$ and $\sharp(A \Rightarrow B)$ have no interesting properties—for instance, they are* not *subtypes of $\sharp A \Rightarrow \sharp B$. In practice, the type constructor $\sharp$ is only used on top of an algebraic*

type, constructed using one of $\mathbb{U}$, $+$, *or* $\times$.

1) Pure types and simple types: In what follows, we shall say that a type A is *pure* when its unitary semantics only contains pure values, that is: when $[\![A]\!] \subseteq \mathrm{V}$. Equivalently, a type A is pure when the type equivalence $\flat A \simeq A$ is valid (or when $A \leq \flat B$ for some type B). We easily check that:

Lemma IV.7. *For all types* A *and* B*:*

1) *The types* $\mathbb{U}$, $\flat A$ *and* $A \to B$ *are pure.*
2) *If* A *and* B *are pure, then so are* $A + B$ *and* $A \times B$.
3) $\sharp A$ *and* $A \Rightarrow B$ *are not pure, unless they are empty.* □

A particular case of pure types are the *simple types*, that are syntactically defined from the following sub-grammar of the grammar of Table IV:

$$A, B ::= \mathbb{U} \mid A + B \mid A \times B \mid A \to B$$

It is clear from Lemma IV.7 that all simple types are pure types. The converse is false, since the type $\sharp\mathbb{U} \to \sharp\mathbb{U}$ is pure, although it is not generated from the above grammar.

2) Pure arrow vs unitary arrow: The pure arrow $A \to B$ and the unitary arrow $A \Rightarrow B$ only differ in the shape of the functions which they contain: the pure arrow $A \to B$ only contains pure abstractions whereas the unitary arrow $A \Rightarrow B$ contains arbitrary unitary distributions of abstractions mapping values of type A to realizers of type B. However, the functions that are captured by both sets $[\![A \to B]\!] \subseteq \mathrm{V}$ and $[\![A \Rightarrow B]\!] \subseteq \mathcal{S}_1$ are extensionally the same:

Proposition IV.8. *For all unitary distributions of abstractions* $\left(\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{t_i}\right) \in \mathcal{S}_1$*, one has:*

$$\left(\textstyle\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{t_i}\right) \in [\![A \Rightarrow B]\!]$$
$$\textit{iff} \quad \lambda x\,.\,\left(\textstyle\sum_{i=1}^n \alpha_i \cdot \vec{t_i}\right) \in [\![A \to B]\!]\,. \qquad \square$$

E. Representation of unitary operators

Recall that the type of *unitary Booleans* is defined as $\sharp\mathbb{B} = \sharp(\mathbb{U} + \mathbb{U}) = \mathbb{U} \oplus \mathbb{U}$, so that for all closed term distributions $\vec{t}$, we have $\vec{t} \Vdash \sharp\mathbb{B}$ iff

$$\begin{array}{ll}
\vec{t} \twoheadrightarrow \alpha \cdot \mathtt{tt} & \text{for some } \alpha \in \mathbb{C} \text{ s.t. } |\alpha| = 1, \quad \text{or} \\
\vec{t} \twoheadrightarrow \beta \cdot \mathtt{ff} & \text{for some } \beta \in \mathbb{C} \text{ s.t. } |\beta| = 1, \quad \text{or} \\
\vec{t} \twoheadrightarrow \alpha \cdot \mathtt{tt} + \beta \cdot \mathtt{ff} & \text{for some } \alpha, \beta \in \mathbb{C} \text{ s.t. } |\alpha|^2 + |\beta|^2 = 1\,.
\end{array}$$

We can observe that the unitary semantics of the type $\sharp\mathbb{B}$ simultaneously contains the vectors $\alpha \cdot \mathtt{tt}$ and $\alpha \cdot \mathtt{tt} + 0 \cdot \mathtt{ff}$, that can be considered as "morally" equivalent (although they are not according to the congruence $\equiv$). To identify such vectors, it is convenient to introduce the *Boolean projection* $\pi_{\mathbb{B}} : \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\}) \to \mathbb{C}^2$ defined by

$$\pi_{\mathbb{B}}(\alpha \cdot \mathtt{tt}) = (\alpha, 0), \qquad \pi_{\mathbb{B}}(\beta \cdot \mathtt{ff}) = (0, \beta),$$
$$\text{and} \qquad \pi_{\mathbb{B}}(\alpha \cdot \mathtt{tt} + \beta \cdot \mathtt{ff}) = (\alpha, \beta)$$

for all $\alpha, \beta \in \mathbb{C}$. By construction, the function $\pi_{\mathbb{B}} : \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\}) \to \mathbb{C}^2$ is linear, surjective, and neglects the difference between $\alpha \cdot \mathtt{tt} + 0 \cdot \mathtt{ff}$ and $\alpha \cdot \mathtt{tt}$ (and between $0 \cdot \mathtt{tt} + \beta \cdot \mathtt{ff}$ and $\beta \cdot \mathtt{ff}$). Moreover, the map $\pi_{\mathbb{B}} : \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\}) \to \mathbb{C}^2$ preserves the inner product, in the sense that for all $\vec{v}, \vec{w} \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$, we have

$$\langle \pi_{\mathbb{B}}(\vec{v}) \mid \pi_{\mathbb{B}}(\vec{w}) \rangle_{\mathbb{C}^2} = \langle \vec{v} \mid \vec{w} \rangle_{\vec{\mathrm{V}}}$$

Definition IV.9. *We say that a closed term distribution* $\vec{t}$ represents *a function* $F : \mathbb{C}^2 \to \mathbb{C}^2$ *when for all* $\vec{v} \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$*, there exists* $\vec{w} \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$ *such that*

$$\vec{t}\,\vec{v} \twoheadrightarrow \vec{w} \qquad \textit{and} \qquad \pi_{\mathbb{B}}(\vec{w}) = F(\pi_{\mathbb{B}}(\vec{v}))\,.$$

Remark IV.10. *From the bilinearity of application, it is clear that each function* $F : \mathbb{C}^2 \to \mathbb{C}^2$ *that is represented by a closed term distribution is necessarily linear.*

Recall that an operator $F : \mathbb{C}^2 \to \mathbb{C}^2$ is *unitary* when it preserves the inner product of $\mathbb{C}^2$, in the sense that $\langle F(u) \mid F(v) \rangle = \langle u \mid v \rangle$ for all $u, v \in \mathbb{C}^2$. Equivalently, an operator $F : \mathbb{C}^2 \to \mathbb{C}^2$ is unitary if and only if $\|F(1,0)\|_{\mathbb{C}^2} = \|F(0,1)\|_{\mathbb{C}^2} = 1$ and $\langle F(1,0) \mid F(0,1) \rangle_{\mathbb{C}^2} = 0$. The following propositions expresses that the types $\sharp\mathbb{B} \to \sharp\mathbb{B}$ and $\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$ capture unitary operators:

Proposition IV.11. *Given a closed* λ*-abstraction* $\lambda x\,.\,\vec{t}$*, we have* $\lambda x\,.\,\vec{t} \in [\![\sharp\mathbb{B} \to \sharp\mathbb{B}]\!]$ *if and only if there are two value distributions* $\vec{v}_1, \vec{v}_2 \in [\![\sharp\mathbb{B}]\!]$ *such that we have* $\vec{t}[x := \mathtt{tt}] \twoheadrightarrow \vec{v}_1$*,* $\vec{t}[x := \mathtt{ff}] \twoheadrightarrow \vec{v}_2$ *and* $\langle \vec{v}_1 \mid \vec{v}_2 \rangle = 0$*.* □

Theorem IV.12 (Characterization of the values of type $\sharp\mathbb{B} \to \sharp\mathbb{B}$)**.** *A closed* λ*-abstraction* $\lambda x\,.\,\vec{t}$ *is a value of type* $\sharp\mathbb{B} \to \sharp\mathbb{B}$ *if and only if it represents a unitary operator* $F : \mathbb{C}^2 \to \mathbb{C}^2$*.* □

Corollary IV.13 (Characterization of the values of type $\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$)**.** *A unitary distribution of abstractions* $\left(\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{t_i}\right) \in \mathcal{S}_1$ *is a value of type* $\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$ *if and only if it represents a unitary operator* $F : \mathbb{C}^2 \to \mathbb{C}^2$*.* □

V. Typing Judgements

In Section IV, we introduced a simple type algebra (Table IV) together with the corresponding unitary semantics (Table V). We also introduced the two judgments $A \leq B$ and $A \simeq B$. Now, it is time to introduce the typing judgment $\Gamma \vdash \vec{t} : A$ together with the corresponding notion of validity.

A. Typing Rules

As usual, we call a *typing context* (or a *context*) any finite function from the set of variables to the set of types. Contexts Γ are traditionally written $\Gamma = x_1 : A_1, \ldots, x_\ell : A_\ell$ where $\{x_1, \ldots, x_\ell\} = \mathrm{dom}(\Gamma)$ and where $A_i = \Gamma(x_i)$ for all $i = 1..\ell$. The empty context is written $\emptyset$, and the concatenation of two contexts Γ and Δ such that $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Delta) = \varnothing$ is defined by $\Gamma, \Delta := \Gamma \cup \Delta$ (that is: as the union of the underlying functions).

Similarly, we call a *substitution* any finite function from the set of variables to the set $\vec{\mathrm{V}}$ of closed value distributions. Substitutions σ are traditionally written $\sigma = \{x_1 := \vec{v}_1, \ldots, x_\ell := \vec{v}_\ell\}$ where $\{x_1, \ldots, x_\ell\} = \mathrm{dom}(\sigma)$ and where $\vec{v}_i = \sigma(x_i)$ for all $i = 1..\ell$. The empty substitution is written $\emptyset$, and the concatenation of two substitutions σ and τ such that

$\mathrm{dom}(\sigma) \cap \mathrm{dom}(\tau) = \varnothing$ is defined by $\sigma, \tau := \sigma \cup \tau$ (that is: as the union of the underlying functions). Given an open term distribution $\vec{t}$ and a substitution $\sigma = \{x_1 := \vec{v}_1, \dots, x_\ell := \vec{v}_\ell\}$, we write $\vec{t}\langle\sigma\rangle \;:=\; \vec{t}\langle x_1 := \vec{v}_1\rangle \cdots \langle x_\ell := \vec{v}_\ell\rangle$. Note that since the value distributions $\vec{v}_1, \dots, \vec{v}_\ell$ are closed, the order in which the (closed) bilinear substitutions $\langle x_i := \vec{v}_i\rangle$ $(i = 1..\ell)$ are applied to $\vec{t}$ is irrelevant.

Definition V.1 (Unitary semantics of a typing context)**.** *Given a typing context Γ, we call the* unitary semantics *of Γ and write $[\![\Gamma]\!]$ the set of substitutions defined by*

$$[\![\Gamma]\!] := \big\{\sigma \textit{ substitution } : \mathrm{dom}(\sigma) = \mathrm{dom}(\Gamma) \\ \textit{and } \forall x \in \mathrm{dom}(\sigma),\ \sigma(x) \in [\![\Gamma(x)]\!]\big\}\,.$$

Finally, we call the *strict domain* of a context Γ and write $\mathrm{dom}^\sharp(\Gamma)$ the set

$$\mathrm{dom}^\sharp(\Gamma) \;:=\; \{x \in \mathrm{dom}(\Gamma) \;:\; [\![\Gamma(x)]\!] \neq \flat[\![\Gamma(x)]\!]\}\,.$$

Intuitively, the elements of the set $\mathrm{dom}^\sharp(\Gamma)$ are the variables of the context Γ whose type is not a type of pure values. As we shall see below, these variables are the variables that must occur in all the term distributions that are well-typed in the context Γ. (This restriction is essential to ensure the validity of the rule (UnitLam), Table VI).

Definition V.2 (Typing judgments)**.** *A* typing judgment *is a triple $\Gamma \vdash \vec{t} : A$ formed by a typing context Γ, a (possibly open) term distribution $\vec{t}$ and a type A. This judgment is valid when:*

1) $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$*; and*
2) $\vec{t}\langle\sigma\rangle \Vdash A$ *for all* $\sigma \in [\![\Gamma]\!]$.

Proposition V.3. *The typing rules of Table VI are valid.* □

Remark V.4. *In the rule* (PureLam)*, the notation $\flat\Gamma \simeq \Gamma$ refers to the conjunction of premises $\flat A_1 \simeq A_1$ & $\cdots$ & $\flat A_\ell \simeq A_\ell$, where $A_1, \dots, A_\ell$ are the types occurring in the context Γ.*

Remark V.5. *The proof of validity of the typing rule* (UnitLam) *crucially relies on the fact that the body $\vec{t}$ of the abstraction $\lambda x\,.\,\vec{t}$ is a* raw *distribution (i.e. an expression that is considered only up to α-conversion, and not $\equiv$). This is the reason why we endowed term distributions (Section II-B) with the congruence $\equiv$ that is shallow, in the sense that it does not propagate in the bodies of abstractions, in the bodies of let-constructs, or in the branches of match-constructs.*

B. Simply-typed lambda-calculus

Recall that simple types (Section IV-D1) are generated from the following sub-grammar of the grammar of Table IV:

$$A, B ::= \mathbb{U} \mid A + B \mid A \times B \mid A \to B$$

By construction, all simple types A are pure types, in the sense that $\flat A \simeq A$. Since pure types allow the use of weakening and contraction, it is a straightforward exercise to check that any typing judgment $\Gamma \vdash t : A$ that is derivable in the simply-typed λ-calculus with sums and products is also derivable from the typing rules of Table VI.

C. Typing Church numerals

Let us recall that Church numerals $\bar{n}$ are defined for all $n \in \mathbb{N}$ by $\bar{n} \;:=\; \lambda f\,.\,\lambda x\,.\,f^n\,x$. From the typing rules of Table VI, we easily derive that $\vdash \bar{n} : (\mathbb{B} \to \mathbb{B}) \to (\mathbb{B} \to \mathbb{B})$ (by simple typing) and even that $\vdash \bar{n} : (\sharp\mathbb{B} \to \sharp\mathbb{B}) \to (\sharp\mathbb{B} \to \sharp\mathbb{B})$, using the fact that $\sharp\mathbb{B} \to \sharp\mathbb{B}$ is a pure type, that is subject to arbitrary weakenings and contractions. On the other hand, since we cannot use weakening or contraction for the non pure type $\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$, we cannot derive the judgments $\vdash \bar{n} : (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}) \to (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B})$ and $\vdash \bar{n} : (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}) \Rightarrow (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B})$ but for $n = 1$. (cf. Fact A.11 in Appendix C).

D. Orthogonality as a Typing Rule

The typing rules of Table VI allow us to derive that the terms $I := \lambda x\,.\,x$, $K_{\mathtt{tt}} := \lambda x\,.\,\mathtt{tt}$, $K_{\mathtt{ff}} := \lambda x\,.\,\mathtt{ff}$ and $N := \lambda x\,.\,\mathtt{if}\ x\ \{\mathtt{ff} \mid \mathtt{tt}\}$ have type $\mathbb{B} \to \mathbb{B}$; they even allow us to derive that I has type $\sharp\mathbb{B} \to \sharp\mathbb{B}$, but they do not allow us (yet) to derive that the Boolean negation N or the Hadamard H have type $\sharp\mathbb{B} \to \sharp\mathbb{B}$. For that, we need to introduce a new form of judgment: *orthogonality judgments*.

Definition V.6 (Orthogonality judgments)**.** *An* orthogonality judgment *is a sextuple*

$$\Gamma \vdash (\Delta_1 \vdash \vec{t}_1) \perp (\Delta_2 \vdash \vec{t}_2) : A$$

formed by three typing contexts Γ, Δ_1 and Δ_2, two (possibly open) term distributions $\vec{t}_1$, $\vec{t}_2$ and a type A. This judgment is valid when:

1) *both judgments $\Gamma, \Delta_1 \vdash \vec{t}_1 : A$ and $\Gamma, \Delta_2 \vdash \vec{t}_2 : A$ are valid; and*
2) *for all $\sigma \in [\![\Gamma]\!]$, $\sigma_1 \in [\![\Delta_1]\!]$ and $\sigma_2 \in [\![\Delta_2]\!]$, if $\vec{t}_1\langle\sigma, \sigma_1\rangle \twoheadrightarrow \vec{v}_1$ and $\vec{t}_2\langle\sigma, \sigma_2\rangle \twoheadrightarrow \vec{v}_2$, then $\langle \vec{v}_1 \mid \vec{v}_2\rangle = 0$.*

When both contexts Δ_1 and Δ_2 are empty, the orthogonality judgment $\Gamma \vdash (\Delta_1 \vdash \vec{t}_1) \perp (\Delta_2 \vdash \vec{t}_2) : A$ is simply written $\Gamma \vdash \vec{t}_1 \perp \vec{t}_2 : A$.

With this definition, we can prove a new typing rule, which can be used to type Hadamard:

Proposition V.7. *The rule* (UnitaryMatch) *given below is valid.*

$$\frac{\Gamma \vdash \vec{t} : A_1 \oplus A_2 \quad \Delta \vdash (x_1 : \sharp A_1 \vdash \vec{s}_1) \perp (x_2 : \sharp A_2 \vdash \vec{s}_2) : \sharp C}{\Gamma, \Delta \vdash \mathtt{match}\ \vec{t}\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\} : \sharp C}$$ □

Example V.8. *We have $\vdash \mathtt{tt} \perp \mathtt{ff} : \mathbb{B}$. Consider the terms $|+\rangle = \frac{1}{\sqrt{2}} \cdot \mathtt{tt} + \frac{1}{\sqrt{2}} \cdot \mathtt{ff}$ and $|-\rangle = \frac{1}{\sqrt{2}} \cdot \mathtt{tt} + (-\frac{1}{\sqrt{2}}) \cdot \mathtt{ff}$. Then we can prove that $\vdash |+\rangle \perp |-\rangle : \sharp\mathbb{B}$.*

We can also prove that

$$\vdash (x : \sharp\mathbb{U} \vdash x; |+\rangle) \perp (y : \sharp\mathbb{U} \vdash y; |-\rangle) : \sharp\mathbb{B}$$

Using this fact, and the rule (UnitaryMatch) *from Proposition V.7, we can derive the type $\sharp\mathbb{B} \to \sharp\mathbb{B}$ for the Hadamard gate H defined in Example III.6. Recall that $\sharp\mathbb{B} = \sharp(\mathbb{U}+\mathbb{U}) = \mathbb{U} \oplus \mathbb{U}$.*

$$\frac{}{x : A \vdash x : A}\ \text{(Axiom)} \qquad \frac{\Gamma \vdash \vec{t} : A \quad A \leq A'}{\Gamma \vdash \vec{t} : A'}\ \text{(Sub)} \qquad \frac{\Gamma, x : A \vdash \vec{t} : B \quad \flat\Gamma \simeq \Gamma}{\Gamma \vdash \lambda x\,.\,\vec{t} : A \to B}\ \text{(PureLam)} \qquad \frac{\Gamma, x : A \vdash \vec{t} : B}{\Gamma \vdash \lambda x\,.\,\vec{t} : A \Rightarrow B}\ \text{(UnitLam)}$$

$$\frac{\Gamma \vdash \vec{s} : A \Rightarrow B \quad \Delta \vdash \vec{t} : A}{\Gamma, \Delta \vdash \vec{s}\vec{t} : B}\ \text{(App)} \qquad \frac{}{\vdash\ * : \mathbb{U}}\ \text{(Void)} \qquad \frac{\Gamma \vdash \vec{t} : \mathbb{U} \quad \Delta \vdash \vec{s} : A}{\Gamma, \Delta \vdash \vec{t}; \vec{s} : A}\ \text{(Seq)} \qquad \frac{\Gamma \vdash \vec{t} : \sharp\mathbb{U} \quad \Delta \vdash \vec{s} : \sharp A}{\Gamma, \Delta \vdash \vec{t}; \vec{s} : \sharp A}\ \text{(SeqSharp)}$$

$$\frac{\Gamma \vdash \vec{v} : A \quad \Delta \vdash \vec{w} : B}{\Gamma, \Delta \vdash (\vec{v}, \vec{w}) : A \times B}\ \text{(Pair)} \qquad \frac{\Gamma \vdash \vec{t} : A \times B \quad \Delta, x : A, y : B \vdash \vec{s} : C}{\Gamma, \Delta \vdash \mathtt{let}\ (x, y) = \vec{t}\ \mathtt{in}\ \vec{s} : C}\ \text{(LetPair)} \qquad \frac{\Gamma \vdash \vec{t} : A \otimes B \quad \Delta, x : \sharp A, y : \sharp B \vdash \vec{s} : \sharp C}{\Gamma, \Delta \vdash \mathtt{let}\ (x, y) = \vec{t}\ \mathtt{in}\ \vec{s} : \sharp C}\ \text{(LetTens)}$$

$$\frac{\Gamma \vdash \vec{v} : A}{\Gamma \vdash \mathtt{inl}(\vec{v}) : A + B}\ \text{(InL)} \qquad \frac{\Gamma \vdash \vec{w} : B}{\Gamma \vdash \mathtt{inr}(\vec{w}) : A + B}\ \text{(InR)} \qquad \frac{\Gamma \vdash \vec{t} : A + B \quad \Delta, x_1 : A \vdash \vec{s}_1 : C \quad \Delta, x_2 : B \vdash \vec{s}_2 : C}{\Gamma, \Delta \vdash \mathtt{match}\ \vec{t}\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\} : C}\ \text{(PureMatch)}$$

$$\frac{\Gamma \vdash \vec{t} : B \quad \flat A \simeq A}{\Gamma, x : A \vdash \vec{t} : B}\ \text{(Weak)} \qquad \frac{\Gamma, x : A, y : A \vdash \vec{t} : B \quad \flat A \simeq A}{\Gamma, x : A \vdash \vec{t}[y := x] : B}\ \text{(Contr)}$$

TABLE VI
SOME VALID TYPING RULES

VI. UNIFYING MODEL OF CLASSICAL AND QUANTUM CONTROL

We showed in Section V-B that the unitary linear algebraic lambda calculus strictly contains the simply-typed lambda calculus. With Theorem IV.12 and Corollary IV.13 we expressed how the "only" valid functions were unitary maps, and in Section V-D we hinted at how to type orthogonality with the model. This section is devoted to showing how the model can be used as a model for quantum computation, with the model providing an *operational semantics* to a high-level operation on circuits: the control of a circuit.

A. A Quantum Lambda-Calculus

The language we consider, called λ_Q, is a circuit-description language similar to QWIRE [6] or Proto-Quipper [26]. Formally, the types of λ_Q are defined from the following grammar:

$$A, B ::= \mathbb{U} \mid A \to B \mid A \times B \mid \mathsf{bit} \mid A_Q \multimap B_Q$$
$$A_Q, B_Q ::= \mathsf{qbit} \mid A_Q \otimes B_Q$$

The types denoted by A, B are the usual simple types, which we call *classical types*. (Note that they contain a type bit of classical bits, that corresponds to the type $\mathbb{U}+\mathbb{U}$ in our model.) The types denoted by A_Q, B_Q are the *quantum types*; they basically consist in tensor products of the type qbit of quantum bits. As the former types are duplicable while the latter are non-duplicable, we define a special (classical) function-type $A_Q \multimap B_Q$ between quantum types.

The term syntax for λ_Q is defined from the following grammar:

$$\begin{aligned} t, r, s ::= {} & x \mid * \mid \lambda x.t \mid t\,r \mid (t, r) \mid \pi_1(t) \mid \pi_2(t) \\ & \mid \mathtt{tt} \mid \mathtt{ff} \mid \mathtt{if}\ t\ \{r \mid s\} \\ & \mid t \otimes r \mid \mathsf{let}\ x \otimes y = t\ \mathsf{in}\ r \\ & \mid \mathsf{new}(t) \mid U(t) \mid \lambda^Q x.t \mid t@r \end{aligned}$$

The first two lines of the definition describe the usual constructions of the simply-typed lambda calculus with (ordinary) pairs. The last two lines adds the quantum specificities: a tensor for dealing with systems of several quantum bits (together with the corresponding destructor), an operator new to create a new quantum bit, and a family of operators $U(t)$ to apply a given unitary operator on t. We also provide a special lambda abstraction λ^Q to make a closure out of a quantum computation, as well as a special application to apply such a closure. Note that for simplicity, we only consider unary quantum operators—that is: operators on the type qbit—, but this can be easily extended to quantum operators acting on tensor products of the form $\mathsf{qbit}^{\otimes n}$. Also note that we do not consider measurements, for our realizability model does not natively support it.

The language λ_Q features two kinds of typing judgments: a *classical judgment* $\Delta \vdash_C t : A$, where Δ is a typing context of classical types and where A is a classical type, and a *quantum judgment* $\Delta|\Gamma \vdash_Q t : A_Q$, where Δ is a typing context of classical types, Γ a typing context of quantum types, and where A_Q is a quantum type. An empty typing context is always denoted by $\emptyset$. As usual, we write Γ, Δ for $\Gamma \cup \Delta$ (when $\Gamma \cap \Delta = \emptyset$), and we use the notation $FV(t) : \mathsf{qbit}$ to represent the quantum context $x_1 : \mathsf{qbit}, \ldots, x_n : \mathsf{qbit}$ made up of the finite set $FV(t) = \{x_1, \ldots, x_n\}$.

The typing rules for classical judgements are standard and are given in the Appendix D. Rules for quantum judgements are given in the Table VII. The last three rules allows to navigate between classical and quantum judgments. Note that in the above rules, classical variables (declared in the Δ's) can be freely duplicated whereas quantum variables (declared in the Γ's) cannot. Also note that in λ_Q, pure quantum computations are essentially first-order.

The first of the last three rules makes a qbit out of a bit, the second rule makes a closure out of a quantum computation, while the third rule opens a closure containing a quantum computation. These last two operations give a hint of higher-order to quantum computations in λ_Q.

A *value* is a term belonging to the grammar:

$$u, v \quad ::= \quad x \mid \lambda x.t \mid \lambda^Q x.t \mid (u, v) \mid * \mid u \otimes v\,.$$

The language λ_Q is equipped with the standard operational semantics presented in [27]: the quantum environment is

$$\frac{}{\Delta|x : A_Q \vdash_Q x : A_Q} \qquad \frac{\Delta|\Gamma_1 \vdash_Q s : A_Q \quad \Delta|\Gamma_2 \vdash_Q t : B_Q}{\Delta|\Gamma_1, \Gamma_2 \vdash_Q s \otimes t : A_Q \otimes B_Q}$$

$$\frac{\Delta|\Gamma \vdash_Q t : \mathsf{qbit}}{\Delta|\Gamma \vdash_Q U(t) : \mathsf{qbit}}$$

$$\frac{\Delta|\Gamma_1 \vdash_Q s : A_Q \otimes B_Q \quad \Delta|\Gamma_2, x : A_Q, y : B_Q \vdash_Q t : C_Q}{\Delta|\Gamma_1, \Gamma_2 \vdash_Q \mathsf{let}\ x \otimes y = s\ \mathsf{in}\ t : C_Q}$$

$$\frac{\Delta \vdash_C t : \mathsf{bit}}{\Delta|\emptyset \vdash_Q \mathsf{new}(t) : \mathsf{qbit}} \qquad \frac{\Delta|x : A_Q \vdash_Q t : B_Q}{\Delta \vdash_C \lambda^Q x.t : A_Q \multimap B_Q}$$

$$\frac{\Delta \vdash_C s : A_Q \multimap B_Q \quad \Delta|\Gamma \vdash_Q t : A_Q}{\Delta|\Gamma \vdash_Q s@t : B_Q}$$

TABLE VII
TYPING RULES FOR λ_Q

$$\begin{aligned}
[Q, L, C\{(\lambda x.t)u\}] &\to [Q, L, C\{t[x := u]\}] \\
[Q, L, C\{(\lambda^Q x.t)@u\}] &\to [Q, L, C\{t[x := u]\}] \\
[Q, L, C\{\pi_1(u, v)\}] &\to [Q, L, C\{u\}] \\
[Q, L, C\{\pi_2(u, v)\}] &\to [Q, L, C\{v\}] \\
[Q, L, C\{\mathtt{if\ tt}\ \{t \mid r\}\}] &\to [Q, L, C\{t\}] \\
[Q, L, C\{\mathtt{if\ ff}\ \{t \mid r\}\}] &\to [Q, L, C\{r\}] \\
[Q, L, C\{\mathsf{let}\ x \otimes y = u \otimes v\ \mathsf{in}\ s\}] &\to [Q, L, C\{s[x := u, y := v]\}] \\
[Q, L, C\{\mathsf{new}(\mathtt{tt})\}] &\to [Q \otimes |0\rangle, L \cup \{x \mapsto n{+}1\}, C\{x\}] \\
[Q, L, C\{\mathsf{new}(\mathtt{ff})\}] &\to [Q \otimes |1\rangle, L \cup \{x \mapsto n{+}1\}, C\{x\}] \\
[Q, L, C\{U(x)\}] &\to [Q', L, C\{x\}]
\end{aligned}$$

where Q' is obtained by applying U to the quantum bit L(x)

TABLE VIII
OPERATIONAL SEMATICS OF λ_Q

separated from the term, in the spirit of the QRAM model of [4]. Formally, a *program* is defined as a triplet $[Q, L, t]$ where t is a term, L is a bijection from $FV(t)$ to $\{1, \ldots, n\}$ and Q is an n-quantum bit system: a normalized vector in the 2^n-dimensional vector space $(\mathbb{C}^2)^{\otimes n}$. We say that a program $[Q, L, t]$ is well-typed of type A_Q when the judgment $\emptyset|FV(t) : \mathsf{qbit} \vdash_Q t : A_Q$ is derivable. In particular, well-typed programs correspond to *quantum* typing judgements, closed with respect to classically-typed term-variables.

The operational semantics is call-by-value and relies on applicative contexts, that are defined as follows:

$$\begin{aligned}
C\{\cdot\} ::= {} & \{\cdot\} \mid C\{\cdot\}u \mid rC\{\cdot\} \mid (C\{\cdot\}, r) \mid (u, C\{\cdot\}) \\
& \mid \pi_1(C\{\cdot\}) \mid \pi_2(C\{\cdot\}) \mid \mathtt{if}\ C\{\cdot\}\ \{t \mid r\} \mid C\{\cdot\} \otimes r \\
& \mid u \otimes C\{\cdot\} \mid \mathsf{let}\ x \otimes y = C\{\cdot\}\ \mathsf{in}\ t \mid \mathsf{new}(C\{\cdot\}) \\
& \mid U(C\{\cdot\}) \mid C\{\cdot\}@u \mid r@C\{\cdot\}
\end{aligned}$$

The operational semantics of the calculus is formally defined from the rules given in Table VIII. The language λ_Q satisfies the usual safety properties, proved as in [27].

Theorem VI.1 (Safety properties)**.** *If $[Q, L, t] : A_Q$ and $[Q, L, t] \to [Q', L', r]$, then $[Q', L', r] : A_Q$. Moreover, whenever a program $[Q, L, t]$ is well-typed, either t is already a value or it reduces to some other program.* □

B. Modelling λ_Q

The realizability model based on the unitary linear-algebraic lambda-calculus is a model for the quantum lambda-calculus λ_Q. We write $(\!|t|\!)$ for the translation of a term of λ_Q into its model. The model can indeed not only accomodate classical features, using pure terms, but also quantum states, using linear combinations of terms.

We map qbit to $\sharp\mathbb{B}$ and bit to $\mathbb{B}$. This makes bit a subtype of qbit: the model captures the intuition that booleans are "pure" quantum bits. Classical arrows $\to$ are mapped to $\to$ and classical product $\times$ is mapped to the product of the model, in the spirit of the encoding of simply-typed lambda-calculus. Finally, the tensor of λ_Q is mapped to the tensor of the model.

The interesting type is $A_Q \multimap B_Q$. We need this type to be both classical *and* capture the fact that a term of this type is a pure quantum computation from A_Q to B_Q, that is, a unitary map. The encoding we propose consists in using "thunk", as proposed by [28]. Formally, the translation of types is as follows: $(\!|\mathsf{bit}|\!) = \mathbb{B}$, $(\!|A \times B|\!) = (\!|A|\!) \times (\!|B|\!)$, $(\!|A \to B|\!) = (\!|A|\!) \to (\!|B|\!)$, $(\!|A_Q \multimap B_Q|\!) = \mathbb{U} \to ((\!|A_Q|\!) \Rightarrow (\!|B_Q|\!))$, $(\!|\mathsf{qbit}|\!) = \sharp\mathbb{B}$, $(\!|A_Q \otimes B_Q|\!) = (\!|A_Q|\!) \otimes (\!|B_Q|\!) = \sharp((\!|A_Q|\!) \times (\!|B_Q|\!))$, and $(\!|\mathbb{U}|\!) = \mathbb{U}$.

Lemma VI.2. *For all classical types A, $\flat(\!|A|\!) \simeq (\!|A|\!)$.* □

Lemma VI.3. *For all qbit types A_Q, $\sharp(\!|A_Q|\!) \simeq (\!|A_Q|\!)$.* □

The classical structural term constructs of λ_Q are translated literally: $(\!|x|\!) = x$, $(\!|*|\!) = *$, $(\!|\lambda x.t|\!) = \lambda x.(\!|t|\!)$, $(\!|tr|\!) = (\!|t|\!)(\!|r|\!)$, $(\!|(t, r)|\!) = ((\!|t|\!), (\!|r|\!))$, $(\!|\mathtt{if}\ t\ \{r \mid s\}|\!) = \mathtt{match}\ (\!|t|\!)\ \{\mathtt{inl}(z_1) \mapsto z_1; (\!|r|\!) \mid \mathtt{inr}(z_2) \mapsto z_2; (\!|s|\!)\}$ with z_1 and z_2 fresh variables, $(\!|\mathtt{tt}|\!) = \mathtt{inl}(*)$, $(\!|\mathtt{ff}|\!) = \mathtt{inr}(*)$, $(\!|\pi_i(t)|\!) = \mathsf{let}\ (x_1, x_2) = (\!|t|\!)\ \mathsf{in}\ x_i$. Finally, the term constructs related to quantum bits make use of the algebraic aspect of the language. First, new is simply the identity, since booleans are subtypes of quantum bits: $(\!|\mathsf{new}(t)|\!) = (\!|t|\!)$. Then, the translation of the unitary operators is done with the construction already encountered in e.g. Example III.6: $(\!|U(t)|\!) = \bar{U}(\!|t|\!)$ where $\bar{U}$ is defined as follows. If $U = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, then $\bar{U} = \lambda x.\mathtt{match}\ x\ \{\mathtt{inl}(x_1) \mapsto a{\cdot}\mathtt{inl}(x_1){+}c{\cdot}\mathtt{inr}(x_1) \mid \mathtt{inr}(x_2) \mapsto b{\cdot}\mathtt{inl}(x_2){+}d{\cdot}\mathtt{inr}(x_2)\}$.

Then, the tensor is defined with the pairing construct, which is distributive: $(\!|t \otimes r|\!) = ((\!|t|\!), (\!|r|\!))$ and $(\!|\mathsf{let}\ x \otimes y = s\ \mathsf{in}\ t|\!) = \mathsf{let}\ (x, y) = (\!|s|\!)\ \mathsf{in}\ (\!|t|\!)$. Finally, the quantum closure and applications are defined by remembering the use of the thunk: $(\!|\lambda^Q x.t|\!) = \lambda z x.(\!|t|\!)$, where z is a fresh variable, and $(\!|t@r|\!) = ((\!|t|\!)*)(\!|r|\!)$: one first "open" the thunk before applying the function.

We also define the translation of typing contexts as follows: if $\Gamma = \{x_i : A_i\}_i$, we write $(\!|\Gamma|\!)$ for $\{x_i : (\!|A_i|\!)\}_i$, and we write $(\!|\Delta|\Gamma|\!)$ for $(\!|\Delta|\!), (\!|\Gamma|\!)$. Finally, a program is translated as follows: $(\!|[\sum_{i=1}^m \alpha_i.|y_1^i, \ldots, y_n^i\rangle, \{x_1 := p(1), \ldots, x_n := p(n)\}, t]|\!) = \sum_{i=1}^m \alpha_i \cdot (\!|t|\!)[x_1 := \bar{y}^i_{p(1)}, \ldots, x_n := \bar{y}^i_{p(n)}]$ where p is a permutation of n and $\bar{0} = \mathtt{tt}$ and $\bar{1} = \mathtt{ff}$.

Example VI.4. *Let P be the program $[\alpha|00\rangle + \beta|11\rangle, \{x := 1, y := 2\}, (x \otimes y)]$. It consists on a pair of the two quantum bits given in the quantum context on the first component of the triple. The translation of this program is as follows. $(\!|P|\!) =$*

$\alpha \cdot (x,y)[x := \mathtt{tt}, y := \mathtt{tt}] + \beta \cdot (x,y)[x := \mathtt{ff}, y := \mathtt{ff}] = \alpha \cdot (\mathtt{tt}, \mathtt{tt}) + \beta \cdot (\mathtt{ff}, \mathtt{ff})$.

The translation is compatible with typing and rewriting. This is to be put in reflection with Theorem IV.12: not only the realizability model captures unitarity, but it is expressive enough to comprehend a higher-order quantum programming language.

Theorem VI.5. *Translation preserves typeability:*

1) *If* $\Gamma \vdash_Q t : A_Q$ *then* $(\!|\Gamma|\!) \vdash (\!|t|\!) : (\!|A_Q|\!)$.
2) *If* $\Delta | \Gamma \vdash_C t : A$ *then* $(\!|\Delta|\!), (\!|\Gamma|\!) \vdash (\!|t|\!) : (\!|A|\!)$.
3) *If* $[Q, L, t] : A$ *then* $\vdash (\!|[Q, L, t]|\!) : (\!|A|\!)$. □

Theorem VI.6 (Adequacy)**.** *If* $[Q, L, t] \to [Q', L', r]$*, then* $(\!|[Q, L, t]|\!) \twoheadrightarrow (\!|[Q', L', r]|\!)$. □

C. A Circuit-Description Language

Quantum algorithms do not only manipulate quantum bits: they also manipulate *circuits*. A quantum circuit is a sequence of elementary operations that are buffered before being sent to the quantum memory. If one can construct a quantum circuit by concatenating elementary operations, several high-level operations on circuits are allowed for describing quantum algorithms: repetition, control (discussed in Section VI-D), inversion, *etc*.

In recent years, several quantum programming languages have been designed to allow the manipulation of circuits: Quipper [5] and its variant ProtoQuipper [26], QWIRE [6], *etc*. These languages share a special function-type $\mathsf{Circ}(A, B)$ standing for the type of circuits from wires of type A to wires of type B. Two built-in constructors are used to go back and forth between circuits and functions acting on quantum bits:

- $\mathsf{box} : (A_Q \multimap B_Q) \to \mathsf{Circ}(A_Q, B_Q)$. Its operational semantics is to evaluate the input function on a phantom element of type A, collect the list of elementary quantum operations to be performed and store them in the output circuit.
- $\mathsf{unbox} : \mathsf{Circ}(A_Q, B_Q) \to (A_Q \multimap B_Q)$. This operator is the dual: it takes a circuit — a list of elementary operations — and return a concrete function.

The advantage of distinguishing between functions and circuits is that a circuit is a concrete object: it is literally a list of operations that can be acted upon. A function is a suspended computation: it is *a priori* not possible to inspect its body.

The language λ_Q does not technically possess a type constructor for circuits: the typing construct $\multimap$ is really a lambda-abstraction. However, it is very close to being a circuit: one could easily add a typing construct Circ in the classical type fragment and implement operators box and unbox, taking inspiration for the operational semantics on what has been done by [26] for PROTOQUIPPER.

How would this be reflected in the realizability model? We claim that the translation of the type $\mathsf{Circ}(A_Q, B_Q)$ can be taken to be the same as the translation of $A_Q \multimap B_Q$, the operator box and unbox simply being the identity. The realizability model is then rich enough to express several high-level operations on circuits: this permits to extend the language λ_Q. The fact that the model "preserves unitarity" (Theorem IV.12) ensuring the soundness of the added constructions.

In what follows, by abuse of notation, we identify $\mathsf{Circ}(A_Q, B_Q)$ and $A_Q \multimap B_Q$.

D. Control Operator

Suppose that we are given a closed term t of λ_Q with type $\mathsf{qbit} \multimap \mathsf{qbit}$. This function corresponds to a unitary matrix $U = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, sending $|0\rangle$ to $a|0\rangle + c|1\rangle$ and $|1\rangle$ to $b|0\rangle + d|1\rangle$. We might want to write $\mathsf{ctl}(t)$ of type $(\mathsf{qbit} \otimes \mathsf{qbit}) \multimap (\mathsf{qbit} \otimes \mathsf{qbit})$ behaving as the control of U, whose behavior is to send $|0\rangle \otimes \phi$ to $|0\rangle \otimes \phi$ and $|1\rangle \otimes \phi$ to $|1\rangle \otimes (U\phi)$: if the first input quantum bit is in state $|0\rangle$, control-U acts as the identity. If the first input quantum bit is in state $|1\rangle$, control-U performs U on the second quantum bit.

This is really a "quantum test" [29]. It has been formalized in the context of linear algebraic lambda-calculi by [1]. It can be ported to the unitary linear algebraic lambda-calculus as follows:

$$\begin{aligned} \overline{\mathsf{ctl}} \quad := \lambda f.\lambda z.\mathtt{let}((x,y)) = z \ \mathtt{in} \\ \mathtt{match}\ x\ \{\mathtt{inl}(z_1) \mapsto (\mathtt{inl}(z_1), fy) \\ |\mathtt{inr}(z_2) \mapsto (\mathtt{inr}(z_2), y)\} \end{aligned}$$

and $\overline{\mathsf{ctl}}$ can be given the type

$$(\sharp A \Rightarrow \sharp B) \to ((\mathbb{B} \otimes A) \Rightarrow (\mathbb{B} \otimes B)).$$

Note how the definition is very semantical: the control operation is literally defined as a test on the first quantum bit.

We can then add an opaque term construct $\mathsf{ctl}(s)$ to λ_Q with typing rule

$$\frac{\Delta \vdash_C t : A_Q \multimap B_Q}{\Delta \vdash_C \mathsf{ctl}(t) : (\mathsf{qbit} \otimes A_Q) \multimap (\mathsf{qbit} \otimes B_Q)}.$$

The translation of this new term construct is then $(\!|\mathsf{ctl}(t)|\!) = \lambda z.(\overline{\mathsf{ctl}}((\!|t|\!)*))$ with z a fresh variable, and Theorem VI.6 still holds.

VII. CONCLUSIONS

In this paper we have presented a language based on Lineal [1], [2]. Then, we have given a set of unitary types and proposed a realizability semantics associating terms and types.

The main result of this paper can be pinpointed to Theorem IV.12 and Corollary IV.13, which, together with normalization, progress, and subject reduction of the calculus (which are axiomatic properties in realizability models), imply that every term of type $\sharp B \to \sharp B$ represent a unitary operator. In addition, the Definition V.6 of orthogonal judgements led to Proposition V.7 proving rule (UnitaryMatch). Indeed, one of the main historic drawbacks for considering a calculus with quantum control has been to define the notion of orthogonality needed to encode unitary gates (cf., for example, [29]).

Finally, as an example to show the expressiveness of the language, we have introduced λ_Q and showed that the calculus presented in this paper can be considered as a denotational semantics of it.

References

[1] P. Arrighi and G. Dowek, "Linear-algebraic λ-calculus: higher-order, encodings, and confluence." in *Rewriting Techniques and Applications*, A. Voronkov, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 17–31.

[2] ——, "Lineal: A linear-algebraic lambda-calculus," *Logical Methods in Computer Science*, vol. 13, 2017.

[3] B. Valiron, "A typed, algebraic, computational lambda-calculus," *Mathematical Structures in Computer Science*, vol. 23, no. 2, pp. 504–554, 2013.

[4] E. H. Knill, "Conventions for quantum pseudocode," Los Alamos National Laboratory, Tech. Rep. LA-UR-96-2724, 1996.

[5] A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron, "Quipper: a scalable quantum programming language," *ACM SIGPLAN Notices (PLDI'13)*, vol. 48, no. 6, pp. 333–342, 2013.

[6] J. Paykin, R. Rand, and S. Zdancewic, "Qwire: A core language for quantum circuits," in *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, ser. POPL 2017. New York, NY, USA: ACM, 2017, pp. 846–858.

[7] A. van Tonder, "A lambda calculus for quantum computation," *SIAM Journal on Computing*, vol. 33, pp. 1109–1135, 2004.

[8] A. Díaz-Caro, "Du typage vectoriel," Ph.D. dissertation, Université de Grenoble, France, Sep. 2011.

[9] P. Arrighi, A. Díaz-Caro, and B. Valiron, "The vectorial lambda-calculus," *Information and Computation*, vol. 254, no. 1, pp. 105–139, 2017.

[10] P. Arrighi and A. Díaz-Caro, "A System F accounting for scalars," *Logical Methods in Computer Science*, vol. 8, 2012.

[11] S. C. Kleene, "On the interpretation of intuitionistic number theory," *Journal of Symbolic Logic*, vol. 10, pp. 109–124, 1945.

[12] C. Bdescu and P. Panangaden, "Quantum alternation: Prospects and problems," in *Proceedings of QPL-2015*, ser. Electronic Proceedings in Theoretical Computer Science, C. Heunen, P. Selinger, and J. Vicary, Eds., vol. 195, 2015, pp. 33–42.

[13] A. Díaz-Caro and B. Petit, "Linearity in the non-deterministic call-by-value setting," in *Proceedings of WoLLIC 2012*, ser. LNCS, L. Ong and R. de Queiroz, Eds., vol. 7456. Buenos Aires, Argentina: Springer, 2012, pp. 216–231.

[14] A. Díaz-Caro and G. Dowek, "Typing quantum superpositions and measurement," in *Theory and Practice of Natural Computing (TPNC 2017)*, ser. Lecture Notes in Computer Science, C. Martín-Vide, R. Neruda, and M. A. Vega-Rodríguez, Eds., vol. 10687. Prague, Czech Republic: Springer, Cham, 2017, pp. 281–293.

[15] A. Díaz-Caro and O. Malherbe, "A concrete categorical semantics for lambda-s," in *13th Workshop on Logical and Semantic Frameworks with Applications (LSFA 2018)*, 2018, pp. 143–172, to appear in ENTCS. Available at arXiv:1806.09236.

[16] A. Sabry, B. Valiron, and J. K. Vizzotto, "From symmetric pattern-matching to quantum control," in *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018,*, ser. LNCS, C. Baier and U. D. Lago, Eds., vol. 10803. Thessalonikis, Greece: Springer, 2018, pp. 348–364.

[17] L. Vaux, "The algebraic lambda calculus," *Mathematical Structures in Computer Science*, vol. 19, pp. 1029–1059, 2009.

[18] T. Ehrhard and L. Regnier, "The differential lambda-calculus," *Theoretical Computer Science*, vol. 309, no. 1, pp. 1–41, 2003.

[19] A. Assaf, A. Díaz-Caro, S. Perdrix, C. Tasson, and B. Valiron, "Call-by-value, call-by-name and the vectorial behaviour of the algebraic λ-calculus," *Logical Methods in Computer Science*, vol. 10, 2014.

[20] P. Selinger, "Towards a quantum programming language," *Mathematical Structures in Computer Science*, vol. 14, no. 4, pp. 527–586, 2004.

[21] P. Selinger and B. Valiron, "On a fully abstract model for a quantum linear functional language," in *Proceedings of the Fourth International Workshop on Quantum Programming Languages (QPL'06)*, ser. Electronic Notes in Theoretical Computer Science, P. Selinger, Ed., vol. 210, Oxford, UK., July 2008, pp. 123–137.

[22] O. Malherbe, P. Scott, and P. Selinger, "Presheaf models of quantum computation: An outline," in *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky - Essays Dedicated to Samson Abramsky on the Occasion of His 60th Birthday*, ser. Lecture Notes in Computer Science, B. Coecke, L. Ong, and P. Panangaden, Eds. Springer, 2013, vol. 7860, pp. 178–194.

[23] M. Pagani, P. Selinger, and B. Valiron, "Applying quantitative semantics to higher-order quantum computing," *ACM SIGPLAN Notices (POPL'14)*, vol. 49, no. 1, pp. 647–658, 2014.

[24] F. Rios and P. Selinger, "A categorical model for a quantum circuit description language," in *Proceedings of the 14th International Conference on Quantum Physics and Logic, QPL 2017*, ser. EPTCS, B. Coecke and A. Kissinger, Eds., vol. 266, 2017, pp. 164–178.

[25] B. Lindenhovius, M. Mislove, and V. Zamdzhiev, "Enriching a linear/non-linear lambda calculus: A programming language for string diagrams," in *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2018)*. ACM, 2018, pp. 659–668.

[26] N. J. Ross, "Algebraic and logical methods in quantum computation," Ph.D. dissertation, Dalhousie University, 2015.

[27] P. Selinger and B. Valiron, "A lambda calculus for quantum computation with classical control," *Mathematical Structures in Computer Science*, vol. 16, no. 3, pp. 527–552, 2006.

[28] P. Z. Ingerman, "Thunks: A way of compiling procedure statements with some comments on procedure declarations," *Communication of the ACM*, vol. 4, no. 1, pp. 55–58, 1961.

[29] T. Altenkirch and J. Grattage, "A functional quantum programming language," in *Proceedings of LICS 2005*. Chicago, USA: IEEE, 2005, pp. 249–258.

APPENDIX

A. Proofs related to Section III

Lemma A.1 (Simplifying equalities)**.** *Let scalars $\alpha_1, \alpha_2 \in \mathbb{C}$, pure terms t_1, t_2 and term distributions $\vec{s}_1$, $\vec{s}_2$ such that $\alpha_1 \cdot t_1 + \vec{s}_1 \equiv \alpha_2 \cdot t_2 + \vec{s}_2$.*

1) *If $t_1 = t_2 = t$ and $\alpha_1 = \alpha_2$, then: $\vec{s}_1 \equiv \vec{s}_2$ or $\vec{s}_1 \equiv \vec{s}_2 + 0 \cdot t$ or $\vec{s}_2 \equiv \vec{s}_1 + 0 \cdot t$.*
2) *If $t_1 = t_2 = t$ but $\alpha_1 \neq \alpha_2$, then: $\vec{s}_1 \equiv \vec{s}_2 + (\alpha_2 - \alpha_1) \cdot t$ or $\vec{s}_2 \equiv \vec{s}_1 + (\alpha_1 - \alpha_2) \cdot t$.*
3) *If $t_1 \neq t_2$, then: $\vec{s}_1 \equiv \vec{s}_3 + \alpha_2 \cdot t_2$ and $\vec{s}_2 \equiv \vec{s}_3 + \alpha_1 \cdot t_1$ for some distribution $\vec{s}_3$.*

(All the above disjunctions are inclusive). □

Lemma III.2 (Weak diamond)**.** *if $\vec{t} \succ \vec{t}'_1$ and $\vec{t} \succ \vec{t}'_2$, then one of the following holds: either $\vec{t}'_1 = \vec{t}'_2$; either $\vec{t}'_1 \succ \vec{t}'_2$ or $\vec{t}'_2 \succ \vec{t}'_1$; either $\vec{t}'_1 \succ \vec{t''}$ and $\vec{t}'_2 \succ \vec{t''}$ for some $\vec{t''}$.*

Proof of Lemma III.2. Since $\vec{t} \succ \vec{t}'_1$ and $\vec{t} \succ \vec{t}'_2$, there are decompositions

$$\begin{array}{rclcrcll} \vec{t} & = & \alpha_1 \cdot s_1 + \vec{r}_1 & \qquad & \vec{t}'_1 & = & \alpha_1 \cdot \vec{s'}_1 + \vec{r}_1 & \text{where } s_1 \triangleright \vec{s'}_1 \\ \vec{t} & = & \alpha_2 \cdot s_2 + \vec{r}_2 & & \vec{t}'_2 & = & \alpha_2 \cdot \vec{s'}_2 + \vec{r}_2 & \text{where } s_2 \triangleright \vec{s'}_2 \end{array}$$

We distinguish three cases:

- Case where $s_1 = s_2 = s$ and $\alpha_1 = \alpha_2 = \alpha$. In this case, we have $\vec{s'}_1 = \vec{s'}_2 = \vec{s'}$ since atomic evaluation is deterministic. And by Lemma A.1 (1), we deduce that:
  - Either $\vec{r}_1 = \vec{r}_2$, so that: $\vec{t}'_1 = \alpha \cdot \vec{s'} + \vec{r}_1 = \alpha \cdot \vec{s'} + \vec{r}_2 = \vec{t}'_2$.
  - Either $\vec{r}_1 = \vec{r}_2 + 0 \cdot s$, so that:
$$\begin{array}{rcl} \vec{t}'_1 = \alpha \cdot \vec{s'} + \vec{r}_1 & = & \alpha \cdot \vec{s'} + \vec{r}_2 + 0 \cdot s \\ & \succ & \alpha \cdot \vec{s'} + \vec{r}_2 + 0 \cdot \vec{s'} = (\alpha + 0) \cdot \vec{s'} + \vec{r}_2 = \vec{t}'_2 \,. \end{array}$$
  - Either $\vec{r}_2 = \vec{r}_1 + 0 \cdot s$, so that:
$$\begin{array}{rcl} \vec{t}'_2 = \alpha \cdot \vec{s'} + \vec{r}_2 & = & \alpha \cdot \vec{s'} + \vec{r}_1 + 0 \cdot s \\ & \succ & \alpha \cdot \vec{s'} + \vec{r}_1 + 0 \cdot \vec{s'} = (\alpha + 0) \cdot \vec{s'} + \vec{r}_1 = \vec{t}'_1 \,. \end{array}$$
- Case where $s_1 = s_2 = s$, but $\alpha_1 \neq \alpha_2$. In this case, we have $\vec{s'}_1 = \vec{s'}_2 = \vec{s'}$ since atomic evaluation is deterministic. And by Lemma A.1 (2), we deduce that:
  - Either $\vec{r}_1 = \vec{r}_2 + (\alpha_2 - \alpha_1) \cdot s$, so that:
$$\begin{array}{rcl} \vec{t}'_1 = \alpha_1 \cdot \vec{s'} + \vec{r}_1 & = & \alpha_1 \cdot \vec{s'} + \vec{r}_2 + (\alpha_2 - \alpha_1) \cdot s \\ & \succ & \alpha_1 \cdot \vec{s'} + \vec{r}_2 + (\alpha_2 - \alpha_1) \cdot \vec{s'} = \alpha_2 \cdot \vec{s'} + \vec{r}_2 = \vec{t}'_2 \,. \end{array}$$
  - Either $\vec{r}_2 = \vec{r}_1 + (\alpha_1 - \alpha_2) \cdot s$, so that:
$$\begin{array}{rcl} \vec{t}'_2 = \alpha_2 \cdot \vec{s'} + \vec{r}_2 & = & \alpha_2 \cdot \vec{s'} + \vec{r}_1 + (\alpha_1 - \alpha_2) \cdot s \\ & \succ & \alpha_2 \cdot \vec{s'} + \vec{r}_1 + (\alpha_1 - \alpha_2) \cdot \vec{s'} = \alpha_1 \cdot \vec{s'} + \vec{r}_1 = \vec{t}'_1 \,. \end{array}$$
- Case where $s_1 \neq s_2$. In this case, we know by Lemma A.1 (3) that $\vec{r}_1 = \vec{r}_3 + \alpha_2 \cdot s_2$ and $\vec{r}_2 = \vec{r}_3 + \alpha_1 \cdot s_1$ for some $\vec{r}_3$. Writing $\vec{t''} = \alpha_1 \cdot \vec{s'}_1 + \alpha_2 \cdot \vec{s'}_2 + \vec{r}_3$, we conclude that
$$\begin{array}{rcl} \vec{t}'_1 = \alpha_1 \cdot \vec{s'}_1 + \vec{r_1} = \alpha_1 \cdot \vec{s'}_1 + \alpha_2 \cdot s_2 + \vec{r_3} & \succ & \alpha_1 \cdot \vec{s'}_1 + \alpha_2 \cdot \vec{s'}_2 + \vec{r_3} = \vec{t''} \\ \vec{t}'_2 = \alpha_2 \cdot \vec{s'}_2 + \vec{r_2} = \alpha_1 \cdot s_1 + \alpha_2 \cdot \vec{s'}_2 + \vec{r_3} & \succ & \alpha_1 \cdot \vec{s'}_1 + \alpha_2 \cdot \vec{s'}_2 + \vec{r_3} = \vec{t''} \end{array}$$ □

B. Proofs related to Section IV

Proposition A.2. *For all value distributions $\vec{v}_1, \vec{v}_2, \vec{w}_1, \vec{w}_2$, we have:*

$$\begin{aligned} \langle \mathtt{inl}(\vec{v}_1) \mid \mathtt{inl}(\vec{v}_2) \rangle &= \langle \vec{v}_1 \mid \vec{v}_2 \rangle \\ \langle \mathtt{inr}(\vec{w}_1) \mid \mathtt{inr}(\vec{w}_2) \rangle &= \langle \vec{w}_1 \mid \vec{w}_2 \rangle \\ \langle (\vec{v}_1, \vec{w}_1) \mid (\vec{v}_2, \vec{w}_2) \rangle &= \langle \vec{v}_1 \mid \vec{v}_2 \rangle \langle \vec{w}_1 \mid \vec{w}_2 \rangle \\ \langle \mathtt{inl}(\vec{v}_1) \mid \mathtt{inr}(\vec{w}_2) \rangle &= 0 \\ \langle \mathtt{inl}(\vec{v}_1) \mid (\vec{v}_2, \vec{w}_2) \rangle &= 0 \\ \langle \mathtt{inr}(\vec{w}_1) \mid (\vec{v}_2, \vec{w}_2) \rangle &= 0 \end{aligned}$$ □

Proof. Let us write $\vec{v}_1 = \sum_{i_1=1}^{n_1} \alpha_{1,i_1} \cdot v_{1,i_1}$, $\vec{v}_2 = \sum_{i_2=1}^{n_2} \alpha_{2,i_2} \cdot v_{2,i_1}$, $\vec{w}_1 = \sum_{j_1=1}^{m_1} \beta_{1,j_1} \cdot w_{1,j_1}$ and $\vec{w}_2 = \sum_{j_2=1}^{m_2} \beta_{2,j_2} \cdot w_{2,j_1}$ (all in canonical form). Writing $\delta_{v,v'} = 1$ when $v = v'$ and $\delta_{v,v'} = 0$ when $v \neq v'$ (Kronecker symbol), we observe that:

$$\begin{array}{rcl}
\langle \mathtt{inl}(v)_1 \mid \mathtt{inl}(v)_2 \rangle & = & \big\langle \sum_{i_1=1}^{n_1} \alpha_{1,i_1} \cdot \mathtt{inl}(v_{1,i_1}) \;\big|\; \sum_{i_2=1}^{n_2} \alpha_{2,i_2} \cdot \mathtt{inl}(v_{2,i_2}) \big\rangle \\
& = & \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \overline{\alpha_{1,i_1}}\, \alpha_{2,i_2} \langle \mathtt{inl}(v_{1,i_1}) \mid \mathtt{inl}(v_{2,i_2}) \rangle \\
& = & \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \overline{\alpha_{1,i_1}}\, \alpha_{2,i_2}\, \delta_{\mathtt{inl}(v_{1,i_1}),\mathtt{inl}(v_{2,i_2})} \\
& = & \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \overline{\alpha_{1,i_1}}\, \alpha_{2,i_2}\, \delta_{v_{1,i_1},v_{2,i_2}} \\
& = & \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \overline{\alpha_{1,i_1}}\, \alpha_{2,i_2} \langle v_{1,i_1} \mid v_{2,i_2} \rangle \;=\; \langle \vec{v}_1 \mid \vec{v}_2 \rangle
\end{array}$$

$$\begin{array}{rcl}
\langle \mathtt{inl}(v)_1 \mid \mathtt{inr}(w)_2 \rangle & = & \big\langle \sum_{i_1=1}^{n_1} \alpha_{1,i_1} \cdot \mathtt{inr}(v_{1,i_1}) \;\big|\; \sum_{j_2=1}^{m_2} \beta_{2,j_2} \cdot \mathtt{inl}(w_{2,j_2}) \big\rangle \\
& = & \sum_{i_1=1}^{n_1} \sum_{j_2=1}^{m_2} \overline{\alpha_{1,i_1}}\, \beta_{2,j_2} \langle \mathtt{inl}(v_{1,i_1}) \mid \mathtt{inr}(w_{2,j_2}) \rangle \\
& = & \sum_{i_1=1}^{n_1} \sum_{j_2=1}^{m_2} \overline{\alpha_{1,i_1}}\, \beta_{2,j_2}\, \delta_{\mathtt{inl}(v_{1,i_1}),\mathtt{inr}(w_{2,j_2})} \\
& = & \sum_{i_1=1}^{n_1} \sum_{j_2=1}^{m_2} \overline{\alpha_{1,i_1}}\, \beta_{2,j_2} \times 0 \;=\; 0
\end{array}$$

$$\begin{array}{rcl}
\langle (\vec{v}_1, \vec{w}_1) \mid (\vec{v}_2, \vec{w}_2) \rangle & = & \big\langle \sum_{i_1=1}^{n_1} \sum_{j_1=1}^{m_1} \alpha_{1,i_1} \beta_{1,j_1} \cdot (v_{1,i_1}, w_{1,j_1}) \;\big|\; \sum_{i_2=1}^{n_2} \sum_{j_2=1}^{m_2} \alpha_{2,i_2} \beta_{2,j_2} \cdot (v_{2,i_2}, w_{2,j_2}) \big\rangle \\
& = & \sum_{i_1=1}^{n_1} \sum_{j_1=1}^{m_1} \sum_{i_2=1}^{n_2} \sum_{j_2=1}^{m_2} \overline{\alpha_{1,i_1} \beta_{1,j_1}} \alpha_{2,i_2} \beta_{2,j_2} \langle (v_{1,i_1}, w_{1,j_1}) \mid (v_{2,i_2}, w_{2,j_2}) \rangle \\
& = & \sum_{i_1=1}^{n_1} \sum_{j_1=1}^{m_1} \sum_{i_2=1}^{n_2} \sum_{j_2=1}^{m_2} \overline{\alpha_{1,i_1} \beta_{1,j_1}} \alpha_{2,i_2} \beta_{2,j_2} \delta_{(v_{1,i_1},w_{1,j_1}),(v_{2,i_2},w_{2,j_2})} \\
& = & \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \sum_{j_1=1}^{m_1} \sum_{j_2=1}^{m_2} \overline{\alpha_{1,i_1}}\, \alpha_{2,i_2}\, \overline{\beta_{1,j_1}}\, \beta_{2,j_2}\, \delta_{v_{1,i_1},v_{2,i_2}}\, \delta_{w_{1,j_1},w_{2,j_2}} \\
& = & \big(\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \overline{\alpha_{1,i_1}}\, \alpha_{2,i_2}\, \delta_{v_{1,i_1},v_{2,i_2}}\big) \big(\sum_{j_1=1}^{m_1} \sum_{j_2=1}^{m_2} \overline{\beta_{1,j_1}}\, \beta_{2,j_2}\, \delta_{w_{1,j_1},w_{2,j_2}}\big) \\
& = & \big(\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \overline{\alpha_{1,i_1}}\, \alpha_{2,i_2} \langle v_{1,i_1} \mid v_{2,i_2} \rangle\big) \big(\sum_{j_1=1}^{m_1} \sum_{j_2=1}^{m_2} \overline{\beta_{1,j_1}}\, \beta_{2,j_2} \langle w_{1,j_1} \mid w_{2,j_2} \rangle\big) \\
& = & \langle \vec{v}_1 \mid \vec{v}_2 \rangle \langle \vec{w}_1 \mid \vec{w}_2 \rangle
\end{array}$$

The other equalities are proved similarly. □

Lemma IV.3. *For all types A, we have $[\![A]\!] = \{\Vdash A\} \cap \vec{\mathrm{V}}$.*

Proof. The inclusion $[\![A]\!] \subseteq \{\Vdash A\} \cap \vec{\mathrm{V}}$ is clear from the definition of $\{\Vdash A\}$. Conversely, suppose that $\vec{v} \in \{\Vdash A\} \cap \vec{\mathrm{V}}$. From the definition of the set $\{\Vdash A\}$, we know that $\vec{v} \twoheadrightarrow \vec{v'}$ for some $\vec{v'} \in [\![A]\!]$. But since $\vec{v}$ is a normal form, we deduce that $\vec{v} = \vec{v'} \in [\![A]\!]$. □

Lemma IV.5. *Given any two types A and B:*

1) *$A \leq B$ is valid if and only if $\{\Vdash A\} \subseteq \{\Vdash B\}$.*
2) *$A \simeq B$ is valid if and only if $\{\Vdash A\} = \{\Vdash B\}$.*

Proof. The direct implications are obvious from the definition of $\{\Vdash A\}$, and the converse implications immediately follow from Lemma IV.3. □

Proposition IV.11. *Given a closed λ-abstraction $\lambda x\,.\,\vec{t}$, we have $\lambda x\,.\,\vec{t} \in [\![\sharp\mathbb{B} \to \sharp\mathbb{B}]\!]$ if and only if there are two value distributions $\vec{v}_1, \vec{v}_2 \in [\![\sharp\mathbb{B}]\!]$ such that*

$$\vec{t}[x := \mathtt{tt}] \twoheadrightarrow \vec{v}_1, \quad \vec{t}[x := \mathtt{ff}] \twoheadrightarrow \vec{v}_2, \quad \textit{and } \langle \vec{v}_1 \mid \vec{v}_2 \rangle = 0\,.$$

Proof. The condition is necessary. Suppose that $\lambda x\,.\,\vec{t} \in [\![\sharp\mathbb{B} \to \sharp\mathbb{B}]\!]$. Since $\mathtt{tt}, \mathtt{ff} \in [\![\sharp\mathbb{B}]\!]$, there are $\vec{v}_1, \vec{v}_2 \in [\![\sharp\mathbb{B}]\!]$ such that $\vec{t}[x := \mathtt{tt}] \twoheadrightarrow \vec{v}_1$ and $\vec{t}[x := \mathtt{ff}] \twoheadrightarrow \vec{v}_2$. It remains to prove that $\langle \vec{v}_1 \mid \vec{v}_2 \rangle = 0$. For that, consider $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$. By linearity, we observe that

$$\vec{t}\langle x := \alpha \cdot \mathtt{tt} + \beta \cdot \mathtt{ff} \rangle \;=\; \alpha \cdot \vec{t}[x := \mathtt{tt}] + \beta \cdot \vec{t}[x := \mathtt{ff}] \;\twoheadrightarrow\; \alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2\,.$$

But since $\alpha \cdot \mathtt{tt} + \beta \cdot \mathtt{ff} \in [\![\sharp\mathbb{B}]\!]$, we must have $\alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2 \in [\![\sharp\mathbb{B}]\!]$ too, and in particular $\|\alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2\| = 1$. From this, we get

$$\begin{array}{rcl}
1 \;=\; \|\alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2\|^2 & = & \langle \alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2 \mid \alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2 \rangle \\
& = & |\alpha|^2 \langle \vec{v}_1 \mid \vec{v}_1 \rangle + \bar{\alpha}\beta \langle \vec{v}_1 \mid \vec{v}_2 \rangle + \alpha\bar{\beta} \langle \vec{v}_2 \mid \vec{v}_1 \rangle + |\beta|^2 \langle \vec{v}_2 \mid \vec{v}_2 \rangle \\
& = & |\alpha|^2 + |\beta|^2 + \bar{\alpha}\beta \langle \vec{v}_1 \mid \vec{v}_2 \rangle + \overline{\bar{\alpha}\beta \langle \vec{v}_1 \mid \vec{v}_2 \rangle} \;=\; 1 + 2\mathrm{Re}\big(\alpha\bar{\beta} \langle \vec{v}_1 \mid \vec{v}_2 \rangle\big)
\end{array}$$

and thus $\mathrm{Re}(\bar{\alpha}\beta \langle \vec{v}_1 \mid \vec{v}_2 \rangle) = 0$. Taking $\alpha = \beta = \frac{1}{\sqrt{2}}$, we deduce that $\mathrm{Re}(\langle \vec{v}_1 \mid \vec{v}_2 \rangle) = 0$. And taking $\alpha = i\frac{1}{\sqrt{2}}$ and $\beta = \frac{1}{\sqrt{2}}$, we deduce that $\mathrm{Im}(\langle \vec{v}_1 \mid \vec{v}_2 \rangle) = 0$. Therefore: $\langle \vec{v}_1 \mid \vec{v}_2 \rangle = 0$.

The condition is sufficient. Suppose that there are $\vec{v}_1, \vec{v}_2 \in [\![\sharp\mathbb{B}]\!]$ such that $\vec{t}[x := \mathtt{tt}] \twoheadrightarrow \vec{v}_1$, $\vec{t}[x := \mathtt{ff}] \twoheadrightarrow \vec{v}_2$ and $\langle \vec{v}_1 \mid \vec{v}_2 \rangle = 0$. In particular, we have $\vec{v}_1, \vec{v}_2 \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$ and $\|\vec{v}_1\| = \|\vec{v}_2\| = 1$. Now, given any $\vec{v} \in [\![\sharp\mathbb{B}]\!]$, we distinguish three cases:

- Either $\vec{v} = \alpha \cdot \mathtt{tt}$, where $|\alpha| = 1$. In this case, we observe that
$$\vec{t}\langle x := \vec{v} \rangle \;=\; \alpha \cdot \vec{t}[x := \mathtt{tt}] \;\twoheadrightarrow\; \alpha \cdot \vec{v}_1 \;\in\; [\![\sharp\mathbb{B}]\!]\,,$$
since $\alpha \cdot \vec{v}_1 \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$ and $\|\alpha \cdot \vec{v}_1\| = |\alpha|\,\|\vec{v}_1\| = 1$.
- Either $\vec{v} = \beta \cdot \mathtt{ff}$, where $|\beta| = 1$. In this case, we observe that
$$\vec{t}\langle x := \vec{v} \rangle \;=\; \beta \cdot \vec{t}[x := \mathtt{ff}] \;\twoheadrightarrow\; \beta \cdot \vec{v}_2 \;\in\; [\![\sharp\mathbb{B}]\!]\,,$$
since $\beta \cdot \vec{v}_2 \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$ and $\|\beta \cdot \vec{v}_2\| = |\beta|\,\|\vec{v}_2\| = 1$.
- Either $\vec{v} = \alpha \cdot \mathtt{tt} + \beta \cdot \mathtt{ff}$, where $|\alpha|^2 + |\beta|^2 = 1$. In this case, we observe that
$$\vec{t}\langle x := \vec{v} \rangle \;=\; \alpha \cdot \vec{t}[x := \mathtt{tt}] + \beta \cdot \vec{t}[x := \mathtt{ff}] \;\twoheadrightarrow\; \alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2 \;\in\; [\![\sharp\mathbb{B}]\!]\,,$$
since $\alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2 \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$ and
$$\begin{array}{rcl} \|\alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2\|^2 & = & \langle \alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2 \mid \alpha \cdot \vec{v}_1 + \beta \cdot \vec{v}_2 \rangle \\ & = & |\alpha|^2 \langle \vec{v}_1 \mid \vec{v}_1 \rangle + \alpha\bar{\beta}\,\langle \vec{v}_1 \mid \vec{v}_2 \rangle + \bar{\alpha}\beta\,\langle \vec{v}_2 \mid \vec{v}_1 \rangle + |\beta|^2 \langle \vec{v}_2 \mid \vec{v}_2 \rangle \\ & = & |\alpha|^2 \|\vec{v}_1\|^2 + 0 + 0 + |\beta|^2 \|\vec{v}_2\|^2 \;=\; |\alpha|^2 + |\beta|^2 \;=\; 1\,. \end{array}$$

We have thus shown that $\vec{t}\langle x := \vec{v} \rangle \Vdash \sharp\mathbb{B}$ for all $\vec{v} \in [\![\sharp\mathbb{B}]\!]$. Therefore $\lambda x\,.\,\vec{t} \in [\![\sharp\mathbb{B} \to \sharp\mathbb{B}]\!]$. □

Theorem IV.12 (Characterization of the values of type $\sharp\mathbb{B} \to \sharp\mathbb{B}$)**.** *A closed λ-abstraction $\lambda x\,.\,\vec{t}$ is a value of type $\sharp\mathbb{B} \to \sharp\mathbb{B}$ if and only if it represents a unitary operator $F : \mathbb{C}^2 \to \mathbb{C}^2$.*

Proof. The condition is necessary. Suppose that $\lambda x\,.\,\vec{t} \in [\![\sharp\mathbb{B} \to \sharp\mathbb{B}]\!]$. From Prop. IV.11, there are $\vec{v}_1, \vec{v}_2 \in [\![\sharp\mathbb{B}]\!]$ such that $\vec{t}[x := \mathtt{tt}] \twoheadrightarrow \vec{v}_1$, $\vec{t}[x := \mathtt{ff}] \twoheadrightarrow \vec{v}_2$ and $\langle \vec{v}_1 \mid \vec{v}_2 \rangle = 0$. Let $F : \mathbb{C}^2 \to \mathbb{C}^2$ be the operator defined by $F(1,0) = \pi_{\mathbb{B}}(\vec{v}_1)$ and $F(0,1) = \pi_{\mathbb{B}}(\vec{v}_2)$. From the properties of linearity of the calculus, it is clear that the abstraction $\lambda x\,.\,\vec{t}$ represents the operator $F : \mathbb{C}^2 \to \mathbb{C}^2$. Moreover, the operator F is unitary since $\|\pi_{\mathbb{B}}(\vec{v}_1)\|_{\mathbb{C}^2} = \|\pi_{\mathbb{B}}(\vec{v}_2)\|_{\mathbb{C}^2} = 1$ and $\langle \pi_{\mathbb{B}}(\vec{v}_1) \mid \pi_{\mathbb{B}}(\vec{v}_2) \rangle_{\mathbb{C}^2} = 0$.

The condition is sufficient. Let us assume that the abstraction $\lambda x\,.\,\vec{t}$ represents a unitary operator $F : \mathbb{C}^2 \to \mathbb{C}^2$. From this, we deduce that:

- $(\lambda x\,.\,\vec{t})\,\mathtt{tt} \twoheadrightarrow \vec{v}_1$ for some $\vec{v}_1 \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$ such that $\pi_{\mathbb{B}}(\vec{v}_1) = F(\pi_{\mathbb{B}}(\mathtt{tt})) = F(1,0)$;
- $(\lambda x\,.\,\vec{t})\,\mathtt{ff} \twoheadrightarrow \vec{v}_2$ for some $\vec{v}_2 \in \mathrm{span}(\{\mathtt{tt}, \mathtt{ff}\})$ such that $\pi_{\mathbb{B}}(\vec{v}_2) = F(\pi_{\mathbb{B}}(\mathtt{ff})) = F(0,1)$.

Using the property of confluence, we deduce that

- $\vec{t}[x := \mathtt{tt}] \twoheadrightarrow \vec{v}_1 \in [\![\sharp\mathbb{B}]\!]$, since $\|\vec{v}_1\| = \|F(1,0)\|_{\mathbb{C}^2} = 1$;
- $\vec{t}[x := \mathtt{ff}] \twoheadrightarrow \vec{v}_2 \in [\![\sharp\mathbb{B}]\!]$, since $\|\vec{v}_2\| = \|F(0,1)\|_{\mathbb{C}^2} = 1$.

We deduce that $\lambda x\,.\,\vec{t} \in [\![\sharp\mathbb{B} \to \sharp\mathbb{B}]\!]$ by Prop. IV.11, since $\langle \vec{v}_1 \mid \vec{v}_2 \rangle = \langle F(1,0) \mid F(0,1) \rangle_{\mathbb{C}^2} = 0$. □

Corollary IV.13 (Characterization of the values of type $\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$)**.** *A unitary distribution of abstractions $\left(\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{t_i}\right) \in \mathcal{S}_1$ is a value of type $\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$ if and only if it represents a unitary operator $F : \mathbb{C}^2 \to \mathbb{C}^2$.*

Proof. Indeed, given $\left(\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{t_i}\right) \in \mathcal{S}_1$, we have

$$\begin{array}{ll} & \left(\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{t_i}\right) \in [\![\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}]\!] \\ \text{iff} & \lambda x\,.\left(\sum_{i=1}^n \alpha_i \cdot \vec{t_i}\right) \in [\![\sharp\mathbb{B} \to \sharp\mathbb{B}]\!] \\ \text{iff} & \lambda x\,.\left(\sum_{i=1}^n \alpha_i \cdot \vec{t_i}\right) \text{ represents a unitary operator } F : \mathbb{C}^2 \to \mathbb{C}^2 \\ \text{iff} & \left(\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{t_i}\right) \text{ represents a unitary operator } F : \mathbb{C}^2 \to \mathbb{C}^2 \end{array}$$

since both functions $\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{t_i}$ and $\lambda x\,.\left(\sum_{i=1}^n \alpha_i \cdot \vec{t_i}\right)$ are extensionally equivalent. □

Lemma A.3. *For all term distributions $\vec{t}$, $\vec{t'}$, $\vec{s}$, $\vec{s_1}$, $\vec{s_2}$ and for all value distributions $\vec{v}$ and $\vec{w}$:*

1) $(\lambda x\,.\,\vec{t})\,\vec{v} \;\twoheadrightarrow\; \vec{t}\langle x := \vec{v} \rangle$
2) $\mathtt{let}\ (x,y) = (\vec{v}, \vec{w})\ \mathtt{in}\ \vec{s} \;\twoheadrightarrow\; \vec{s}\langle x := \vec{v} \rangle\langle y := \vec{w} \rangle$ *(if $y \notin FV(\vec{v})$)*
3) $\mathtt{match}\ \mathtt{inl}(\vec{v})\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\} \;\twoheadrightarrow\; \vec{s}_1\langle x_1 := \vec{v} \rangle$
4) $\mathtt{match}\ \mathtt{inr}(\vec{v})\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\} \;\twoheadrightarrow\; \vec{s}_2\langle x_2 := \vec{v} \rangle$
5) *If* $\vec{t} \twoheadrightarrow \vec{t'}$, *then* $\vec{s}\,\vec{t} \twoheadrightarrow \vec{s}\,\vec{t'}$
6) *If* $\vec{t} \twoheadrightarrow \vec{t'}$, *then* $\vec{t}\,\vec{v} \twoheadrightarrow \vec{t'}\,\vec{v}$
7) *If* $\vec{t} \twoheadrightarrow \vec{t'}$, *then* $\vec{t}; \vec{s} \twoheadrightarrow \vec{t'}; \vec{s}$
8) *If* $\vec{t} \twoheadrightarrow \vec{t'}$, *then* $\mathtt{let}\ (x_1, x_2) = \vec{t}\ \mathtt{in}\ \vec{s} \;\twoheadrightarrow\; \mathtt{let}\ (x_1, x_2) = \vec{t'}\ \mathtt{in}\ \vec{s}$

9) *If* $\vec{t} \twoheadrightarrow \vec{t}'$, *then* $\texttt{match}\, \vec{t}\, \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\} \twoheadrightarrow \texttt{match}\, \vec{t}'\, \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\}$

10) *If* $\vec{t} \twoheadrightarrow \vec{t}'$, *then* $\vec{t}\langle x := \vec{v}\rangle \twoheadrightarrow \vec{t}'\langle x := \vec{v}\rangle$.

Proof. (1) Assume that $\vec{v} = \sum_{i=1}^n \alpha_i \cdot v_i$. Then we observe that

$$(\lambda x\,.\,\vec{t})\,\vec{v} \;=\; \textstyle\sum_{i=1}^n \alpha_i \cdot (\lambda x\,.\,\vec{t})\,v_i \;\twoheadrightarrow\; \sum_{i=1}^n \alpha_i \cdot \vec{t}[x := v_i] \;=\; \vec{t}\langle x := \vec{v}\rangle\,.$$

(2) Assume that $\vec{v} = \sum_{i=1}^n \alpha_i \cdot v_i$ and $\vec{w} = \sum_{j=1}^m \beta_j \cdot w_j$. Then we observe that

$$\begin{array}{rcl}
\texttt{let}\ (x,y) = (\vec{v},\vec{w})\ \texttt{in}\ \vec{s} & = & \texttt{let}\ (x,y) = \big(\sum_{i=1}^n \sum_{j=1}^m \alpha_j\beta_j \cdot (v_i, w_j)\big)\ \texttt{in}\ \vec{s} \\
& = & \sum_{i=1}^n \sum_{j=1}^m \alpha_i\beta_j \cdot \texttt{let}\ (x,y) = (v_i,w_j)\ \texttt{in}\ \vec{s} \\
& \twoheadrightarrow & \sum_{i=1}^n \sum_{j=1}^m \alpha_i\beta_j \cdot \vec{s}[x := v_i, y := w_j] \\
& = & \sum_{i=1}^n \sum_{j=1}^m \alpha_i\beta_j \cdot \vec{s}[x := v_i][y := w_j] \quad (\text{since } y \notin \mathit{FV}(\vec{v})) \\
& = & \vec{s}\langle x := \vec{v}\rangle\langle y := \vec{w}\rangle
\end{array}$$

Items (3) and (4) are proved similarly as item (2). Then, items (5), (6), (7), (8), (9) and (10) are all proved following the same pattern, first treating the case where $\vec{t} \succ \vec{t}'$ (one step), and then deducing the general case by induction on the number of evaluation steps. Let us prove for instance (5), first assuming that $\vec{t} \succ \vec{t}'$ (one step). This means that there exist a scalar $\alpha \in \mathbb{R}$, a pure term t_0 and term distributions $\vec{t}_0'$ and $\vec{r}$ such that

$$\vec{t} \;=\; \alpha \cdot t_0 + \vec{r}, \qquad \vec{t}' \;=\; \alpha \cdot \vec{t}_0' + \vec{r} \qquad \text{and} \qquad t_0 \triangleright \vec{t}_0'\,.$$

So that for all term distributions $\vec{s} = \sum_{i=1}^n \beta_i \cdot s_i$, we have:

$$\begin{array}{rcl}
\vec{s}\,\vec{t} & = & \big(\sum_{i=1}^n \beta_i \cdot s_i\big)\,(\alpha \cdot t_0 + \vec{r}) \;=\; \sum_{i=1}^n (\alpha\beta_i \cdot s_i\, t_0 + \beta_i \cdot s_i\, \vec{r}) \\
& \twoheadrightarrow & \sum_{i=1}^n (\alpha\beta_i \cdot s_i\, \vec{t}_0' + \beta_i \cdot s_i\, \vec{r}) \;=\; \big(\sum_{i=1}^n \beta_i \cdot s_i\big)\,(\alpha \cdot \vec{t}_0' + \vec{r}) \;=\; \vec{s}\,\vec{t}'
\end{array}$$

observing that $s_i\, t_0 \triangleright s_i\, \vec{t}_0'$, hence $\alpha\beta_i \cdot s_i\, t_0 + \beta_i \cdot s_i\, \vec{r} \;\succ\; \alpha\beta_i \cdot s_i\, \vec{t}_0' + \beta_i \cdot s_i\, \vec{r}$ for all $i = 1..n$. Hence we proved that $\vec{t} \succ \vec{t}'$ implies $\vec{s}\,\vec{t} \twoheadrightarrow \vec{s}\,\vec{t}'$. By a straightforward induction on the number of evaluation steps, we deduce that $\vec{t} \twoheadrightarrow \vec{t}'$ implies $\vec{s}\,\vec{t} \twoheadrightarrow \vec{s}\,\vec{t}'$. □

Lemma A.4 (Application of realizers)**.** *If* $\vec{s} \Vdash A \Rightarrow B$ *and* $\vec{t} \Vdash A$, *then* $\vec{s}\,\vec{t} \Vdash B$ □

Proof. Since $\vec{t} \Vdash A$, we have $\vec{t} \twoheadrightarrow \vec{v}$ for some vector $\vec{v} \in [\![A]\!]$. And since $\vec{s} \Vdash A \Rightarrow B$, we have $\vec{s} \twoheadrightarrow \sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{s}_i$ for some unitary distribution of abstractions $\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{s}_i \in [\![A \Rightarrow B]\!]$. Therefore, we get

$$\vec{s}\,\vec{t} \;\twoheadrightarrow\; \vec{s}\,\vec{v} \;\twoheadrightarrow\; \big(\textstyle\sum_{i=1}^n \alpha_i \cdot \lambda x\,.\,\vec{s}_i\big)\,\vec{v} \;=\; \sum_{i=1}^n \alpha_i \cdot (\lambda x\,.\,\vec{s}_i)\,\vec{v} \;\twoheadrightarrow\; \sum_{i=1}^n \alpha_i \cdot \vec{s}_i\langle x := \vec{v}\rangle \;\in\; [\![B]\!]$$

from Lemma A.3 (5), (6), (1) and from the definition of $[\![A \Rightarrow B]\!]$. □

C. Proofs related to Section V

Lemma A.5. *Given a type* A, *two vectors* $\vec{u}_1, \vec{u}_2 \in [\![\sharp A]\!]$ *and a scalar* $\alpha \in \mathbb{C}$, *there exists a vector* $\vec{u}_0 \in [\![\sharp A]\!]$ *and a scalar* $\lambda \in \mathbb{C}$ *such that* $\vec{u}_1 + \alpha \cdot \vec{u}_2 = \lambda \cdot \vec{u}_0$.

Proof. Let $\lambda := \|\vec{u}_1 + \alpha \cdot \vec{u}_2\|$. When $\lambda \neq 0$, we take $\vec{u}_0 := \frac{1}{\lambda} \cdot (\vec{u}_1 + \alpha \cdot \vec{u}_2) \in [\![\sharp A]\!]$, and we are done. Let us now consider the (subtle) case where $\lambda = 0$. In this case, we first observe that $\alpha \neq 0$, since $\alpha = 0$ would imply that $\|\vec{u}_1 + \alpha \cdot \vec{u}_2\| = \|\vec{u}_1\| = 0$, which would be absurd, since $\|\vec{u}_1\| = 1$. Moreover, since $\lambda = \|\vec{u}_1 + \alpha \cdot \vec{u}_2\| = 0$, we observe that all the coefficients of the distribution $\vec{u}_1 + \alpha \cdot \vec{u}_2$ are zeros (when written in canonical form), which implies that

$$\vec{u}_1 + \alpha \cdot \vec{u}_2 \;=\; 0 \cdot (\vec{u}_1 + \alpha \cdot \vec{u}_2) \;=\; 0 \cdot \vec{u}_1 + 0 \cdot \vec{u}_2\,.$$

Using the triangular inequality, we also observe that

$$0 \;<\; 2|\alpha| \;=\; \|2\alpha \cdot \vec{u}_2\| \;\leq\; \|\vec{u}_1 + \alpha \cdot \vec{u}_2\| + \|\vec{u}_1 + (-\alpha) \cdot \vec{u}_2\| \;=\; \|\vec{u}_1 + (-\alpha) \cdot \vec{u}_2\|\,,$$

hence $\lambda' := \|\vec{u}_1 + (-\alpha) \cdot \vec{u}_2\| \neq 0$. Taking $u_0 := \frac{1}{\lambda'} \cdot (\vec{u}_1 + (-\alpha) \cdot \vec{u}_2) \in [\![\sharp A]\!]$, we easily see that

$$\vec{u}_1 + \alpha \cdot \vec{u}_2 \;=\; 0 \cdot \vec{u}_1 + 0 \cdot \vec{u}_2 \;=\; 0 \cdot \big(\tfrac{1}{\lambda'} \cdot (\vec{u}_1 + (-\alpha) \cdot \vec{u}_2)\big) \;=\; \lambda \cdot \vec{u}_0\,. \qquad \square$$

Proposition A.6 (Polarisation identity)**.** *For all value distributions* $\vec{v}$ *and* $\vec{w}$, *we have:*

$$\begin{array}{rl}
\langle \vec{v} \mid \vec{w}\rangle = \frac{1}{4}(& \|\vec{v} + \vec{w}\|^2 - \|\vec{v} + (-1) \cdot \vec{w}\|^2 \\
& - i\|\vec{v} + i \cdot \vec{w}\|^2 + i\|\vec{v} + (-i) \cdot \vec{w}\|^2)\,.
\end{array}$$

□

Lemma A.7. *Given a valid typing judgment of the form* $\Delta, x : \sharp A \vdash \vec{s} : C$, *a substitution* $\sigma \in [\![\Delta]\!]$, *and value distributions* $\vec{u}_1, \vec{u}_2 \in [\![\sharp A]\!]$, *there are value distributions* $\vec{w}_1, \vec{w}_2 \in [\![C]\!]$ *such that*

$$\vec{s}\langle \sigma, x := \vec{u}_1\rangle \twoheadrightarrow \vec{w}_1, \qquad \vec{s}\langle \sigma, x := \vec{u}_2\rangle \twoheadrightarrow \vec{w}_2 \qquad \textit{and} \qquad \langle \vec{w}_1 \mid \vec{w}_2\rangle = \langle \vec{u}_1 \mid \vec{u}_2\rangle .$$

Proof. From the validity of the judgment $\Delta, x : \sharp A \vdash \vec{s} : C$, we know that there are $\vec{w}_1, \vec{w}_2 \in [\![C]\!]$ such that $\vec{s}\langle \sigma, x := \vec{u}_1\rangle \twoheadrightarrow \vec{w}_1$ and $\vec{s}\langle \sigma, x := \vec{u}_2\rangle \twoheadrightarrow \vec{w}_2$. In particular, we have $\|\vec{w}_1\| = \|\vec{w}_2\| = 1$. Now applying Lemma A.5 four times, we know that there are vectors $\vec{u}_{0,1}, \vec{u}_{0,2}, \vec{u}_{0,3}, \vec{u}_{0,4} \in [\![\sharp A]\!]$ and scalars $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{C}$ such that

$$\begin{array}{rclcrcl} \vec{u}_1 + \vec{u}_2 & = & \lambda_1 \cdot \vec{u}_{0,1} & \qquad & \vec{u}_1 + i \cdot \vec{u}_2 & = & \lambda_3 \cdot \vec{u}_{0,3} \\ \vec{u}_1 + (-1) \cdot \vec{u}_2 & = & \lambda_2 \cdot \vec{u}_{0,2} & & \vec{u}_1 + (-i) \cdot \vec{u}_2 & = & \lambda_4 \cdot \vec{u}_{0,4} \end{array}$$

From the validity of the judgment $\Delta, x : \sharp A \vdash \vec{s} : C$, we also know that there are value distributions $\vec{w}_{0,1}, \vec{w}_{0,2}, \vec{w}_{0,3}, \vec{w}_{0,4} \in [\![C]\!]$ such that $\vec{s}\langle \sigma, x := \vec{u}_{0,j}\rangle \twoheadrightarrow \vec{w}_{0,j}$ for all $j = 1..4$. Combining the linearity of evaluation with the uniqueness of normal forms, we deduce from what precedes that

$$\begin{array}{rclcrcl} \vec{w}_1 + \vec{w}_2 & = & \lambda_1 \cdot \vec{w}_{0,1} & \qquad & \vec{w}_1 + i \cdot \vec{w}_2 & = & \lambda_3 \cdot \vec{w}_{0,3} \\ \vec{w}_1 + (-1) \cdot \vec{w}_2 & = & \lambda_2 \cdot \vec{w}_{0,2} & & \vec{w}_1 + (-i) \cdot \vec{w}_2 & = & \lambda_4 \cdot \vec{w}_{0,4} \end{array}$$

Using the polarization identity (Prop. A.6), we conclude that:

$$\begin{array}{rcl} \langle \vec{w}_1 \mid \vec{w}_2\rangle & = & \frac{1}{4}\big(\|\vec{w}_1 + \vec{w}_2\|^2 - \|\vec{w}_1 + (-1)\cdot\vec{w}_2\|^2 - i\|\vec{w}_1 + i\cdot\vec{w}_2\|^2 + i\|\vec{w}_1 + (-i)\cdot\vec{w}_2\|^2\big) \\ & = & \frac{1}{4}(\lambda_1^2\|\vec{w}_{0,1}\|^2 - \lambda_2^2\|\vec{w}_{0,2}\|^2 - i\lambda_3^2\|\vec{w}_{0,3}\|^2 + i\lambda_4^2\|\vec{w}_{0,4}\|^2) \;=\; \frac{1}{4}(\lambda_1^2 - \lambda_2^2 - i\lambda_3^2 + i\lambda_4^2) \\ & = & \frac{1}{4}(\lambda_1^2\|\vec{u}_{0,1}\|^2 - \lambda_2^2\|\vec{u}_{0,2}\|^2 - i\lambda_3^2\|\vec{u}_{0,3}\|^2 + i\lambda_4^2\|\vec{u}_{0,4}\|^2) \\ & = & \frac{1}{4}\big(\|\vec{u}_1 + \vec{u}_2\|^2 - \|\vec{u}_1 + (-1)\cdot\vec{u}_2\|^2 - i\|\vec{u}_1 + i\cdot\vec{u}_2\|^2 + i\|\vec{u}_1 + (-i)\cdot\vec{u}_2\|^2\big) \\ & = & \langle \vec{u}_1 \mid \vec{u}_2\rangle . \end{array}$$

□

Lemma A.8. *Given a valid typing judgment of the form* $\Delta, x : \sharp A, y : \sharp B \vdash \vec{s} : C$, *a substitution* $\sigma \in [\![\Delta]\!]$, *and value distributions* $\vec{u}_1, \vec{u}_2 \in [\![\sharp A]\!]$ *and* $\vec{v}_1, \vec{v}_2 \in [\![\sharp B]\!]$ *such that* $\langle \vec{u}_1 \mid \vec{u}_2\rangle = \langle \vec{v}_1 \mid \vec{v}_2\rangle = 0$, *there are value distributions* $\vec{w}_1, \vec{w}_2 \in [\![C]\!]$ *such that*

$$\vec{s}\langle \sigma, x := \vec{u}_j, y := \vec{v}_j\rangle \twoheadrightarrow \vec{w}_j \quad (j = 1..2) \qquad \textit{and} \qquad \langle \vec{w}_1 \mid \vec{w}_2\rangle = 0 .$$

Proof. From Lemma A.5, we know that there are $\vec{u}_0 \in [\![\sharp A]\!]$, $\vec{v}_0 \in [\![\sharp B]\!]$ and $\lambda, \mu \in \mathbb{C}$ such that

$$\vec{u}_2 + (-1)\cdot\vec{u}_1 \;=\; \lambda \cdot \vec{u}_0 \qquad \text{and} \qquad \vec{v}_2 + (-1)\cdot\vec{v}_1 \;=\; \mu \cdot \vec{v}_0 .$$

For all $j, k \in \{0, 1, 2\}$, we have $\sigma, x := \vec{u}_j, y := \vec{v}_k \in [\![\Delta, x : \sharp A, y : \sharp B]\!]$, hence there is $\vec{w}_{j,k} \in [\![C]\!]$ such that $\vec{s}\langle \sigma, x := \vec{u}_j, y := \vec{v}_k\rangle \twoheadrightarrow \vec{w}_{j,k}$. In particular, we can take $\vec{w}_1 := \vec{w}_{1,1}$ and $\vec{w}_2 := \vec{w}_{2,2}$. Now, we observe that

1) $\vec{u}_1 + \lambda\cdot\vec{u}_0 = \vec{u}_1 + \vec{u}_2 + (-1)\cdot\vec{u}_1 = \vec{u}_2 + 0\cdot\vec{u}_1$, so that from the linearity of substitution, the linearity of evaluation and from the uniqueness of normal forms, we get

$$\begin{array}{rcll} \vec{w}_{1,k} + \lambda\cdot\vec{w}_{0,k} & = & \vec{w}_{2,k} + 0\cdot\vec{w}_{1,k} & \\ \text{as well as} \qquad \vec{w}_{2,k} + (-\lambda)\cdot\vec{w}_{0,k} & = & \vec{w}_{1,k} + 0\cdot\vec{w}_{2,k} & \text{(for all } k \in \{0,1,2\}) \end{array}$$

2) $\vec{v}_1 + \mu\cdot\vec{v}_0 = \vec{v}_1 + \vec{v}_2 + (-1)\cdot\vec{v}_1 = \vec{v}_2 + 0\cdot\vec{v}_1$, so that from the linearity of substitution, the linearity of evaluation and from the uniqueness of normal forms, we get

$$\begin{array}{rcll} \vec{w}_{j,1} + \mu\cdot\vec{w}_{j,0} & = & \vec{w}_{j,2} + 0\cdot\vec{w}_{j,1} & \\ \text{as well as} \qquad \vec{w}_{j,2} + (-\mu)\cdot\vec{w}_{j,0} & = & \vec{w}_{j,1} + 0\cdot\vec{w}_{j,2} & \text{(for all } j \in \{0,1,2\}) \end{array}$$

3) $\langle \vec{u}_1 \mid \vec{u}_2\rangle = 0$, so that from Lemma A.7 we get $\langle \vec{w}_{1,k} \mid \vec{w}_{2,k}\rangle \;=\; 0$ (for all $k \in \{0,1,2\}$)
4) $\langle \vec{v}_1 \mid \vec{v}_2\rangle = 0$, so that from Lemma A.7 we get $\langle \vec{w}_{j,1} \mid \vec{w}_{j,2}\rangle \;=\; 0$ (for all $j \in \{0,1,2\}$)

From the above, we get:

$$\begin{array}{rcll} \langle \vec{w}_1 \mid \vec{w}_2\rangle & = & \langle \vec{w}_{1,1} \mid \vec{w}_{2,2}\rangle \;=\; \langle \vec{w}_{1,1} \mid \vec{w}_{2,2} + 0\cdot\vec{w}_{1,2}\rangle & \\ & = & \langle \vec{w}_{1,1} \mid \vec{w}_{1,2} + \lambda\cdot\vec{w}_{0,2}\rangle & \text{(from (1), } k = 2) \\ & = & \langle \vec{w}_{1,1} \mid \vec{w}_{1,2}\rangle + \lambda\langle \vec{w}_{1,1} \mid \vec{w}_{0,2}\rangle & \\ & = & 0 + \lambda\langle \vec{w}_{1,1} \mid \vec{w}_{0,2}\rangle & \text{(from (4), } j = 1) \\ & = & \lambda\langle \vec{w}_{1,1} + 0\cdot\vec{w}_{2,1} \mid \vec{w}_{0,2}\rangle & \\ & = & \lambda\langle \vec{w}_{2,1} + (-\lambda)\cdot\vec{w}_{0,1} \mid \vec{w}_{0,2}\rangle & \text{(from (1), } k = 1) \\ & = & \lambda\langle \vec{w}_{2,1} \mid \vec{w}_{0,2}\rangle - |\lambda|^2\langle \vec{w}_{0,1} \mid \vec{w}_{0,2}\rangle & \\ & = & \lambda\langle \vec{w}_{2,1} \mid \vec{w}_{0,2}\rangle - 0 & \text{(from (4), } j = 0) \\ & = & \langle \vec{w}_{2,1} \mid \vec{w}_{2,2} + (-1)\cdot\vec{w}_{1,2}\rangle & \\ & = & \langle \vec{w}_{2,1} \mid \vec{w}_{2,2}\rangle - \langle \vec{w}_{2,1} \mid \vec{w}_{1,2}\rangle & \\ & = & 0 - \langle \vec{w}_{2,1} \mid \vec{w}_{1,2}\rangle & \text{(from (4), } j = 2) \end{array}$$

Hence $\langle \vec{w}_1 \mid \vec{w}_2 \rangle = \langle \vec{w}_{1,1} \mid \vec{w}_{2,2} \rangle = -\langle \vec{w}_{2,1} \mid \vec{w}_{1,2} \rangle$. Exchanging the indices j and k in the above reasoning, we also get $\langle \vec{w}_1 \mid \vec{w}_2 \rangle = \langle \vec{w}_{1,1} \mid \vec{w}_{2,2} \rangle = -\langle \vec{w}_{1,2} \mid \vec{w}_{2,1} \rangle$, so that we have $\langle \vec{w}_1 \mid \vec{w}_2 \rangle = -\langle \vec{w}_{2,1} \mid \vec{w}_{1,2} \rangle = -\overline{\langle \vec{w}_{2,1} \mid \vec{w}_{1,2} \rangle} \in \mathbb{R}$. If we now replace $\vec{u}_2 \in [\![\sharp A]\!]$ with $i\,\vec{u}_2 \in [\![\sharp A]\!]$, the very same technique allows us to prove that $i\langle \vec{w}_1 \mid \vec{w}_2 \rangle = \langle \vec{w}_1 \mid i\vec{w}_2 \rangle \in \mathbb{R}$. Therefore $\langle \vec{w}_1 \mid \vec{w}_2 \rangle = 0$. □

Lemma A.9. *Given a valid typing judgment of the form* $\Delta, x : \sharp A, y : \sharp B \vdash \vec{s} : C$, *a substitution* $\sigma \in [\![\Delta]\!]$, *and value distributions* $\vec{u}_1, \vec{u}_2 \in [\![\sharp A]\!]$ *and* $\vec{v}_1, \vec{v}_2 \in [\![\sharp B]\!]$, *there are value distributions* $\vec{w}_1, \vec{w}_2 \in [\![C]\!]$ *such that*

$$\vec{s}\langle \sigma, x := \vec{u}_j, y := \vec{v}_j \rangle \twoheadrightarrow \vec{w}_j \quad (j = 1..2) \qquad \textit{and} \qquad \langle \vec{w}_1 \mid \vec{w}_2 \rangle = \langle \vec{u}_1 \mid \vec{u}_2 \rangle \langle \vec{v}_1 \mid \vec{v}_2 \rangle \,.$$

Proof. Let $\alpha = \langle \vec{u}_1 \mid \vec{u}_2 \rangle$ and $\beta = \langle \vec{v}_1 \mid \vec{v}_2 \rangle$. We observe that

$$\langle \vec{u}_1 \mid \vec{u}_2 + (-\alpha) \cdot \vec{u}_1 \rangle \;=\; \langle \vec{u}_1 \mid \vec{u}_2 \rangle - \alpha \langle \vec{u}_1 \mid \vec{u}_1 \rangle \;=\; \alpha - \alpha = 0$$

and, similarly, that $\langle \vec{v}_1 \mid \vec{v}_2 + (-\beta) \cdot \vec{v}_1 \rangle = 0$. From Lemma A.5, we know that there are $\vec{u}_0 \in [\![\sharp A]\!]$, $\vec{v}_0 \in [\![\sharp B]\!]$ and $\lambda, \mu \in \mathbb{C}$ such that

$$\vec{u}_2 + (-\alpha) \cdot \vec{u}_1 \;=\; \lambda \cdot \vec{u}_0 \qquad \text{and} \qquad \vec{v}_2 + (-\beta) \cdot \vec{v}_1 \;=\; \mu \cdot \vec{v}_0 \,.$$

For all $j, k \in \{0, 1, 2\}$, we have $\sigma, x := \vec{u}_j, y := \vec{v}_k \in [\![\Delta, x : \sharp A, y : \sharp B]\!]$, hence there is $\vec{w}_{j,k} \in [\![C]\!]$ such that $\vec{s}\langle \sigma, x := \vec{u}_j, y := \vec{v}_k \rangle \twoheadrightarrow \vec{w}_{j,k}$. In particular, we can take $\vec{w}_1 := \vec{w}_{1,1}$ and $\vec{w}_2 := \vec{w}_{2,2}$. Now, we observe that

1) $\lambda \cdot \vec{u}_0 + \alpha \cdot \vec{u}_1 = \vec{u}_2 + (-\alpha) \cdot \vec{u}_1 + \alpha \cdot \vec{u}_1 = \vec{u}_2 + 0 \cdot \vec{u}_1$, so that from the linearity of substitution, the linearity of evaluation and from the uniqueness of normal forms, we get
$$\lambda \cdot \vec{w}_{0,k} + \alpha \cdot \vec{w}_{1,k} = \vec{w}_{2,k} + 0 \cdot \vec{w}_{1,k} \qquad \text{(for all } k \in \{0,1,2\}\text{)}$$
2) $\mu \cdot \vec{v}_0 + \beta \cdot \vec{v}_1 = \vec{v}_2 + (-\beta) \cdot \vec{v}_1 + \beta \cdot \vec{v}_1 = \vec{v}_2 + 0 \cdot \vec{v}_1$, so that from the linearity of substitution, the linearity of evaluation and from the uniqueness of normal forms, we get
$$\mu \cdot \vec{w}_{j,0} + \beta \cdot \vec{w}_{j,1} = \vec{w}_{j,2} + 0 \cdot \vec{w}_{j,1} \qquad \text{(for all } j \in \{0,1,2\}\text{)}$$
3) $\langle \vec{u}_1 \mid \lambda \cdot \vec{u}_0 \rangle = \langle \vec{u}_1 \mid \vec{u}_2 + (-\alpha) \cdot \vec{u}_1 \rangle = 0$, so that from Lemma A.7 we get
$$\langle \vec{w}_{1,k} \mid \lambda \cdot \vec{w}_{0,k} \rangle \;=\; 0 \qquad \text{(for all } k \in \{0,1,2\}\text{)}$$
(The equality $\langle \vec{w}_{1,k} \mid \lambda \cdot \vec{w}_{0,k} \rangle = 0$ is trivial when $\lambda = 0$, and when $\lambda \neq 0$, we deduce from the above that $\langle \vec{u}_1 \mid \vec{u}_0 \rangle = 0$, from which we get $\langle \vec{w}_{1,k} \mid \vec{w}_{0,k} \rangle = 0$ by Lemma A.7.)
4) $\langle \vec{v}_1 \mid \mu \cdot \vec{v}_0 \rangle = \langle \vec{v}_1 \mid \vec{v}_2 + (-\beta) \cdot \vec{v}_1 \rangle = 0$, so that from Lemma A.7 we get
$$\langle \vec{w}_{j,1} \mid \mu \cdot \vec{w}_{j,0} \rangle \;=\; 0 \qquad \text{(for all } j \in \{0,1,2\}\text{)}$$
5) $\langle \vec{u}_1 \mid \lambda \cdot \vec{u}_0 \rangle = \langle \vec{v}_1 \mid \mu \cdot \vec{v}_0 \rangle = 0$, so that from Lemma A.8 we get
$$\langle \vec{w}_{1,1} \mid \lambda\mu \cdot \vec{w}_{0,0} \rangle \;=\; 0$$
(Again, the equality $\langle \vec{w}_{1,1} \mid \lambda\mu \cdot \vec{w}_{0,0} \rangle = 0$ is trivial when $\lambda = 0$ or $\mu = 0$, and when $\lambda, \mu \neq 0$, we deduce from the above that $\langle \vec{u}_1 \mid \vec{u}_0 \rangle = \langle \vec{v}_1 \mid \vec{v}_0 \rangle = 0$, from which we get $\langle \vec{w}_{1,1} \mid \vec{w}_{0,0} \rangle = 0$ by Lemma A.8.)

From the above, we get

$$\begin{array}{rll} & \vec{w}_{2,2} + 0 \cdot \vec{w}_{1,2} + 0 \cdot \vec{w}_{0,1} + 0 \cdot \vec{w}_{1,1} & \\ = & \lambda \cdot \vec{w}_{0,2} + \alpha \cdot \vec{w}_{1,2} + 0 \cdot \vec{w}_{0,1} + 0 \cdot \vec{w}_{1,1} & \text{(from (1), } k = 1\text{)} \\ = & \lambda \cdot (\vec{w}_{0,2} + 0 \cdot \vec{w}_{0,1}) + \alpha \cdot (\vec{w}_{1,2} + 0 \cdot \vec{w}_{1,1}) & \\ = & \lambda \cdot (\mu \cdot \vec{w}_{0,0} + \beta \cdot \vec{w}_{0,1}) + \alpha \cdot (\mu \cdot \vec{w}_{1,0} + \beta \cdot \vec{w}_{1,1}) & \text{(from (2), } j = 0, 1\text{)} \\ = & \lambda\mu \cdot \vec{w}_{0,0} + \beta\lambda \cdot \vec{w}_{0,1} + \alpha\mu \cdot \vec{w}_{1,0} + \alpha\beta \cdot \vec{w}_{1,1} & \end{array}$$

Therefore:

$$\begin{array}{rcl} \langle \vec{w}_1 \mid \vec{w}_2 \rangle & = & \langle \vec{w}_{1,1} \mid \vec{w}_{2,2} + 0 \cdot \vec{w}_{1,2} + 0 \cdot \vec{w}_{0,1} + 0 \cdot \vec{w}_{1,1} \rangle \\ & = & \langle \vec{w}_{1,1} \mid \lambda\mu \cdot \vec{w}_{0,0} + \beta\lambda \cdot \vec{w}_{0,1} + \alpha\mu \cdot \vec{w}_{1,0} + \alpha\beta \cdot \vec{w}_{1,1} \rangle \\ & = & \langle \vec{w}_{1,1} \mid \lambda\mu \cdot \vec{w}_{0,0} \rangle + \beta \langle \vec{w}_{1,1} \mid \lambda \cdot \vec{w}_{0,1} \rangle + \alpha \langle \vec{w}_{1,1} \mid \mu \cdot \vec{w}_{1,0} \rangle + \alpha\beta \langle \vec{w}_{1,1} \mid \vec{w}_{1,1} \rangle \\ & = & 0 + 0 + 0 + \alpha\beta \cdot 1 \;=\; \langle \vec{u}_1 \mid \vec{u}_2 \rangle \, \langle \vec{v}_1 \mid \vec{v}_2 \rangle \end{array}$$

from (5), (3) (with $k = 1$) and (4) (with $j = 1$), and concluding with the definition of α and β. □

Lemma A.10. *For all* $\vec{t}, \vec{s}, \vec{s}_1, \vec{s}_2 \in \vec{\Lambda}(\mathcal{X})$ *and* $\vec{v}, \vec{v}_1, \vec{v}_2, \vec{w} \in \vec{\mathrm{V}}(\mathcal{X})$*:*

1) $\mathtt{inl}(\vec{v})\langle x := \vec{w} \rangle \;=\; \mathtt{inl}(\vec{v}\langle x := \vec{w} \rangle)$
2) $\mathtt{inr}(\vec{v})\langle x := \vec{w} \rangle \;=\; \mathtt{inr}(\vec{v}\langle x := \vec{w} \rangle)$

3) *If* $x \notin FV(\vec{v}_1)$, *then* $(\vec{v}_1, \vec{v}_2)\langle x := \vec{w}\rangle \;=\; (\vec{v}_1, \vec{v}_2\langle x := \vec{w}\rangle)$
4) *If* $x \notin FV(\vec{v}_2)$, *then* $(\vec{v}_1, \vec{v}_2)\langle x := \vec{w}\rangle \;=\; (\vec{v}_1\langle x := \vec{w}\rangle, \vec{v}_2)$
5) *If* $x \notin FV(\vec{s})$, *then* $(\vec{s}\,\vec{t})\langle x := \vec{w}\rangle \;=\; \vec{s}\,\vec{t}\langle x := \vec{w}\rangle$
6) *If* $x \notin FV(\vec{t})$, *then* $(\vec{s}\,\vec{t})\langle x := \vec{w}\,\rangle \;=\; \vec{s}\langle x := \vec{w}\rangle\;\vec{t}$
7) *If* $x \notin FV(\vec{s})$, *then* $(\vec{t}; \vec{s})\langle x := \vec{w}\rangle \;=\; \vec{t}\langle x := \vec{w}\rangle; \vec{s}$
8) *If* $x \notin FV(\vec{s})$, *then* $(\texttt{let}\ (x_1, x_2) = \vec{t}\ \texttt{in}\ \vec{s})\langle x := \vec{w}\rangle \;=\; \texttt{let}\ (x_1, x_2) = \vec{t}\langle x := \vec{w}\rangle\ \texttt{in}\ \vec{s}$
9) *If* $x \notin FV(\vec{s}_1, \vec{s}_2)$, *then* $(\texttt{match}\ \vec{t}\ \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\})\langle x := \vec{w}\rangle \;=\; \texttt{match}\ \vec{t}\langle x := \vec{w}\rangle\ \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\}$ □

Proposition V.3. *The typing rules of Table VI are valid.*

Proof. (Axiom) It is clear that $\mathrm{dom}^\sharp(x : A) \subseteq \{x\} = \mathrm{dom}(x : A)$. Moreover, given $\sigma \in [\![x : A]\!]$, we have $\sigma = \{x := \vec{v}\}$ for some $\vec{v} \in [\![A]\!]$. Therefore $x\langle\sigma\rangle = x\langle x := \vec{v}\rangle = \vec{v} \Vdash A$.

(Sub) Obvious since $\{\Vdash A\} \subseteq \{\Vdash A'\}$.

(App) Suppose that both judgments $\Gamma \vdash \vec{s} : A \Rightarrow B$ and $\Delta \vdash \vec{t} : A$ are valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq FV(\vec{s}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{s}\langle\sigma\rangle \Vdash A \Rightarrow B$ for all $\sigma \in [\![\Gamma]\!]$.
- $\mathrm{dom}^\sharp(\Delta) \subseteq FV(\vec{t}) \subseteq \mathrm{dom}(\Delta)$ and $\vec{t}\langle\sigma\rangle \Vdash A$ for all $\sigma \in [\![\Delta]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq FV(\vec{s}\,\vec{t}) \subseteq \mathrm{dom}(\Gamma, \Delta)$. Now, given $\sigma \in [\![\Gamma, \Delta]\!]$, we observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. And since $FV(\vec{t}) \cap \mathrm{dom}(\sigma_\Gamma) = \varnothing$ and $FV(\vec{s}) \cap \mathrm{dom}(\sigma_\Delta) = \varnothing$, we deduce from Lemma A.10 (5), (6) p. 19 that

$$(\vec{s}\,\vec{t})\langle\sigma\rangle \;=\; (\vec{s}\,\vec{t})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \;=\; (\vec{s}\langle\sigma_\Gamma\rangle\;\vec{t})\langle\sigma_\Delta\rangle \;=\; \vec{s}\langle\sigma_\Gamma\rangle\;\vec{t}\langle\sigma_\Delta\rangle\,.$$

We conclude that $(\vec{s}\,\vec{t})\langle\sigma\rangle = \vec{s}\langle\sigma_\Gamma\rangle\;\vec{t}\langle\sigma_\Delta\rangle \Vdash B$ from Lemma A.4.

(PureLam) Given a context $\Gamma = x_1 : A_1, \ldots, x_\ell : A_\ell$ such that $\flat A_i \simeq A_i$ for all $i = 1..\ell$, we suppose that the judgment $\Gamma, x : A \vdash \vec{t} : B$ is valid, that is:

- $\mathrm{dom}^\sharp(\Gamma, x : A) \subseteq FV(\vec{t}) \subseteq \mathrm{dom}(\Gamma, x : A)$ and $\vec{t}\langle\sigma\rangle \Vdash B$ for all $\sigma \in [\![\Gamma, x : A]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma) \subseteq FV(\lambda x\,.\,\vec{t}) \subseteq \mathrm{dom}(\Gamma)$. Now, given $\sigma \in [\![\Gamma]\!]$, we want to prove that $(\lambda x\,.\,\vec{t})\langle\sigma\rangle \Vdash A \to B$. Due to our initial assumption on the context Γ, it is clear that $\sigma = \{x_1 := v_1, \ldots, x_\ell := v_\ell\}$ for some closed pure values $v_1, \ldots, v_\ell$. Hence

$$(\lambda x\,.\,\vec{t})\langle\sigma\rangle \;=\; (\lambda x\,.\,\vec{t})[x_1 := v_1]\cdots[x_\ell := v_\ell] \;=\; \lambda x\,.\,\vec{t}[x_1 := v_1]\cdots[x_\ell := v_\ell]$$

(since the variables $x_1, \ldots, x_\ell$ are all distinct from x). For all $\vec{v} \in [\![A]\!]$, we observe that

$$(\vec{t}[x_1 := v_1]\cdots[x_\ell := v_\ell])\langle x := \vec{v}\,\rangle \;=\; \vec{t}\langle\sigma, \{x := \vec{v}\,\}\rangle \;\Vdash\; B\,,$$

since $\sigma, \{x := \vec{v}\,\} \in [\![\Gamma, x : A]\!]$. Therefore $(\lambda x\,.\,\vec{t})\langle\sigma\rangle \Vdash A \to B$.

(UnitLam) Suppose that the judgment $\Gamma, x : A \vdash \vec{t} : B$ is valid, that is:

- $\mathrm{dom}^\sharp(\Gamma, x : A) \subseteq FV(\vec{t}) \subseteq \mathrm{dom}(\Gamma, x : A)$ and $\vec{t}\langle\sigma\rangle \Vdash B$ for all $\sigma \in [\![\Gamma, x : A]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma) \subseteq FV(\lambda x\,.\,\vec{t}) \subseteq \mathrm{dom}(\Gamma)$. Now, given $\sigma \in [\![\Gamma]\!]$, we want to prove that $(\lambda x\,.\,\vec{t})\langle\sigma\rangle \Vdash A \Rightarrow B$. For that, we write:

- $\Gamma = x_1 : A_1, \ldots, x_\ell : A_\ell$ (where $x_1, \ldots, x_\ell$ are all distinct from x);
- $\sigma = \{x_1 := \vec{v}_1, \ldots, x_\ell := \vec{v}_\ell\}$ (where $\vec{v}_i \in [\![A_i]\!]$ for all $i = 1..\ell$);
- $\vec{v}_i = \sum_{j=1}^{n_i} \alpha_{i,j} \cdot v_{i,j}$ (in canonical form) for all $i = 1..\ell$.

Now we observe that

$$\begin{array}{rcl}(\lambda x\,.\,t)\langle\sigma\rangle & = & \sum_{i_1=1}^{n_1}\cdots\sum_{i_\ell=1}^{n_\ell} \alpha_{1,i_1}\cdots\alpha_{\ell,i_\ell} \cdot (\lambda x\,.\,\vec{t})[x_1 := v_{1,i_1}]\cdots[x_\ell := v_{\ell,i_\ell}] \\ & = & \sum_{i_1=1}^{n_1}\cdots\sum_{i_\ell=1}^{n_\ell} \alpha_{1,i_1}\cdots\alpha_{\ell,i_\ell} \cdot \lambda x\,.\,\vec{t}[x_1 := v_{1,i_1}]\cdots[x_\ell := v_{\ell,i_\ell}] \\ & = & \sum_{i\in I} \alpha_i \cdot \lambda x\,.\,\vec{t}_i\end{array}$$

writing

- $I := [1..n_1] \times \cdots \times [1..n_\ell]$ the (finite) set of all multi-indices $i = (i_1, \ldots, i_\ell)$;
- $\alpha_i := \alpha_{1,i_1}\cdots\alpha_{\ell,i_\ell}$ and $\vec{t}_i := \vec{t}[x_1 := v_{1,i_1}]\cdots[x_\ell := v_{\ell,i_\ell}]$ for each multi-index $i = (i_1, \ldots, i_\ell) \in I$.

We now want to prove that $\left(\sum_{i\in I} \alpha_i \cdot \lambda x\,.\,\vec{t}_i\right) \in \mathcal{S}_1$. For that, we first observe that

$$\textstyle\sum_{i\in I} |\alpha_i|^2 \;=\; \sum_{i_1=1}^{n_1}\cdots\sum_{i_\ell=1}^{n_\ell} |\alpha_{1,i_1}\cdots\alpha_{\ell,i_\ell}|^2 \;=\; \left(\sum_{i_1=1}^{n_1} |\alpha_{1,i_1}|^2\right) \times \cdots \times \left(\sum_{i_\ell=1}^{n_\ell} |\alpha_{\ell,i_\ell}|^2\right) \;=\; 1\,.$$

Then we need to check that the λ-abstractions $\lambda x\,.\,\vec{t_i}$ $(i \in I)$ are pairwise distinct. For that, consider two multi-indices $i = (i_1, \dots, i_\ell)$ and $i' = (i'_1, \dots, i'_\ell)$ such that $i \neq i'$. This means that $i_k \neq i'_k$ for some $k \in [1..\ell]$. From the latter, we deduce that $n_k \geq 2$, hence $\vec{v}_k = \sum_{j=1}^{n_k} \alpha_{k,j} \cdot v_{k,j}$ is not a pure value, and thus $[\![A_k]\!] \neq \flat[\![A_k]\!]$. Therefore $x_k \in \mathrm{dom}^\sharp(\Gamma)$, from which we deduce that $x_k \in \mathit{FV}(\vec{t})$ from our initial assumption. Let us now consider the first occurrence of the variable x_k in the (raw) term distribution $\vec{t}$. At this occurrence, the variable x_k is replaced

- by v_{k,i_k} in the multiple substitution $\vec{t}[x_1 := v_{1,i_1}] \cdots [x_\ell := v_{\ell,i_\ell}]$ $(= \vec{t_i})$, and
- by v_{k,i'_k} in the multiple substitution $\vec{t}[x_1 := v_{1,i'_1}] \cdots [x_\ell := v_{\ell,i'_\ell}]$ $(= \vec{t_{i'}})$.

And since $v_{k,i_k} \neq v_{k,i'_k}$ (recall that $\vec{v}_k = \sum_{j=1}^{n_k} \alpha_{k,j} \cdot v_{k,j}$ is in canonical form), we deduce that $\vec{t_i} \neq \vec{t_{i'}}$. Which concludes the proof that $\big(\sum_{i\in I} \alpha_i \cdot \lambda x\,.\,\vec{t_i}\big) \in \mathcal{S}_1$. Now, given $\vec{v} \in [\![A]\!]$, it remains to show that $\sum_{i\in I} \alpha_i \cdot \vec{t_i}\langle x := \vec{v}\rangle \Vdash B$. For that, it suffices to observe that:

$$\begin{array}{rcl}
\sum_{i\in I} \alpha_i \cdot \vec{t_i}\langle x := \vec{v}\rangle & = & \big(\sum_{i\in I} \alpha_i \cdot \vec{t_i}\big)\langle x := \vec{v}\rangle \\
& = & \big(\sum_{i_1=1}^{n_1} \cdots \sum_{i_\ell=1}^{n_\ell} \alpha_{1,i_1} \cdots \alpha_{\ell,i_\ell} \cdot \vec{t}[x_1 := v_{1,i_1}] \cdots [x_\ell := v_{\ell,i_\ell}]\big)\langle x := \vec{v}\rangle \\
& = & \big(\vec{t}\langle\sigma\rangle\big)\langle x := \vec{v}\rangle \;=\; \vec{t}\langle\sigma, \{x := \vec{v}\}\rangle \;\Vdash\; B
\end{array}$$

since $\sigma, \{x := \vec{v}\} \in [\![\Gamma, x : A]\!]$. Therefore $(\lambda x\,.\,\vec{t})\langle\sigma\rangle = \sum_{i\in I} \alpha_i \cdot \vec{t_i} \in [\![A \Rightarrow B]\!] \subseteq \{\Vdash A \Rightarrow B\}$.

(Void) Obvious.

(Seq) Suppose that the judgments $\Gamma \vdash \vec{t} : \mathbb{U}$ and $\Delta \vdash \vec{s} : A$ are valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma\rangle \twoheadrightarrow *$ for all $\sigma \in [\![\Gamma]\!]$.
- $\mathrm{dom}^\sharp(\Delta) \subseteq \mathit{FV}(\vec{s}) \subseteq \mathrm{dom}(\Delta)$ and $\vec{s}\langle\sigma\rangle \Vdash A$ for all $\sigma \in [\![\Delta]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathit{FV}(\vec{t}; \vec{s}) \subseteq \mathrm{dom}(\Gamma, \Delta)$. Now, given $\sigma \in [\![\Gamma, \Delta]\!]$, we observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. From our initial hypotheses, we get

$$(\vec{t}; \vec{s})\langle\sigma\rangle \;=\; (\vec{t}; \vec{s})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \;=\; (\vec{t}\langle\sigma_\Gamma\rangle; \vec{s})\langle\sigma_\Delta\rangle \;\twoheadrightarrow\; (*; \vec{s})\langle\sigma_\Delta\rangle \;\twoheadrightarrow\; \vec{s}\langle\sigma_\Delta\rangle \;\Vdash\; A$$

(using Lemma A.10 (7) p. 19 and Lemma A.3 (7), (10) p. 16).

(SeqSharp) Suppose that the judgments $\Gamma \vdash \vec{t} : \sharp\mathbb{U}$ and $\Delta \vdash \vec{s} : \sharp A$ are valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma\rangle \Vdash \sharp\mathbb{U}$ for all $\sigma \in [\![\Gamma]\!]$.
- $\mathrm{dom}^\sharp(\Delta) \subseteq \mathit{FV}(\vec{s}) \subseteq \mathrm{dom}(\Delta)$ and $\vec{s}\langle\sigma\rangle \Vdash \sharp A$ for all $\sigma \in [\![\Delta]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathit{FV}(\vec{t}; \vec{s}) \subseteq \mathrm{dom}(\Gamma, \Delta)$. Now, given $\sigma \in [\![\Gamma, \Delta]\!]$, we observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. From our first hypothesis, we get $\vec{t}\langle\sigma_\Gamma\rangle \twoheadrightarrow \alpha \cdot *$ for some $\alpha \in \mathbb{C}$ such that $|\alpha| = 1$. And from the second hypothesis, we have $\vec{s}\langle\sigma_\Delta\rangle \Vdash \sharp A$, and thus $\alpha \cdot \vec{s}\langle\sigma_\Delta\rangle \Vdash \sharp A$ (since $|\alpha| = 1$). Therefore, we get

$$(\vec{t}; \vec{s})\langle\sigma\rangle \;=\; (\vec{t}; \vec{s})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \;=\; (\vec{t}\langle\sigma_\Gamma\rangle; \vec{s})\langle\sigma_\Delta\rangle \;\twoheadrightarrow\; (\alpha \cdot *; \vec{s})\langle\sigma_\Delta\rangle \;=\; \alpha \cdot (*; \vec{s})\langle\sigma_\Delta\rangle \;\twoheadrightarrow\; \alpha \cdot \vec{s}\langle\sigma_\Delta\rangle \;\Vdash\; A$$

(using Lemma A.10 (7) p. 19 and Lemma A.3 (7), (10) p. 16).

(Pair) Suppose that the judgments $\Gamma \vdash \vec{v} : A$ and $\Delta \vdash \vec{w} : B$ are valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{v}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{v}\langle\sigma\rangle \Vdash A$ for all $\sigma \in [\![\Gamma]\!]$.
- $\mathrm{dom}^\sharp(\Delta) \subseteq \mathit{FV}(\vec{w}) \subseteq \mathrm{dom}(\Delta)$ and $\vec{w}\langle\sigma\rangle \Vdash B$ for all $\sigma \in [\![\Delta]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathit{FV}((\vec{v}, \vec{w})) \subseteq \mathrm{dom}(\Gamma, \Delta)$. Now, given $\sigma \in [\![\Gamma, \Delta]\!]$, we observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. From our initial hypotheses, we deduce that $\vec{v}\langle\sigma_\Gamma\rangle \Vdash A$ and $\vec{w}\langle\sigma_\Delta\rangle \Vdash B$, which means that $\vec{v}\langle\sigma_\Gamma\rangle \in [\![A]\!]$ and $\vec{w}\langle\sigma_\Delta\rangle \in [\![B]\!]$ (from Lemma IV.3), since $\vec{v}\langle\sigma_\Gamma\rangle$ and $\vec{w}\langle\sigma_\Delta\rangle$ are value distributions. And since $\mathit{FV}(\vec{v}) \cap \mathrm{dom}(\sigma_\Delta) = \varnothing$ and $\mathit{FV}(\vec{w}) \cap \mathrm{dom}(\sigma_\Gamma) = \varnothing$, we deduce from Lemma A.10 (3), (4) p. 19 that

$$(\vec{v}, \vec{w})\langle\sigma\rangle \;=\; (\vec{v}, \vec{w})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \;=\; (\vec{v}\langle\sigma_\Gamma\rangle, \vec{w})\langle\sigma_\Delta\rangle \;=\; (\vec{v}\langle\sigma_\Gamma\rangle, \vec{w}\langle\sigma_\Delta\rangle) \;\in\; [\![A \times B]\!]$$

from the definition of $[\![A \times B]\!]$.

(LetPair) Suppose that the judgments $\Gamma \vdash \vec{t} : A \times B$ and $\Delta, x : A, y : B \vdash \vec{s} : C$ are valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma\rangle \Vdash A \times B$ for all $\sigma \in [\![\Gamma]\!]$.
- $\mathrm{dom}^\sharp(\Delta, x : A, y : B) \subseteq \mathit{FV}(\vec{s}) \subseteq \mathrm{dom}(\Delta, x : A, y : B)$ and $\vec{s}\langle\sigma\rangle \Vdash C$ for all $\sigma \in [\![\Delta, x : A, y : B]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma,\Delta) \subseteq \mathit{FV}(\mathtt{let}\ (x,y) = \vec{t}\ \mathtt{in}\ \vec{s}) \subseteq \mathrm{dom}(\Gamma,\Delta)$. Now, given $\sigma \in [\![\Gamma,\Delta]\!]$, we observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. Since $\sigma_\Gamma \in [\![\Gamma]\!]$, we know from our first hypothesis that $\vec{t}\langle\sigma_\Gamma\rangle \Vdash A \times B$, which means that $\vec{t}\langle\sigma_\Gamma\rangle \twoheadrightarrow (\vec{v},\vec{w})$ for some $\vec{v} \in [\![A]\!]$ and $\vec{w} \in [\![B]\!]$. So that we get

$$\begin{array}{rcll}
(\mathtt{let}\ (x,y) = \vec{t}\ \mathtt{in}\ \vec{s})\langle\sigma\rangle & = & (\mathtt{let}\ (x,y) = \vec{t}\ \mathtt{in}\ \vec{s})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle & \\
& = & (\mathtt{let}\ (x,y) = \vec{t}\langle\sigma_\Gamma\rangle\ \mathtt{in}\ \vec{s})\langle\sigma_\Delta\rangle & \text{(by Lemma A.10 (8))} \\
& \twoheadrightarrow & (\mathtt{let}\ (x,y) = (\vec{v},\vec{w})\ \mathtt{in}\ \vec{s})\langle\sigma_\Delta\rangle & \text{(by Lemma A.3 (8), (10))} \\
& \twoheadrightarrow & (\vec{s}\langle x := \vec{v}\rangle\langle y := \vec{w}\rangle)\langle\sigma_\Delta\rangle & \text{(by Lemma A.3 (2), (10))} \\
& & = \ \vec{s}\langle\sigma_\Delta, x := \vec{v}, y := \vec{w}\rangle \ \Vdash\ C &
\end{array}$$

using our second hypothesis with the substitution $\sigma_\Delta, \{x := \vec{v}, y := \vec{w}\} \in [\![\Delta, x : A, y : B]\!]$.

(LetTens) Suppose that the judgments $\Gamma \vdash \vec{t} : A \otimes B$ and $\Delta, x : \sharp A, y : \sharp B \vdash \vec{s} : \sharp C$ are valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma\rangle \Vdash A \otimes B$ for all $\sigma \in [\![\Gamma]\!]$.
- $\mathrm{dom}^\sharp(\Delta, x : \sharp A, y : \sharp B) \subseteq \mathit{FV}(\vec{s}) \subseteq \mathrm{dom}(\Delta, x : \sharp A, y : \sharp B)$ and $\vec{s}\langle\sigma\rangle \Vdash \sharp C$ for all $\sigma \in [\![\Delta, x : \sharp A, y : \sharp B]\!]$

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma,\Delta) \subseteq \mathit{FV}(\mathtt{let}\ (x,y) = \vec{t}\ \mathtt{in}\ \vec{s}) \subseteq \mathrm{dom}(\Gamma,\Delta)$. Now, given $\sigma \in [\![\Gamma,\Delta]\!]$, we observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. Since $\sigma_\Gamma \in [\![\Gamma]\!]$, we know from our first hypothesis that $\vec{t}\langle\sigma_\Gamma\rangle \Vdash A \otimes B$, which means that $\vec{t}\langle\sigma_\Gamma\rangle \twoheadrightarrow \sum_{i=1}^n \alpha_i \cdot (\vec{u}_i, \vec{v}_i)$ for some $\alpha_1, \ldots, \alpha_n \in \mathbb{C}$, $\vec{u}_1, \ldots, \vec{u}_n \in [\![A]\!]$ and $\vec{v}_1, \ldots, \vec{v}_n \in [\![B]\!]$, with $\left\|\sum_{i=1}^n \alpha_i \cdot (\vec{u}_i, \vec{v}_i)\right\| = 1$. For each $i = 1..n$, we also observe that $\sigma_\Delta, x := \vec{u}_i, y := \vec{v}_i \in [\![\Delta, x : \sharp A, y : \sharp B]\!]$. From our second hypothesis, we get $\vec{s}\langle\sigma_\Delta, x := \vec{u}_i, y := \vec{v}_i\rangle \Vdash \sharp C$, hence there is $\vec{w}_i \in [\![\sharp C]\!]$ such that $\vec{s}\langle\sigma_\Delta, x := \vec{u}_i, y := \vec{v}_i\rangle \twoheadrightarrow \vec{w}_i$. Therefore, we have:

$$\begin{array}{rcl}
(\mathtt{let}\ (x,y) = \vec{t}\ \mathtt{in}\ \vec{s})\langle\sigma\rangle & = & (\mathtt{let}\ (x,y) = \vec{t}\ \mathtt{in}\ \vec{s})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \\
& = & (\mathtt{let}\ (x,y) = \vec{t}\langle\sigma_\Gamma\rangle\ \mathtt{in}\ \vec{s})\langle\sigma_\Delta\rangle \\
& \twoheadrightarrow & \left(\mathtt{let}\ (x,y) = \sum_{i=1}^n \alpha_i \cdot (\vec{u}_i, \vec{v}_i)\ \mathtt{in}\ \vec{s}\right)\langle\sigma_\Delta\rangle \\
& = & \sum_{i=1}^n \alpha_i \cdot (\mathtt{let}\ (x,y) = (\vec{u}_i, \vec{v}_i)\ \mathtt{in}\ \vec{s})\langle\sigma_\Delta\rangle \\
& \twoheadrightarrow & \sum_{i=1}^n \alpha_i \cdot (\vec{s}\langle x := \vec{u}_i, y := \vec{v}_i\rangle)\langle\sigma_\Delta\rangle \\
& = & \sum_{i=1}^n \alpha_i \cdot \vec{s}\langle\sigma_\Delta, x := \vec{u}_i, y := \vec{v}_i\rangle \\
& \twoheadrightarrow & \sum_{i=1}^n \alpha_i \cdot \vec{w}_i \ \in\ \mathrm{span}([\![C]\!])
\end{array}$$

To conclude, it remains to show that $\left\|\sum_{i=1}^n \alpha_i \cdot \vec{w}_i\right\| = 1$. For that, we observe that:

$$\begin{array}{rcll}
\left\|\sum_{i=1}^n \alpha_i \cdot \vec{w}_i\right\|^2 & = & \left\langle \sum_{i=1}^n \alpha_i \cdot \vec{w}_i \ \middle|\ \sum_{j=1}^n \alpha_j \cdot \vec{w}_j \right\rangle & \\
& = & \sum_{i=1}^n \sum_{j=1}^n \bar{\alpha}_i \alpha_j \langle \vec{w}_i \mid \vec{w}_j \rangle & \\
& = & \sum_{i=1}^n \sum_{j=1}^n \bar{\alpha}_i \alpha_j \langle \vec{u}_i \mid \vec{u}_j \rangle \langle \vec{v}_i \mid \vec{v}_j \rangle & \text{(by Lemma A.9)} \\
& = & \sum_{i=1}^n \sum_{j=1}^n \bar{\alpha}_i \alpha_j \langle (\vec{u}_i, \vec{v}_i) \mid (\vec{u}_j, \vec{v}_j) \rangle & \text{(by Prop. A.2)} \\
& = & \left\langle \sum_{i=1}^n \alpha_i \cdot (\vec{u}_i, \vec{v}_i) \ \middle|\ \sum_{j=1}^n \alpha_j \cdot (\vec{u}_j, \vec{v}_j) \right\rangle & \\
& = & \left\|\sum_{i=1}^n \alpha_i \cdot (\vec{u}_i, \vec{v}_i)\right\|^2 \ = \ 1\,. &
\end{array}$$

(InL) Suppose that the judgment $\Gamma \vdash \vec{v} : A$ is valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{v}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{v}\langle\sigma\rangle \Vdash A$ for all $\sigma \in [\![\Gamma]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\mathtt{inl}(\vec{v})) \subseteq \mathrm{dom}(\Gamma)$. Now, given $\sigma \in [\![\Gamma]\!]$, we know that $\vec{v}\langle\sigma\rangle \Vdash A$, which means that $\vec{v}\langle\sigma\rangle \in [\![A]\!]$ (by Lemma IV.3), since $\vec{v}\langle\sigma\rangle$ is a value distribution. So that by Lemma A.10 (1), we conclude that $\mathtt{inl}(\vec{v})\langle\sigma\rangle = \mathtt{inl}(\vec{v}\langle\sigma\rangle) \in [\![A + B]\!]$.

(InR) Analogous to (InL).

(PureMatch) Suppose that the judgments $\Gamma \vdash \vec{t} : A + B$, $\Delta, x_1 : A \vdash \vec{s}_1 : C$ and $\Delta, x_2 : B \vdash \vec{s}_2 : C$ are valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma\rangle \Vdash A + B$ for all $\sigma \in [\![\Gamma]\!]$.
- $\mathrm{dom}^\sharp(\Delta, x_1 : A) \subseteq \mathit{FV}(\vec{s}_1) \subseteq \mathrm{dom}(\Delta, x_1 : A)$ and $\vec{s}_1\langle\sigma\rangle \Vdash C$ for all $\sigma \in [\![\Delta, x_1 : A]\!]$.
- $\mathrm{dom}^\sharp(\Delta, x_2 : B) \subseteq \mathit{FV}(\vec{s}_2) \subseteq \mathrm{dom}(\Delta, x_2 : B)$ and $\vec{s}_2\langle\sigma\rangle \Vdash C$ for all $\sigma \in [\![\Delta, x_2 : B]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma,\Delta) \subseteq \mathit{FV}(\mathtt{match}\ \vec{t}\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\}) \subseteq \mathrm{dom}(\Gamma,\Delta)$. Now, given a substitution $\sigma \in [\![\Gamma,\Delta]\!]$, we observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. And since $\mathit{FV}(\vec{s}_1, \vec{s}_2) \cap \mathrm{dom}(\sigma_\Gamma) = \varnothing$, we deduce from Lemma A.10 (9) that

$$\begin{array}{rl}
& (\mathtt{match}\ \vec{t}\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma\rangle \\
= & (\mathtt{match}\ \vec{t}\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \\
= & (\mathtt{match}\ \vec{t}\langle\sigma_\Gamma\rangle\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle\,.
\end{array}$$

Moreover, since $\sigma_\Gamma \in [\![\Gamma]\!]$, we have $\vec{t}\langle\sigma_\Gamma\rangle \Vdash A + B$ (from our first hypothesis), so that we distinguish the following two cases:

- Either $\vec{t}\langle\sigma_\Gamma\rangle \succ\!\!\succ \mathtt{inl}(\vec{v})$ for some $\vec{v} \in [\![A]\!]$, so that

$$\begin{array}{rl} & (\mathtt{match}\ \vec{t}\,\{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma\rangle \\ = & (\mathtt{match}\ \vec{t}\langle\sigma_\Gamma\rangle\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle \\ \succ\!\!\succ & (\mathtt{match}\ \mathtt{inl}(\vec{v})\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle \\ \succ\!\!\succ & (\vec{s}_1\langle x_1 := \vec{v}\rangle)\langle\sigma_\Delta\rangle \ = \ \vec{s}_1\langle\sigma_\Delta, x_1 := \vec{v}\rangle \ \Vdash \ C \end{array}$$

  using our second hypothesis with the substitution $\sigma_\Delta, \{x_1 := \vec{v}\} \in [\![\Delta, x_1 : A]\!]$.
- Either $\vec{t}\langle\sigma_\Gamma\rangle \succ\!\!\succ \mathtt{inr}(\vec{w})$ for some $\vec{w} \in [\![B]\!]$, so that

$$\begin{array}{rl} & (\mathtt{match}\ \vec{t}\,\{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma\rangle \\ = & (\mathtt{match}\ \vec{t}\langle\sigma_\Gamma\rangle\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle \\ \succ\!\!\succ & (\mathtt{match}\ \mathtt{inr}(\vec{w})\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle \\ \succ\!\!\succ & (\vec{s}_1\langle x_2 := \vec{w}\rangle)\langle\sigma_\Delta\rangle \ = \ \vec{s}_1\langle\sigma_\Delta, x_2 := \vec{w}\rangle \ \Vdash \ C \end{array}$$

  using our third hypothesis with the substitution $\sigma_\Delta, \{x_2 := \vec{w}\} \in [\![\Delta, x_2 : B]\!]$.

(Weak) Suppose that the judgment $\Gamma \vdash \vec{t} : B$ is valid, that is

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma\rangle \Vdash B$ for all $\sigma \in [\![\Gamma]\!]$.

Given a type A such that $\flat A \simeq A$, it is clear from the above that $\mathrm{dom}^\sharp(\Gamma, x : A)\ (= \mathrm{dom}^\sharp(\Gamma)) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma, x : A)$. Now, given $\sigma \in [\![\Gamma, x : A]\!]$, we observe that $\sigma = \sigma_0, \{x := v\}$ for some substitution $\sigma_0 \in [\![\Gamma]\!]$ and for some pure value $v \in [\![A]\!]\ (= \flat[\![A]\!])$. Therefore, we get

$$\vec{t}\langle\sigma\rangle \ = \ \vec{t}\langle\sigma_0\rangle[x := v] \ = \ \vec{t}[x := v]\langle\sigma_0\rangle \ = \ \vec{t}\langle\sigma_0\rangle \ \Vdash B \qquad \text{(since } x \notin \mathit{FV}(\vec{t}) \text{ and } \sigma_0 \in [\![\Gamma]\!]\text{)}$$

(Contr) Given a type A such that $\flat A \simeq A$, suppose that $\Gamma, x : A, y : A \vdash \vec{t} : B$, that is:

- $\mathrm{dom}^\sharp(\Gamma, x : A, y : A)\ (= \mathrm{dom}^\sharp(\Gamma)) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma, x : A, y : A)$
  and $\vec{t}\langle\sigma\rangle \Vdash B$ for all $\sigma \in [\![\Gamma, x : A, y : A]\!]$.

From the above, it is clear that $\mathrm{dom}^\sharp(\Gamma, x : A)\ (= \mathrm{dom}^\sharp(\Gamma)) \subseteq \mathit{FV}(\vec{t}[y := x]) \subseteq \mathrm{dom}(\Gamma, x : A)$. Now, given $\sigma \in [\![\Gamma, x : A]\!]$, we observe that $\sigma = \sigma_0, \{x := v\}$ for some substitution $\sigma_0 \in [\![\Gamma]\!]$ and for some pure value $v \in [\![A]\!]\ (= \flat[\![A]\!])$. Therefore, we have

$$\begin{array}{rcl} (\vec{t}[y := x])\langle\sigma\rangle & = & (\vec{t}[y := x])\langle\sigma_0, \{x := v\}\rangle \ = \ \vec{t}[y := x][x := v]\langle\sigma_0\rangle \\ & = & \vec{t}[x := v][y := v]\langle\sigma_0\rangle \ = \ \vec{t}\langle\sigma_0, \{x := v, y := v\}\rangle \ \Vdash \ B \end{array}$$

since $\sigma_0, \{x := v, y := v\} \in [\![\Gamma, x : A, y : A]\!]$. □

Fact A.11. *For all $n \neq 1$, one has: $\bar{n} \not\Vdash (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}) \Rightarrow (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B})$.*

Proof. Let $F := \frac{3}{5} \cdot \big(\lambda x\,.\,\frac{5}{6} \cdot x\big) + \frac{4}{5} \cdot \big(\lambda x\,.\,\frac{5}{8} \cdot x\big)$. We observe that $\big|\frac{3}{5}\big|^2 + \big|\frac{4}{5}\big|^2 = \frac{9+16}{25} = 1$. Moreover, for all $\vec{v} \in [\![\mathbb{B}]\!]$, we have

$$\tfrac{3}{5} \cdot \big(\tfrac{5}{6} \cdot x\big)\langle x := \vec{v}\rangle + \tfrac{4}{5} \cdot \big(\tfrac{5}{8} \cdot x\big)\langle x := \vec{v}\rangle \ = \ \tfrac{1}{2} \cdot \vec{v} + \tfrac{1}{2} \cdot \vec{v} \ = \ \vec{v} \ \Vdash \ \sharp\mathbb{B}\,,$$

hence $F \Vdash \sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$. Now, we observe that when $n \neq 1$, we have

$$\begin{array}{rcl} \bar{n}\ F\ \mathtt{tt} & = & \frac{3}{5} \cdot \bar{n}\ \big(\lambda x\,.\,\frac{5}{6} \cdot x\big)\ \mathtt{tt} \ + \ \frac{4}{5} \cdot \bar{n}\ \big(\lambda x\,.\,\frac{5}{8} \cdot x\big)\ \mathtt{tt} \\ & \succ\!\!\succ & \frac{3}{5}\big(\frac{5}{6}\big)^n \cdot \mathtt{tt} \ + \ \frac{4}{5}\big(\frac{5}{8}\big)^n \cdot \mathtt{tt} \ = \ \big(\frac{3}{5}\big(\frac{5}{6}\big)^n + \frac{4}{5}\big(\frac{5}{8}\big)^n\big) \cdot \mathtt{tt} \ \notin \ [\![\sharp\mathbb{B}]\!]\,, \end{array}$$

since $\frac{3}{5}\big(\frac{5}{6}\big)^n + \frac{4}{5}\big(\frac{5}{8}\big)^n = \frac{7}{5} > 1$ when $n = 0$ and $\frac{3}{5}\big(\frac{5}{6}\big)^n + \frac{4}{5}\big(\frac{5}{8}\big)^n < \frac{3}{5} \cdot \frac{5}{6} + \frac{4}{5} \cdot \frac{5}{8} = 1$ when $n \geq 2$. Hence $\bar{n}\,F\,\mathtt{tt} \not\Vdash \sharp\mathbb{B}$, and therefore $\bar{n} \not\Vdash (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}) \Rightarrow (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B})$. □

Proposition V.7. *The rule* (UnitaryMatch) *is valid.*

Proof. Suppose that the judgments $\Gamma \vdash \vec{t} : A_1 \oplus A_2$ and $\Delta \vdash (x_1 : \sharp A_1 \vdash \vec{s}_1) \perp (x_2 : \sharp A_2 \vdash \vec{s}_2) : \sharp C$ are valid, that is:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathit{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma\rangle \Vdash A_1 \oplus A_2$ for all $\sigma \in [\![\Gamma]\!]$.
- For $i = 1, 2$, $\mathrm{dom}^\sharp(\Delta, x_i : \sharp A_i) \subseteq \mathit{FV}(\vec{s}_i) \subseteq \mathrm{dom}(\Delta, x_i : \sharp A_i)$ and $\vec{s}_i\langle\sigma, \sigma_i\rangle \Vdash \sharp C$ for all $\sigma \in [\![\Delta]\!]$ and $\sigma_i \in [\![x_i : \sharp A_i]\!]$.
- For $i = 1, 2$, $\vec{s}_i\langle\sigma, \sigma_i\rangle \succ\!\!\succ \vec{v}_i$ with $\langle\vec{v}_1|\vec{v}_2\rangle = 0$.

From the above, it is clear the $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathit{FV}(\mathtt{match}\ \vec{t}\,\{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\}) \subseteq \mathrm{dom}(\Gamma, \Delta)$. Now, given a substitution $\sigma \in [\![\Gamma, \Delta]\!]$, we observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. And since $\mathit{FV}(\vec{s}_1, \vec{s}_2) \cap \mathrm{dom}(\sigma_\Gamma) = \varnothing$, we deduce from Lemma A.10 (8) that

$$(\mathtt{match}\ \vec{t}\,\{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma\rangle$$

$$= (\texttt{match}\ \vec{t}\ \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle$$
$$= (\texttt{match}\ \vec{t}\langle\sigma_\Gamma\rangle\ \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle\,.$$

Moreover, since $\sigma_\Gamma \in [\![\Gamma]\!]$, we have $\vec{t}\langle\sigma_\Gamma\rangle \Vdash A_1 \oplus A_2$ (from our first hypothesis), so that we have $\vec{t}\langle\sigma_\Gamma\rangle \twoheadrightarrow \alpha \cdot \texttt{inl}(\vec{v}_1) + \beta \cdot \texttt{inr}(\vec{v}_2)$ for some $\vec{v}_1 \in [\![A_1]\!]$ and $\vec{v}_2 \in [\![A_2]\!]$. Therefore

$$\begin{array}{rl} & (\texttt{match}\ \vec{t}\ \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma\rangle \\ = & (\texttt{match}\ \vec{t}\langle\sigma_\Gamma\rangle\ \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle \\ \twoheadrightarrow & (\texttt{match}\ \alpha \cdot \texttt{inl}(\vec{v}_1) + \beta \cdot \texttt{inr}(\vec{v}_2)\ \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle \\ = & \alpha \cdot (\texttt{match}\ \texttt{inl}(\vec{v}_1)\ \{\texttt{inl}(x_1) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle \\ & + \beta \cdot (\texttt{match}\ \texttt{inr}(\vec{v}_2)\ \{\texttt{inl}(x_2) \mapsto \vec{s}_1 \mid \texttt{inr}(x_2) \mapsto \vec{s}_2\})\langle\sigma_\Delta\rangle \\ = & \alpha \cdot \vec{s}_1\langle x_1 := \vec{v}_1\rangle\langle\sigma_\Delta\rangle + \beta \cdot \vec{s}_2\langle x_2 := \vec{v}_2\rangle\langle\sigma_\Delta\rangle \Vdash \sharp C \end{array}$$

using the last two hypotheses, with the substitution $\sigma_\Delta, \langle x_i := \vec{v}_i\rangle \in [\![\Delta, x_i : \sharp A_i]\!]$. □

D. Proofs related to Section VI

Typing rules of the standard judgements for λ_Q

$$\frac{}{\Delta, x : A \vdash_C x : A} \qquad \frac{}{\Delta \vdash_C * : \mathbb{U}} \qquad \frac{\Delta, x : A \vdash_C t : B}{\Delta \vdash_C \lambda x.t : A \to B} \qquad \frac{\Delta \vdash_C t : A \to B \quad \Delta \vdash_C r : A}{\Delta \vdash_C tr : B}$$

$$\frac{\Delta \vdash_C t : A \quad \Delta \vdash_C r : B}{\Delta \vdash_C (t, r) : A \times B} \qquad \frac{\Delta \vdash_C t : A \times B}{\Delta \vdash_C \pi_1 t : A} \qquad \frac{\Delta \vdash_C t : A \times B}{\Delta \vdash_C \pi_2 t : B} \qquad \frac{}{\Delta \vdash_C \texttt{tt} : \mathsf{bit}} \qquad \frac{}{\Delta \vdash_C \texttt{ff} : \mathsf{bit}}$$

$$\frac{\Delta \vdash_C t : \mathsf{bit} \quad \Delta \vdash_C r : A \quad \Delta \vdash_C s : A}{\Delta \vdash_C \texttt{if}\ t\ \{r \mid s\} : A}$$

Lemma VI.2. *For all classical types A, $\flat(\!|A|\!) \simeq (\!|A|\!)$.*

Proof. We proceed by structural induction on A.

- $(\!|\mathbb{U}|\!) = \mathbb{U} = \{*\} \simeq \flat\{*\} = \flat\mathbb{U}$.
- $(\!|A \to B|\!) = (\!|A|\!) \to (\!|B|\!) \simeq \flat((\!|A|\!) \to (\!|B|\!))$ by rule (FlatPureArrow).
- $(\!|A \times B|\!) = (\!|A|\!) \times (\!|B|\!) \simeq \flat(\!|A|\!) \times \flat(\!|B|\!) \simeq \flat((\!|A|\!) \times (\!|B|\!))$, using the induction hypothesis and rules (ProdMono) and (FlatProd).
- $(\!|\mathsf{bit}|\!) = \mathbb{B} = \mathbb{U} + \mathbb{U} = \flat\mathbb{U} + \flat\mathbb{U} = \flat(\mathbb{U} + \mathbb{U}) = \flat(\!|\mathsf{bit}|\!)$ using rules (SumMono) and (FlatSum).
- $(\!|A_Q \multimap B_Q|\!) = \mathbb{U} \to ((\!|A_Q|\!) \Rightarrow (\!|B_Q|\!)) \simeq \flat(\mathbb{U} \to ((\!|A_Q|\!) \Rightarrow (\!|B_Q|\!)))$ by rule (FlatPureArrow). □

Lemma VI.3. *For all qbit types A_Q, $\sharp(\!|A_Q|\!) \simeq (\!|A_Q|\!)$.*

Proof. First notice that for any A from the unitary linear algebraic lambda-calculs, we have $\sharp A \simeq \sharp\sharp A$. Indeed, by rule (SharpIntro) $\sharp A \leq \sharp\sharp A$, and by rules (SubRefl) and (SharpLift), $\sharp\sharp A \leq \sharp A$. Now we proceed by structural induction on A_Q.

- $(\!|\mathsf{qbit}|\!) = \sharp\mathbb{B} \simeq \sharp\sharp\mathbb{B} = \sharp(\!|\mathsf{qbit}|\!)$.
- $(\!|A_Q \otimes B_Q|\!) = \sharp((\!|A_Q|\!) \otimes (\!|B_Q|\!)) \simeq \sharp\sharp((\!|A_Q|\!) \otimes (\!|B_Q|\!)) = \sharp(\!|A_Q \otimes B_Q|\!)$. □

Theorem VI.5. *Translation preserves typeability:*

1) *If $\Gamma \vdash_Q t : A_Q$ then $(\!|\Gamma|\!) \vdash (\!|t|\!) : (\!|A_Q|\!)$.*
2) *If $\Delta|\Gamma \vdash_C t : A$ then $(\!|\Delta|\!), (\!|\Gamma|\!) \vdash (\!|t|\!) : (\!|A|\!)$.*
3) *If $[Q, L, t] : A$ then $\vdash (\!|[Q, L, t]|\!) : (\!|A|\!)$.*

Proof. Since $\vdash_Q$ depends on $\vdash_C$, we prove items (1) and (2) at the same time by induction on the typing derivation.

- $\dfrac{}{\Delta, x : A \vdash_C x : A}$
  By Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rules (Axiom) and (Weak), we have $(\!|\Delta|\!), x : (\!|A|\!) \vdash x : (\!|A|\!)$.

- $\dfrac{}{\Delta \vdash_C * : \mathbb{U}}$
  By Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rules (Void) and (Weak) we conclude $(\!|\Delta|\!) \vdash * : \mathbb{U}$.

- $\dfrac{\Delta, x : A \vdash_C t : B}{\Delta \vdash_C \lambda x.t : A \to B}$
  By the induction hypothesis, $(\!|\Delta|\!), x : (\!|A|\!) \vdash (\!|t|\!) : (\!|B|\!)$ and by Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rule (PureLam), $(\!|\Delta|\!) \vdash \lambda x.(\!|t|\!) : (\!|A|\!) \to (\!|B|\!)$.

- $\dfrac{\Delta \vdash_C t : A \to B \quad \Delta \vdash_C r : A}{\Delta \vdash_C tr : B}$

By the induction hypothesis, $(\!|\Delta|\!) \vdash (\!|t|\!) : (\!|A|\!) \to (\!|B|\!)$ and $(\!|\Delta|\!) \vdash (\!|r|\!) : (\!|A|\!)$. Hence, by rules (SubArrows) and (Sub), we have $(\!|\Delta|\!) \vdash (\!|t|\!) : (\!|A|\!) \Rightarrow (\!|B|\!)$, and also, we have $(\!|\Delta|\!)[\sigma] \vdash (\!|r|\!)[\sigma] : (\!|A|\!)$, where σ is a substitution of every variable in Δ by fresh variables. Then, by rule (App) we can derive, $(\!|\Delta|\!), (\!|\Delta|\!)[\sigma] \vdash (\!|t|\!)(\!|r|\!)[\sigma] : (\!|B|\!)$. By Lemma VI.2, we have $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rule (Contr), we get $(\!|\Delta|\!) \vdash (\!|t|\!)(\!|r|\!) : (\!|B|\!)$.

- $\dfrac{\Delta \vdash_C t : A \quad \Delta \vdash_C r : B}{\Delta, \Delta \vdash_C (t, r) : A \times B}$
By the induction hypothesis, $(\!|\Delta|\!) \vdash (\!|t|\!) : (\!|A|\!)$ and $(\!|\Delta|\!) \vdash (\!|r|\!) : (\!|B|\!)$. Hence, by rule (Pair), $(\!|\Delta|\!), (\!|\Delta|\!) \vdash ((\!|t|\!), (\!|r|\!)) : (\!|A|\!) \times (\!|B|\!)$.

- $\dfrac{\Delta \vdash_C t : A_1 \times A_2}{\Delta \vdash_C \pi_i t : A_i}$
By the induction hypothesis, $(\!|\Delta|\!) \vdash (\!|t|\!) : (\!|A_1|\!) \times (\!|A_2|\!)$. By Lemma VI.2, $(\!|A_i|\!) \simeq \flat(\!|A_i|\!)$ for $i = 1, 2$, hence, by rules (Axiom) and (Weak), we have $x_1 : (\!|A_1|\!), x_2 : (\!|A_2|\!) \vdash x_i : (\!|A_i|\!)$. Therefore, by rule (LetPair), we can derive $(\!|\Delta|\!) \vdash \mathsf{let}\ (x_1, x_2) = (\!|t|\!)\ \mathsf{in}\ x_i : (\!|A_i|\!)$.

- $\dfrac{}{\Delta \vdash_C \mathtt{tt} : \mathsf{bit}}$
By Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, so, by rules (Void), (InL), and (Weak), we can derive $(\!|\Delta|\!) \vdash \mathtt{tt} : \mathbb{B}$.

- $\dfrac{}{\Delta \vdash_C \mathtt{ff} : \mathsf{bit}}$
By Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, so, by rules (Void), (InR), and (Weak), we can derive $(\!|\Delta|\!) \vdash \mathtt{ff} : \mathbb{B}$.

- $\dfrac{\Delta \vdash_C t : \mathsf{bit} \quad \Delta \vdash_C r_1 : A \quad \Delta \vdash_C r_2 : A}{\Delta \vdash_C \mathtt{if}\ t\ \{r_1 \mid r_2\} : A}$
By the induction hypothesis, $(\!|\Delta|\!) \vdash (\!|t|\!) : \mathbb{B} = \mathbb{U} + \mathbb{U}$ and for $i = 1, 2$, $(\!|\Delta|\!) \vdash (\!|r_i|\!) : (\!|A|\!)$. By rules (Axiom) and (Seq), we can derive $(\!|\Delta|\!), x_i : \mathbb{U} \vdash x_i; (\!|r_i|\!) : (\!|A|\!)$ we also have $(\!|\Delta|\!)[\sigma] \vdash (\!|t|\!)[\sigma] : \mathbb{U} + \mathbb{U}$, where σ is a substitution of every variable in Δ by fresh variables. Then, by rule (PureMatch), $(\!|\Delta|\!), (\!|\Delta|\!)[\sigma] \vdash \mathtt{match}\ (\!|t|\!)[\sigma]\ \{\mathtt{inl}(x_1) \mapsto x_1; (\!|r|\!) \mid \mathtt{inr}(x_2) \mapsto x_2; (\!|s|\!)\} : (\!|A|\!)$. By Lemma VI.2, we have $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rule (Cont), we conclude $(\!|\Delta|\!) \vdash \mathtt{match}\ (\!|t|\!)\ \{\mathtt{inl}(x_1) \mapsto x_1; (\!|r|\!) \mid \mathtt{inr}(x_2) \mapsto x_2; (\!|s|\!)\} : (\!|A|\!)$

- $\dfrac{}{\Delta | x : A_Q \vdash x : A_Q}$
By Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rules (Axiom) and (Weak), we have $(\!|\Delta|\!), x : (\!|A|\!) \vdash x : (\!|A|\!)$.

- $\dfrac{\Delta | \Gamma_1 \vdash_Q s : A_Q \quad \Delta | \Gamma_2 \vdash_Q t : B_Q}{\Delta | \Gamma_1, \Gamma_2 \vdash_Q s \otimes t : A_Q \otimes B_Q}$
By the induction hypothesis, $(\!|\Delta|\!), (\!|\Gamma_1|\!) \vdash (\!|s|\!) : (\!|A_Q|\!)$ and $(\!|\Delta|\!), (\!|\Gamma_2|\!) \vdash t : (\!|B_Q|\!)$. Then, we can derive $(\!|\Delta|\!)[\sigma], (\!|\Gamma_1|\!) \vdash (\!|s|\!)[\sigma] : (\!|A_Q|\!)$, where σ is a substitution on every variable in Δ by fresh variables. Hence, by rule (Pair), we can derive $(\!|\Delta|\!)[\sigma], (\!|\Gamma_1|\!), (\!|\Delta|\!), (\!|\Gamma_2|\!) \vdash ((\!|s|\!)[\sigma], (\!|t|\!)) : (\!|A_Q|\!) \times (\!|B_Q|\!)$. By Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rule (Contr), we have $(\!|\Delta|\!), (\!|\Gamma_1|\!), (\!|\Gamma_2|\!) \vdash ((\!|s|\!), (\!|t|\!)) : (\!|A_Q|\!) \times (\!|B_Q|\!)$. Finally, by rules (SharpIntro) and (Sub), we have $(\!|\Delta|\!), (\!|\Gamma_1|\!), (\!|\Gamma_2|\!) \vdash ((\!|s|\!), (\!|t|\!)) : (\!|A_Q|\!) \otimes (\!|B_Q|\!)$.

- $\dfrac{\Delta | \Gamma \vdash_Q t : \mathsf{qbit}}{\Delta | \Gamma \vdash_Q U(t) : \mathsf{qbit}}$
By the induction hypothesis, $(\!|\Delta|\!), (\!|\Gamma|\!) \vdash (\!|t|\!) : \sharp\mathbb{B}$. By Proposition IV.11, $\vdash \bar{U} : \sharp\mathbb{B} \to \sharp\mathbb{B}$, hence, by rules (SubArrows) and (Sub), we have $\vdash \bar{U} : \sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$. Therefore, by rule (App), we can derive $(\!|\Delta|\!), (\!|\Gamma|\!) \vdash \bar{U}(\!|t|\!) : \sharp\mathbb{B}$.

- $\dfrac{\Delta | \Gamma_1 \vdash_Q s : A_Q \otimes B_Q \quad \Delta | \Gamma_2, x : A_Q, y : B_Q \vdash_Q t : C_Q}{\Delta | \Gamma_1, \Gamma_2 \vdash_Q \mathsf{let}\ x \otimes y = s\ \mathsf{in}\ t : C_Q}$
By the induction hypothesis, $(\!|\Delta|\!), (\!|\Gamma_1|\!) \vdash (\!|s|\!) : (\!|A_Q|\!) \otimes (\!|B_Q|\!)$ and $(\!|\Delta|\!), (\!|\Gamma_2|\!), x : (\!|A_Q|\!), y : (\!|B_Q|\!) \vdash (\!|t|\!) : (\!|C_Q|\!)$. Then, we also have $(\!|\Delta|\!)[\sigma], (\!|\Gamma_1|\!) \vdash (\!|s|\!)[\sigma] : (\!|A_Q|\!) \otimes (\!|B_Q|\!)$, where σ is a substitution on every variable in Δ by fresh variables. By Lemma VI.3, $(\!|A_Q|\!) \simeq \sharp(\!|A_Q|\!)$, $(\!|B_Q|\!) \simeq \sharp(\!|B_Q|\!)$, and $(\!|C_Q|\!) \simeq \sharp(\!|C_Q|\!)$. Hence, $(\!|\Delta|\!), (\!|\Gamma_2|\!), x : \sharp(\!|A_Q|\!), y : \sharp(\!|B_Q|\!) \vdash (\!|t|\!) : \sharp(\!|C_Q|\!)$. Therefore, by rule (LetTens), $(\!|\Delta|\!)[\sigma], (\!|\Gamma_1|\!), (\!|\Delta|\!), (\!|\Gamma_2|\!) \vdash \mathsf{let}\ (x, y) = (\!|s|\!)[\sigma]\ \mathsf{in}\ (\!|t|\!) : \sharp(\!|C_Q|\!)$. By Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rule (Contr), we get $(\!|\Delta|\!), (\!|\Gamma_1|\!), (\!|\Gamma_2|\!) \vdash \mathsf{let}\ (x, y) = (\!|s|\!)\ \mathsf{in}\ (\!|t|\!) : \sharp(\!|C_Q|\!)$. Finally, using the fact that $\sharp(\!|C_Q|\!) \simeq (\!|C_Q|\!)$, we get $(\!|\Delta|\!), (\!|\Gamma_1|\!), (\!|\Gamma_2|\!) \vdash \mathsf{let}\ (x, y) = (\!|s|\!)\ \mathsf{in}\ (\!|t|\!) : (\!|C_Q|\!)$.
Notice that we have used the following unproved rule: If $\Gamma, x : A \vdash t : B$ and $A \simeq C$, then $\Gamma, x : C \vdash t : B$. Hence, we prove that this rule is true. Assume $\Gamma, x : A \vdash t : B$, then, $t\langle\sigma\rangle \Vdash [\![B]\!]$ for every $\sigma \in [\![\Gamma, x : A]\!] = [\![\Gamma, x : C]\!]$, and so $\Gamma, x : C \vdash t : B$.

- $\dfrac{\Delta \vdash_C t : \mathsf{bit}}{\Delta | \emptyset \vdash_Q \mathsf{new}(t) : \mathsf{qbit}}$

By the induction hypothesis, $(\!|\Delta|\!) \vdash (\!|t|\!) : \mathbb{B}$. We conclude by rules (SharpIntro) and (Sub) that $(\!|\Delta|\!) \vdash (\!|t|\!) : \sharp\mathbb{B}$.

- $\dfrac{\Delta | x : A_Q \vdash_Q t : B_Q}{\Delta \vdash_C \lambda^Q x.t : A_Q \multimap B_Q}$
  By the induction hypothesis $(\!|\Delta|\!), x : (\!|A_Q|\!) \vdash (\!|t|\!) : (\!|B_Q|\!)$. Since $\mathbb{U} \simeq \flat\mathbb{U}$, by rule (Weak), we have $(\!|\Delta|\!), z : \mathbb{U}, x : (\!|A_Q|\!) \vdash (\!|t|\!) : (\!|B_Q|\!)$ Then, by rules (UnitLam) and (PureLam), we can derive $(\!|\Delta|\!) \vdash \lambda zx.(\!|t|\!) : \mathbb{U} \to ((\!|A_Q|\!) \Rightarrow (\!|B_Q|\!))$.

- $\dfrac{\Delta \vdash_C s : A_Q \multimap B_Q \quad \Delta | \Gamma \vdash_Q t : A_Q}{\Delta | \Gamma \vdash_Q s@t : B_Q}$
  By the induction hypothesis, $(\!|\Delta|\!) \vdash (\!|s|\!) : \mathbb{U} \to ((\!|A_Q|\!) \Rightarrow (\!|B_Q|\!))$ and $(\!|\Delta|\!), (\!|\Gamma|\!) \vdash (\!|t|\!) : (\!|A_Q|\!)$. Then, $(\!|\Delta|\!)[\sigma], (\!|\Gamma|\!) \vdash (\!|t|\!)[\sigma] : (\!|A_Q|\!)$, where σ is a substitution on every variable in Δ by fresh variables. By rules (SubArrows) and (Sub), we have $(\!|\Delta|\!) \vdash (\!|s|\!) : \mathbb{U} \Rightarrow ((\!|A_Q|\!) \Rightarrow (\!|B_Q|\!))$. In addition, by rule (Void), $\vdash * : \mathbb{U}$. Hence, by rule (App) twice, we get $(\!|\Delta|\!), (\!|\Gamma|\!), (\!|\Delta|\!)[\sigma] \vdash ((\!|s|\!)*)(\!|t|\!)[\sigma] : (\!|B_Q|\!)$. By Lemma VI.2, $\flat(\!|\Delta|\!) \simeq (\!|\Delta|\!)$, hence, by rule (Contr), $(\!|\Delta|\!), (\!|\Gamma|\!) \vdash ((\!|s|\!)*)(\!|t|\!) : (\!|B_Q|\!)$.

Now we prove item (3).

Let

$$[\sum_{i=1}^{m} \alpha_i \cdot |y_1^i, \ldots, y_n^i\rangle, \{x_1 := p(1), \ldots, x_n := p(n)\}, t] : A_Q$$

that means $\emptyset | FV(t) : \mathsf{qbit} \vdash_Q t : A_Q$. We must show that

$$\vdash (\!|[\sum_{i=1}^{m} \alpha_i \cdot |y_1^i, \ldots, y_n^i\rangle, \{x_1 := p(1), \ldots, x_n := p(n)\}, t]|\!) : (\!|A|\!)$$

that is

$$\vdash \sum_{i=1}^{m} \alpha_i \cdot (\!|t|\!)[x_1 := \bar{y}^i_{p(1)}, \ldots, x_n := \bar{y}^i_{p(n)}] : (\!|A_Q|\!) \tag{1}$$

From item (1) we have $FV(t) : \sharp\mathbb{B} \vdash (\!|t|\!) : (\!|A_Q|\!)$. Then, by definition, we have $(\!|t|\!)\langle\sigma\rangle \Vdash (\!|A_Q|\!)$ for every $\sigma \in [\![FV(t) : \sharp\mathbb{B}]\!]$. In particular, $[\sigma_i] = [x_1 := \bar{y}^i_{p(1)}, \ldots, x_n := \bar{y}^i_{p(n)}] \in [\![FV(t) : \sharp\mathbb{B}]\!]$, so $(\!|t|\!)\langle\sigma_i\rangle = (\!|t|\!)[\sigma_i] \Vdash (\!|A_Q|\!)$. By Lemma VI.3, $(\!|A_Q|\!) \simeq \sharp(\!|A_Q|\!)$, and so, we have $\sum_{i=1}^{m} \alpha_i \cdot (\!|t|\!)[\sigma_i] \Vdash (\!|A_Q|\!)$, which is, by definition, the same as (1) □

Lemma A.12. *For any terms t and r, $(\!|t[x := r]|\!) = (\!|t|\!)[x := (\!|r|\!)]$.*

Proof. By a straightforward structural induction on t. □

Lemma A.13. *For all value distributions $\vec{v}$ and $\vec{v}$, for all term distributions $\vec{t}$, $\vec{s}$, $\vec{s}_1$, $\vec{s}_2$ and for all pure values w, we have the equalities:*

- $(\vec{v}, \vec{v}')[x := w] = (\vec{v}[x := w], \vec{v}'[x := w])$
- $\mathtt{inl}(\vec{v})[x := w] = \mathtt{inl}(\vec{v}[x := w])$
- $\mathtt{inr}(\vec{v})[x := w] = Inr\vec{v}[x := w]$
- $(\vec{s}\,\vec{t})[x := w] = \vec{s}[x := w]\,\vec{t}[x := w]$
- $(\vec{t}; \vec{s})[x := w] = \vec{t}[x := w]; \vec{s}[x := w]$
- $(\mathtt{let}\ (x_1, x_2) = \vec{t}\ \mathtt{in}\ \vec{s})[x := w] = \mathtt{let}\ (x_1, x_2) = \vec{t}[x := w]\ \mathtt{in}\ \vec{s}[x := w]$ *(if $x_1, x_2 \notin FV(w) \cup \{x\}$)*
- $(\mathtt{match}\ \vec{t}\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1 \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2\})[x := w] = \mathtt{match}\ \vec{t}[x := w]\ \{\mathtt{inl}(x_1) \mapsto \vec{s}_1[x := w] \mid \mathtt{inr}(x_2) \mapsto \vec{s}_2[x := w]\}$

Proof. Let us treat the case of the pair destructing let-construct. Given term distributions $\vec{t} = \sum_{i=1}^{n} \alpha_i \cdot t_i$ and $\vec{s}$, and a pure value w such that $x_1, x_2 \notin FV(w) \cup \{x\}$, we observe that

$$\begin{array}{lll}
 & (\mathtt{let}\ (x_1, x_2) = \vec{t}\ \mathtt{in}\ \vec{s})[x := w] & \\
= & (\sum_{i=1}^{n} \alpha_i \cdot \mathtt{let}\ (x_1, x_2) = t_i\ \mathtt{in}\ \vec{s})[x := w] & \text{(def. of extended let)} \\
= & \sum_{i=1}^{n} \alpha_i \cdot (\mathtt{let}\ (x_1, x_2) = t_i\ \mathtt{in}\ \vec{s})[x := w] & \text{(linearity of pure substitution)} \\
= & \sum_{i=1}^{n} \alpha_i \cdot \mathtt{let}\ (x_1, x_2) = t_i[x := w]\ \mathtt{in}\ \vec{s}[x := w] & \text{(pure substitution in a let-construct)} \\
= & \mathtt{let}\ (x_1, x_2) = (\sum_{i=1}^{n} \alpha_i \cdot \vec{t_i}[x := w])\ \mathtt{in}\ \vec{s}[x := w] & \text{(def. of extended let)} \\
= & \mathtt{let}\ (x_1, x_2) = \vec{t}[x := w]\ \mathtt{in}\ \vec{s}[x := w] & \text{(linearity of pure substitution)}
\end{array}$$

The other cases are treated similarly. □

Remark A.14 (Parallel substitution)**.** *The operation of parallel substitution* $[x_1 := w_1, \dots, x_n := w_n]$ *(where* $x_1, \dots, x_n$ *are pairwise distinct variables) can be easily implemented as a sequence of pure substitutions, by temporarily replacing the* x_i*'s with fresh names in order to avoid undesirable captures between successive pure substitutions. For instance, we can let*

$$\begin{aligned}&\vec{t}[x_1 := w_1, \dots, x_n := w_n] \ :=\\ &\vec{t}[x_1 := z_1] \cdots [x_n := z_n][z_1 := w_1] \cdots [z_n := w_n]\end{aligned}$$

where $z_1, \dots, z_n$ *are fresh names w.r.t.* $\vec{t}, x_1, \dots, x_n, w_1, \dots, w_n$*. Note that this precaution is useless when the substituands* $w_1, \dots, w_n$ *are closed, since in this case, parallel substitution amounts to the following sequential substitution (whose order is irrelevant):*

$$\vec{t}[x_1 := w_1, \dots, x_n := w_n] \ = \ \vec{t}[x_1 := w_1] \cdots [x_n := w_n]\,.$$

Lemma A.15. *For all term distributions* $\vec{t}$ *and for all closed value distributions* $\vec{v}$ *and* $\vec{w}$*:*

$$\vec{t}\langle x := \vec{v}\rangle\langle y := \vec{w}\rangle \ = \ \vec{t}\langle y := \vec{w}\rangle\langle x := \vec{v}\rangle \qquad (\textit{provided } x \neq y) \quad \square$$

Theorem VI.6 (Adequacy)**.** *If* $[Q, L, t] \to [Q', L', r]$*, then* $(\!|[Q, L, t]|\!) \succ\!\!\succ (\!|[Q', L', r]|\!)$*.*

Proof. We proceed by induction on the rewrite relation of λ_Q. We only give the cases where $C(\cdot) = \{\cdot\}$, as other cases are simple calls to the induction hypothesis. In all the cases, we consider $Q = \sum_{i=1}^m \alpha_i |u_1^i, \dots, y_n^i\rangle$, $L = \{x_1 := p(1), \dots, x_n := p(n)\}$, and $[\sigma_i] = [x_1 := \vec{y}^i_{p(1)}, \dots, x_n := \vec{y}^i_{p(n)}]$.

- $[Q, L, (\lambda x.t)u] \to [Q, L, t[x := u]]$.

$$\begin{aligned}(\!|[Q, L, (\lambda x.t)u]|\!) &= \textstyle\sum_{i=1}^m \alpha_i \cdot ((\lambda x.(\!|t|\!))(\!|u|\!))[\sigma_i] \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot ((\lambda x.(\!|t|\!)[\sigma_i])(\!|u|\!)[\sigma_i]) && \text{(Lemma A.13)} \\ &\succ\!\!\succ \textstyle\sum_{i=1}^m \alpha_i \cdot (\!|t|\!)[\sigma_i][x := (\!|u|\!)[\sigma_i]] \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\!|t|\!)[x := (\!|u|\!)][\sigma_i] && \text{(Lemma A.15)} \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\!|t[x := u]|\!)[\sigma_i] && \text{(Lemma A.12)} \\ &= (\!|[Q, L, t[x := u]]|\!)\end{aligned}$$

- $[Q, L, (\lambda^Q x.t)@u] \to [Q, L, t[x := u]]$.

$$\begin{aligned}(\!|[Q, L, (\lambda^Q x.t)@u]|\!) &= \textstyle\sum_{i=1}^m \alpha_i \cdot (((\lambda z x.(\!|t|\!))*)(\!|u|\!))[\sigma_i] \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot (((\lambda z x.(\!|t|\!)[\sigma_i])*)(\!|u|\!)[\sigma_i]) && \text{(Lemma A.13)} \\ &\succ\!\!\succ \textstyle\sum_{i=1}^m \alpha_i \cdot ((\lambda x.(\!|t|\!)[\sigma_i])(\!|u|\!)[\sigma_i]) \\ &\succ\!\!\succ \textstyle\sum_{i=1}^m \alpha_i \cdot (\!|t|\!)[\sigma_i][x := (\!|u|\!)[\sigma_i]] \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\!|t|\!)[x := (\!|u|\!)][\sigma_i] && \text{(Lemma A.15)} \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\!|t[x := u]|\!)[\sigma_i] && \text{(Lemma A.12)} \\ &= (\!|[Q, L, t[x := u]]|\!)\end{aligned}$$

- $[Q, L, \pi_1(u, v)] \to [Q, L, u]$.

$$\begin{aligned}(\!|[Q, L, \pi_1(u, v)]|\!) &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\mathsf{let}\ (x, y) = ((\!|u|\!), (\!|v|\!))\ \mathsf{in}\ x)[\sigma_i] \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\mathsf{let}\ (x, y) = ((\!|u|\!)[\sigma_i], (\!|v|\!)[\sigma_i])\ \mathsf{in}\ x) && \text{(Lemma A.13)} \\ &\succ\!\!\succ \textstyle\sum_{i=1}^m \alpha_i \cdot (\!|u|\!)[\sigma_i] \\ &= (\!|[Q, L, u]|\!)\end{aligned}$$

- $[Q, L, \pi_s(u, v)] \to [Q, L, v]$.

$$\begin{aligned}(\!|[Q, L, \pi_2(u, v)]|\!) &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\mathsf{let}\ (x, y) = ((\!|u|\!), (\!|v|\!))\ \mathsf{in}\ y)[\sigma_i] \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\mathsf{let}\ (x, y) = ((\!|u|\!)[\sigma_i], (\!|v|\!)[\sigma_i])\ \mathsf{in}\ y) && \text{(Lemma A.13)} \\ &\succ\!\!\succ \textstyle\sum_{i=1}^m \alpha_i \cdot (\!|v|\!)[\sigma_i] \\ &= (\!|[Q, L, v]|\!)\end{aligned}$$

- $[Q, L, \mathtt{if}\ \mathtt{tt}\ \{t \mid r\}] \to [Q, L, t]$

$$\begin{aligned}&(\!|[Q, L, \mathtt{if}\ \mathtt{tt}\ \{t \mid r\}]|\!) \\ &= \textstyle\sum_{i=1}^m \alpha_i \cdot (\mathtt{match}\ \mathtt{inl}(*)\ \{\mathtt{inl}(z_1) \mapsto z_1; (\!|t|\!) \mid \mathtt{inr}(z_2) \mapsto z_2; (\!|r|\!)\})[\sigma_i]\end{aligned}$$

$= \sum_{i=1}^m \alpha_i \cdot \texttt{match inl}(*)\ \{\texttt{inl}(z_1) \mapsto z_1; (\!|t|\!)[\sigma_i] \mid \texttt{inr}(z_2) \mapsto z_2; (\!|r|\!)[\sigma_i]\}$ (Lemma A.13)
$\succ\!\!\succ \sum_{i=1}^m \alpha_i \cdot *; (\!|t|\!)[\sigma_i]$
$\succ\!\!\succ \sum_{i=1}^m \alpha_i \cdot (\!|t|\!)[\sigma_i]$
$= (\!|[Q, L, t]|\!)$

- $[Q, L, \texttt{if ff } \{t \mid r\}] \to [Q, L, r]$

$(\!|[Q, L, \texttt{if ff } \{t \mid r\}]|\!)$
$= \sum_{i=1}^m \alpha_i \cdot (\texttt{match inr}(*)\ \{\texttt{inl}(z_1) \mapsto z_1; (\!|t|\!) \mid \texttt{inr}(z_2) \mapsto z_2; (\!|r|\!)\})[\sigma_i]$
$= \sum_{i=1}^m \alpha_i \cdot \texttt{match inr}(*)\ \{\texttt{inl}(z_1) \mapsto z_1; (\!|t|\!)[\sigma_i] \mid \texttt{inr}(z_2) \mapsto z_2; (\!|r|\!)[\sigma_i]\}$ (Lemma A.13)
$\succ\!\!\succ \sum_{i=1}^m \alpha_i \cdot *; (\!|r|\!)[\sigma_i]$
$\succ\!\!\succ \sum_{i=1}^m \alpha_i \cdot (\!|r|\!)[\sigma_i]$
$= (\!|[Q, L, r]|\!)$

- $[Q, L, \mathsf{let}\ x \otimes y = t \otimes r\ \mathsf{in}\ s] \to [Q, L, s[x := t, y := r]]$.

$(\!|[Q, L, \mathsf{let}\ x \otimes y = t \otimes r\ \mathsf{in}\ s]|\!)$
$= \sum_{i=1}^m \alpha_i \cdot (\mathsf{let}\ (x, y) = ((\!|t|\!), (\!|r|\!))\ \mathsf{in}\ (\!|s|\!))[\sigma_i]$
$= \sum_{i=1}^m \alpha_i \cdot (\mathsf{let}\ (x, y) = ((\!|t|\!)[\sigma_i], (\!|r|\!)[\sigma_i])\ \mathsf{in}\ (\!|s|\!)[\sigma_i])$ (Lemma A.13)
$\succ\!\!\succ \sum_{i=1}^m \alpha_i \cdot (\!|s|\!)[\sigma_i][x := (\!|t|\!)[\sigma_i]][y := (\!|r|\!)[\sigma_i]]$
$= \sum_{i=1}^m \alpha_i \cdot ((\!|s|\!)[x := (\!|t|\!)][y := (\!|r|\!)])[\sigma_i]$ (Lemma A.15)
$= \sum_{i=1}^m \alpha_i \cdot ((\!|s[x := t, y := r]|\!))[\sigma_i]$ (Lemmas A.12 and Remark A.14)
$= (\!|[Q, L, s[x := t, y := r]]|\!)$

- $[\emptyset, \emptyset, \mathsf{new}(\texttt{tt})] \to [|1\rangle, \{x \mapsto 1\}, x]$

$$(\!|[\emptyset, \emptyset, \mathsf{new}(\texttt{tt})]|\!) = (\!|\mathsf{new}(\texttt{tt}))|\!) = \texttt{tt} = x[x := \texttt{tt}] = (\!|[|1\rangle, \{x \mapsto 1\}, x]|\!)$$

- $[\emptyset, \emptyset, \mathsf{new}(\texttt{ff})] \to [|0\rangle, \{x \mapsto 1\}, x]$

$$(\!|[\emptyset, \emptyset, \mathsf{new}(\texttt{ff})]|\!) = (\!|\mathsf{new}(\texttt{ff})|\!) = \texttt{ff} = x[x := \texttt{ff}] = (\!|[|0\rangle, \{x \mapsto 1\}, x]|\!)$$

- $[|\psi\rangle, \{x \mapsto 1\}, U(x)] \to [U|\psi\rangle, \{x \mapsto 1\}, x]$.
Let $U|0\rangle = \gamma_0|0\rangle + \delta_0|1\rangle$ and $U|1\rangle = \gamma_1|0\rangle + \delta_1|1\rangle$. Then,

$$\begin{aligned}
(\!|[\alpha|0\rangle + \beta|1\rangle, \{x \mapsto 1\}, U(x)]|\!) &= \alpha \cdot (\!|U(x)|\!)[x := \texttt{tt}] + \beta \cdot (\!|U(x)|\!)[x := \texttt{ff}] \\
&= \alpha \cdot \bar{U}\texttt{tt} + \beta \cdot \bar{U}\texttt{ff} \\
&\succ\!\!\succ \alpha \cdot (\gamma_0 \cdot \texttt{tt} + \delta_0 \cdot \texttt{ff}) + \beta \cdot (\gamma_1 \cdot \texttt{tt} + \delta_1 \cdot \texttt{ff}) \\
&= (\alpha\gamma_0 + \beta\gamma_1) \cdot \texttt{tt} + (\alpha\delta_0 + \beta\delta_1) \cdot \texttt{ff} \\
&= (\alpha\gamma_0 + \beta\gamma_1) \cdot x[x := \texttt{tt}] + (\alpha\delta_0 + \beta\delta_1) \cdot x[x := \texttt{ff}] \\
&= (\!|[(\alpha\gamma_0 + \beta\gamma_1)|0\rangle + (\alpha\delta_0 + \beta\delta_1)|1\rangle, \{x \mapsto 1\}, x]|\!)
\end{aligned}$$

□