Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Titulo de la tesis en español

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área de Ciencias de la Computación

Nombre y Apellido de quien escribe esta tesis

Directora/s de tesis:	Dra. Nombre y Apellido
Directora Asistente:	Dra. Nombre y Apellido
Consejera de estudios:	Dra. Nombre y Apellido
Lugar de trabajo:	Departamento de Computación Facultad de Cs. Exactas y Naturales Universidad de Buenos Aires

Buenos Aires, 2018
Fecha de defensa: 18 de Abril de 2018

Titulo de la tesis en español

Resumen:

Palabras clave: palabra1, palabra2, palabra3, palabra4, palabra5

Título de la tesis en inglés

Abstract:

Keywords: keyword1,keyword2,keyword3,keyword4

Agradecimientos

Contents

Chapter 1

Basis-Sensitive Quantum Typing via Realizability

1.1 Introduction

We previously presented the impossibility theorems which stated that is physically impossible to copy or delete a qubit. There is however, a subtlety in these impossibility theorems. Arbitrary qubits cannot be copied, but it is indeed possible to do so with known qubits. This implies that qubits with known values behave as classical data and can be treated accordingly. Moreover, it suffices to know the basis to which a qubit belongs in order to copy and delete it. This is a known fact in quantum information theory which underlies a number of quantum algorithms.

In most quantum programming languages, qubits are interpreted in a canonical basis (often called the computational basis); see, for instance (CITAR EJEMPLOS). In this fashion, classical bits are represented by the basis vectors, and qubits as norm-1 linear combinations of bits. We are allowed to copy and delete classical bits freely, while such operations on arbitrary qubits remain restricted.

In this chapter we will introduce a quantum lambda calculus in the quantum-data / quantum-control paradigm. It uses as starting point the calculus defined in [1], which was introduced using a realizability technique. In the same manner, our aim is to follow this workflow to extract a type system able to track bases throughout the programs. This should allow us to treat qubit in known bases classically, while still handling unknown qubits linearly.

Realizability is a technique for extracting type systems from the operational semantics of a calculus, resulting in a system in which safety properties hold by construction.

The steps to define a programming language using this technique are as follows. First, define a calculus equipped with a deterministic evaluation strategy. Second, define types as sets of closed values in the language, optionally introducing operations to build more complex types. Third, define the typing judgement $\Gamma \vdash t : A$, where Γ is a context of typed variables, t a term in the calculus, and A a type, as the property that for every valid substitution θ of Γ, the term $\theta(t)$ reduces to a value in A, i.e., $\theta(t) \twoheadrightarrow v \in A$.

$$
\begin{array}{rcll}
v & ::= & x \mid \lambda x_B \,.\, \vec{t} \mid (v, v) \mid |0\rangle \mid |1\rangle & \\
t & ::= & w \mid t\, t \mid \mathsf{let}_{(B,B)}\, (x, y) = \vec{t}\ \mathsf{in}\ \vec{t} \mid & \\
 & & \mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v} \mapsto \vec{t} \mid \cdots \mid \vec{v} \mapsto \vec{t}\} & \\
\vec{v} & ::= & v \mid \vec{v} + \vec{v} \mid \alpha \cdot \vec{v} & (\alpha \in \mathbb{C}) \\
\vec{t} & ::= & t \mid \vec{t} + \vec{t} \mid \alpha \cdot \vec{t} & (\alpha \in \mathbb{C})
\end{array}
$$

Where B is an n-th dimensional orthonormal basis as defined in Def. 2.

Table 1.1: *Syntax of the calculus*

In this setting, each typing rule corresponds to a provable theorem. For instance, if $\Gamma \vdash t : A$ implies $\Delta \vdash r : B$, then the following rule is derivable:

$$\frac{\Gamma \vdash t : A}{\Delta \vdash r : B}$$

The structure of the chapter is as follows: In section 1.2, we define the syntax for the calculus. Then, in 1.3 we detail the reduction system. We define the type algebra and prove a set of valid typing rules in section 1.4. With the calculus fully defined, we showcase a few examples in section 1.5. We give closing remarks and discuss future work in 1.6.

1.2 The calculus

1.2.1 Syntax

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, which form a vector space.

The calculus is divided into four distinct syntactic categories: *pure values, pure terms, value distributions and term distributions*. Values are composed by variables, a decorated lambda abstraction and two boolean values representing perpendicular vectors: $|0\rangle$ and $|1\rangle$. A pair of values is also a value itself. Terms include values, applications, pair constructors and destructors and pattern-matching testing for orthogonal vectors represented by the case operator. Both terms and value distributions are built by a $\mathbb{C}$-linear combination of either terms or values respectively. In Table 1.2 we also include notation for ease of writing of linear distributions of pairs.

Remark 1. *We do not include a single specific term representing the null vector $\vec{0}$ since we do not make use of it. Instead, any distribution $0 \cdot \vec{t}$ will act as one.*

In order to handle the different bases in each abstraction, we need to define a congruence relationship between values. When we define the reduction system, this congruence will allow us to take an

$$
\begin{array}{rcl}
(\alpha \cdot v + \vec{v}_1, \vec{v}_2) & := & \alpha(v, \vec{v}_2) + (\vec{v}_1, \vec{v}_2) \\
(\vec{v}_1, \alpha \cdot v + \vec{v}_2) & := & \alpha(\vec{v}_1, v) + (\vec{v}_1, \vec{v}_2)
\end{array}
$$

Table 1.2: *Notation for writing pair distributions*

argument and interpret it in the corresponding basis of the function. Here, the vector space of value distributions starts to take shape.

We expand on the rationale for the first rule, $\vec{v_1} + 0 \cdot \vec{v_2} \equiv \vec{v_1}$. The main idea of the calculus is to decompose the vectors corresponding to the arguments onto the bases attached to the abstractions. Taking an example from linear algebra, if we were to rewrite the vector $(1, 0)$ as a linear combination of $\{\frac{(1,1)}{\sqrt{2}}, \frac{(1,-1)}{\sqrt{2}}\}$ we would get:

$$
\begin{aligned}
(1,0) &= (1,0) + 0 \cdot (0,1) \\
&= \frac{1}{2}((1,0) + (1,0) + (0,1) - (0,1)) \\
&= \frac{1}{\sqrt{2}} \cdot \left(\frac{(1,1)}{\sqrt{2}} + \frac{(1,-1)}{\sqrt{2}} \right)
\end{aligned}
$$

If we match the vector $(1, 0)$ to $|0\rangle$ and $(0, 1)$ to $|1\rangle$, we would need a way to introduce the second coordinate into the equation. That is where the first rule comes into play. We restrict ourselves to vectors, since introducing variables or abstractions could break safety properties of the system.

We consider the congruence $\equiv$ as shallow. This means it does not act under lambda abstractions nor case alternatives. For example, if $\vec{t} \neq \vec{s}$: $(\lambda x_B \,.\, \vec{t}) \not\equiv (\lambda x_B \,.\, \vec{s})$ (Even in the case that $\vec{t} \equiv \vec{s}$).

The core mechanism of the calculus lies in decorating variable bindings with sets of value distributions. Keeping with linear algebra terminology, we will refer to these sets as *(orthonormal) bases*, for reasons which will shortly become clear. These bases will inform the reduction system on how to operate its arguments.

In order to properly characterize the sets that decorate the lambda abstractions, we first have to define which are the values that they must contain.

Definition 1. *A 1-dimensional qubit is a value distribution of the form: $\alpha\,|0\rangle + \beta\,|1\rangle$ where $|\alpha|^2 + |\beta|^2 = 1$. An n-th dimensional qubit is a value distribution of the form $\alpha(|0\rangle\,, \vec{w}_1) + \beta(|1\rangle\,, \vec{w}_2)$ where $\vec{w}_1$ and $\vec{w}_2$ are $(n-1)$ dimensional qubits and the same previous conditions apply to α and β.*

From this point forward we shall write $\vec{\mathrm{V}}$ to the space of all closed value distributions which we will call *vectors*. This space is equipped with an inner product $\langle \vec{v} \mid \vec{w} \rangle$ and a pseudo-$\ell_2 - norm$ $\|\vec{v}\|$ defined as:

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

$$\vec{v_1} + 0 \cdot \vec{v_2} \equiv \vec{v_1}$$

Where $\vec{v_i}$ is neither an abstraction, nor a variable

$$1 \cdot \vec{t} \equiv \vec{t} \qquad \alpha \cdot (\beta \cdot \vec{t}) \equiv \alpha\beta \cdot \vec{t}$$
$$\vec{t_1} + \vec{t_2} \equiv \vec{t_2} + \vec{t_1} \qquad (\vec{t_1} + \vec{t_2}) + \vec{t_3} \equiv \vec{t_1} + (\vec{t_2} + \vec{t_3})$$
$$(\alpha + \beta) \cdot \vec{t} \equiv \alpha \cdot \vec{t} + \beta \cdot \vec{t}$$
$$\alpha \cdot (\vec{t_1} + \vec{t_2}) \equiv \alpha \cdot \vec{t_1} + \alpha \cdot \vec{t_2}$$
$$\vec{t}(\alpha\vec{s}) \equiv \alpha(\vec{t}\vec{s}) \qquad (\alpha\vec{t})\vec{s} \equiv \alpha(\vec{t}\vec{s})$$
$$(\vec{t} + \vec{s})\vec{r} \equiv \vec{t}\vec{r} + \vec{s}\vec{r} \qquad \vec{t}(\vec{s} + \vec{r}) \equiv \vec{t}\vec{s} + \vec{t}\vec{r}$$
$$\mathsf{let}_{(A_1,B_2)}\ (x_1, x_2) = (\alpha\vec{t})\ \mathsf{in}\ \vec{s} \equiv \alpha(\mathsf{let}_{(A_1,B_2)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{s})$$
$$\mathsf{let}_{(A_1,B_2)}\ (x_1, x_2) = \vec{t} + \vec{s}\ \mathsf{in}\ \vec{r} \equiv (\mathsf{let}_{(A_1,B_2)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{r}) + (\mathsf{let}_{(A_1,B_2)}\ (x_1, x_2) = \vec{s}\ \mathsf{in}\ \vec{r})$$
$$\mathsf{case}\ \alpha\vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\} \equiv \alpha(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})$$
$$\mathsf{case}\ (\vec{t} + \vec{s})\ \mathsf{of}\{\vec{v} \mapsto \vec{r_1} \mid \cdots \mid \vec{w} \mapsto \vec{r_2}\} \equiv \mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v} \mapsto \vec{r_1} \mid \cdots \mid \vec{w} \mapsto \vec{r_2}\} + \mathsf{case}\ \vec{s}\ \mathsf{of}\{\vec{v} \mapsto \vec{r_1} \mid \cdots \mid \vec{w} \mapsto \vec{r_2}\}$$

Table 1.3: *Term congruence*

Where $\vec{v} = \sum_{i=1}^{n} \alpha_i \cdot v_i$ and $\vec{w} = \sum_{j=1}^{m} \beta_j \cdot w_j$, and where δ_{v_i,w_j} is the Kronecker delta such that it is 1 if $v_i = w_j$ and 0 otherwise.

With the notion of an internal product, we can finalize the details on the calculus syntax. As one might expect, we will say two values are orthogonal when their internal product equals to zero. With the previous definition we can describe the sets decorating the abstractions.

Definition 2. *We will say a set of value distributions B is an n-th dimensional orthonormal basis when it satisfies the following conditions:*

1. *Each member of B is a qubit of dimension n.*
2. *Each member has norm equal to* 1.
3. *Each member of B is pairwise orthogonal to every other member.*

Unlike the usual definition of orthonormal basis, we also need to ensure that the members are qubits. In other words, they are neither variables nor abstractions. Morally, these sets will keep track of the basis the term is working on. A qubit which is a member of this set will be treated on a call-by-value strategy and its data can be treated classically. Any other qubit will first be interpreted as a $\mathbb{C}$-linear

combination of elements of the basis and then the function will apply linearly to each component. If the argument cannot be written in the decorating basis, the evaluation gets stuck.

1.2.2 Substitutions

The beta reduction will depend on the basis chosen for the abstraction, so we have to define a new substitution which will take this mechanism into account. This operation will substitute the variables for vectors in the chosen basis. The accompanying coefficients correspond to the value distribution which is the object of the substitution.

With this substitution we also define a special kind of basis which we call $\mathcal{P}$ which will act as the canonical basis for lambda abstractions. In this way, we restrict distributions of functions to a single possible basis.

Definition 3. *For a term distribution $\vec{t}$, value distribution $\vec{v}$, variable x and orthogonal basis A, we define the substitution $\vec{t}\langle\vec{v}/x\rangle_A$ as:*

$$\vec{t}\langle\vec{v}/x\rangle_A = \begin{cases} \sum_{i\in I} \alpha_i \vec{t}\,[\vec{b_i}/x] & A = \{\vec{b_i}\}_{i\in I} \wedge \vec{v} \equiv \sum_{i\in I} \alpha_i \vec{b_i} \\ \sum_{i\in I} \alpha_i \vec{t}\,[v_i/x] & A = \mathcal{P} \wedge \vec{v} = \sum_{i\in I} \alpha_i v_i \\ \textit{Undefined} & \textit{Otherwise} \end{cases}$$

We extend the substitution for more than one pair of variables. This definition is extended to a pair of values in the following way. Let $\vec{v} = \sum_{i\in I} \alpha_i(\vec{v_i}, \vec{w_i})$:

$$\vec{t}\langle\vec{v}/x \otimes y\rangle_{A\otimes B} = \sum_{i\in I} \alpha_i \vec{t}\langle\vec{v_i}/x\rangle_A\langle\vec{w_i}/y\rangle_B$$

With this new substitution defined, we set out to prove some lemmas which will be useful later for proving the validity of some typing judgements. First, we want to show that the basis dependent substitution commutes with the linear combination of terms.

Lemma 1. $(\sum_i \alpha_i \vec{t_i})\langle\vec{v}/x\rangle_A \equiv \sum_i \alpha_i \vec{t_i}\langle\vec{v}/x\rangle_A$

Proof. Let $\vec{v} \equiv \sum_{j=0}^{n} \beta_j \vec{v_j}$

$$\begin{aligned} (\sum_i \alpha_i \vec{t})\langle\vec{v}/x\rangle_A &= \sum_{j=1}^{m} \beta_j (\sum_{i=1}^{n} \alpha_i t_i)[\vec{v}_j/x] \\ &\equiv \sum_{i=1}^{n} \alpha_i (\sum_{j=1}^{m} \beta_j t_i[\vec{v}_j/x]) \\ &= \sum_{i=1}^{n} \alpha_i \vec{t_i}\langle\vec{v}/x\rangle_A \end{aligned}$$

□

The next thing we need to show is that the substitution behaves well with respect to the term congruence previously defined. In essence, the following results states that for each member of the same equivalence class defined by $\equiv$, the result of substitution for those vectors is always syntactically the same.

Lemma 2. *If $\vec{v} \equiv \vec{w}$, then $\vec{t}\langle\vec{v}/x\rangle_A = \vec{t}\langle\vec{w}/x\rangle_A$.*

Proof. Since $\vec{v} \equiv \vec{w}$, by corollary **??**, we have that both $\vec{v}$ and $\vec{w}$ can be written as:

$$\vec{v} \equiv \vec{w} \equiv \sum_{i=1}^{n} \alpha_i \vec{a_i} \qquad \text{Where } \vec{a_i} \in A$$

Then:

$$\vec{t}\langle\vec{v}/x\rangle_A = \sum_{i=1}^{n} \alpha_i \vec{t}[\vec{a_i}/x] = \vec{t}\langle\vec{w}/x\rangle_A$$

□

Remark 2. *The result from lemma 2, does not translate across bases, so $\vec{t}\langle\vec{v}/x\rangle_A \not\equiv \vec{t}\langle\vec{v}/x\rangle_B$. From here onwards we will define $|+\rangle := \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle := \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. As well, we will define $\mathbb{B} = \{|0\rangle, |1\rangle\}$ and $\mathbb{X} = \{|+\rangle, |-\rangle\}$. With this we have:*

$$(\lambda x_C \,.\, y)\langle|+\rangle/y\rangle_{\mathbb{X}} = (\lambda x_C \,.\, |+\rangle) \not\equiv \frac{1}{\sqrt{2}}((\lambda x_C \,.\, |0\rangle) + (\lambda x_C \,.\, |1\rangle)) = (\lambda x_C \,.\, y)\langle|+\rangle/y\rangle_{\mathbb{B}}$$

This boils down to the fact that the $\equiv$-relationship does not commute, neither with the lambda abstraction nor the case construct. This is due to the fact that, despite being computationally equivalent, the terms $\lambda x_X \,.\, \sum_{i=1}^{n} \alpha_i \vec{t_i}$ and $\sum_{i=1}^{n} \alpha_i \lambda x_X \,.\, \vec{t_i}$ are not congruent (Similarly for the case construct)

We now introduce notation for generalized substitutions over a term. A substitution σ can be thought as a set of singular substitutions applied consecutively over a term. More precisely, for a term $\vec{t}$, value distributions $\vec{v_1} \cdots \vec{v_n}$, variables $x_1, \cdots, x_n$ and, bases $B_1, \cdots B_n$:

$$\vec{t}\langle\sigma\rangle := \vec{t}\langle v_1/x_1\rangle_{B_1} \cdots \langle\vec{v_n}/x_n\rangle_{B_n}$$

Since every $\vec{v_1}, \cdots, \vec{v_n}$ is closed, the order of the substitutions is irrelevant. We can think of the substitution σ as a partial function from variables to pairs of value distributions and bases. We denote $x_1, \cdots, x_n$ as the domain of σ ($\mathrm{dom}(\sigma)$). In the same way, we can extend the substitution, for a term $\vec{t}$, substitution σ, value distribution $\vec{v}$, variable x and, basis B:

$$\vec{t}\langle\sigma\rangle\langle\vec{v}/x\rangle_B = \vec{t}\langle\sigma'\rangle$$

Such that σ' behaves the same as σ and, it maps x to $\vec{v}$ in the basis B. This operation can also extend different generalized substitutions, such that $\vec{v}\langle\sigma\rangle = \vec{t}\langle\sigma_1\rangle\langle\sigma_2\rangle$.

$$\sum_{i=1}^{n} \alpha_i (\lambda x_A \,.\, \vec{t_i})\ \vec{v} \to \sum_{i=1}^{n} \alpha_i \vec{t_i} \langle \vec{v}/x \rangle_A$$

$$\text{If } \vec{t_i}\langle \vec{v}/x \rangle_A \text{ is defined}$$

$$\mathsf{let}_{(B,B')}\ (x,y) = \vec{v}\ \mathsf{in}\ \vec{t} \to \vec{t}\langle \vec{v}/x \otimes y \rangle_{B \otimes B'}$$

$$\mathsf{case}\ \vec{v}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{t_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{t_2}\} \to \sum_{i=1}^{n} \alpha_i \vec{t_i}$$

$$\text{If } \vec{v} \equiv \sum_{i=1}^{n} \alpha_i \vec{v_i}$$

$$\frac{t \longrightarrow \vec{r}}{s\,t \longrightarrow s\,\vec{r}} \qquad \frac{t \longrightarrow r}{t\,v \longrightarrow r\,v} \qquad \frac{t \longrightarrow \vec{r}}{\alpha \cdot t + \vec{s} \longrightarrow \alpha \cdot \vec{r} + \vec{s}}$$

$$\frac{t \longrightarrow \vec{r}}{\mathsf{let}_{(A,B)}\ (x,y) = t\ \mathsf{in}\ \vec{s} \longrightarrow \mathsf{let}_{(A,B)}\ (x,y) = \vec{r}\ \mathsf{in}\ \vec{s}}$$

$$\frac{t \longrightarrow \vec{r}}{\begin{array}{l}\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v} \mapsto \vec{s_1} \mid \cdots \mid \vec{w} \mapsto \vec{s_2}\} \longrightarrow \\ \qquad \mathsf{case}\ \vec{r}\ \mathsf{of}\{\vec{v} \mapsto \vec{s_1} \mid \cdots \mid \vec{w} \mapsto \vec{s_2}\}\end{array}}$$

Table 1.4: *Reduction system*

1.3 Reduction system

The reduction system implements a mechanism where every vector in the space is read in the corresponding basis attached to the abstraction. It does this by allowing an evaluation step only when the argument can be decomposed onto that basis. The system works modulo the congruence defined in Table 1.3. We describe it in detail in Table 1.4.

The three main rules are the β-reduction, let-destructor and case pattern matching. The λ abstraction and let construct both attach an orthonormal basis to the variables they are binding. These bases keep track of which vectors it considers as classical data. Any $\mathbb{C}$-combination of them will be treated as quantum data, meaning, linearly.

The only exception is in the case of higher order reductions. Since we do not have defined orthogonal bases for programs, we introduce a special basis $\mathcal{P}$ which acts as the traditional computational basis. We can think of it as being composed of every single pure value. For example:

$$\sum_{i=1}^{n} \alpha_i (\lambda x_{\mathcal{P}} \,.\, \vec{t_i}) \sum_{j=1}^{m} \beta_j (\lambda y_{B_X} \,.\, \vec{s_j}) \to$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\beta_j\vec{t_i}[(\lambda y_{B_X}\,.\,\vec{s_j})/x]$$

The case pattern matching controls the flow of programs. It generalizes the if − then − else branching. However, we do not consider fixed true or false values. Each operator will keep track of a set of orthogonal values. Then it will test the argument for equality against each vector and choose the matching branch. If the argument is a linear combination of several vectors, the result will be the corresponding linear combination of branches. For example:

$$\mathsf{case}\ |-\rangle\ \ \mathsf{of}\{|0\rangle \mapsto \vec{t_1} \mid |1\rangle \mapsto \vec{t_2}\} \to \frac{1}{\sqrt{2}}\cdot\vec{t_1} - \frac{1}{\sqrt{2}}\cdot\vec{t_2}$$

The advantage of this general approach over a binary conditional is the possibility to match against several vectors simultaneously. For boolean tuples, it makes no difference since we can treat each component independently. However, there are orthogonal bases which cannot be written as the product of two smaller bases themselves. In this case, the general case allows us match against these vectors. For example:

$$\begin{aligned}\mathsf{case}\ \vec{v}\ \mathsf{of}\ \{&\frac{|00\rangle + |11\rangle}{2} \mapsto \vec{t_1}\ \mid\\ &\frac{|00\rangle - |11\rangle}{2} \mapsto \vec{t_2}\ \mid\\ &\frac{|01\rangle + |10\rangle}{2} \mapsto \vec{t_3}\ \mid\\ &\frac{|01\rangle - |10\rangle}{2} \mapsto \vec{t_4}\}\end{aligned}$$

This particular set of four vectors is called the *Bell basis*. It is useful in the field of quantum communication. In a later section, we will explore the quantum teleportation algorithm which heavily relies on these states.

Defining the system in this way determines a strategy in the *call-by-value* family, which we dub *call-by-arbitrary-basis*. Note that evaluation is weak, meaning that no reduction occurs under lambda, pairs, let or conditional constructors.

The congruence relation on terms gives rise to different redexes. However, we can show that the relation $\equiv$ commutes with the reflexive-transitive closure of the reduction $\to$ (We shall note $\twoheadrightarrow$ as this reflexive-transitive closure). In other words, equivalence is preserved by the reduction $\twoheadrightarrow$.

Theorem 1 (Commutation of $\equiv$ and $\twoheadrightarrow$?)**.** *Let $\vec{t}$ and $\vec{s}$ be closed term distributions such that $\vec{t} \equiv \vec{s}$. If $\vec{t} \to \vec{t'}$, and $\vec{s} \to \vec{s'}$. Then there exists term distributions $\vec{r_1}, \vec{r_2}$ such that $\vec{t'} \twoheadrightarrow \vec{w}$ and $\vec{v} \equiv \vec{w}$.*

Proof. Induction over the relation $\equiv$.

$\vec{t}(\alpha\vec{s}) \equiv \alpha(\vec{t}\vec{s})$**:** If either $\vec{t}$ or $\vec{s}$ reduce, then merely reduce on both sides of the congruence. If $\vec{t} = \sum_{i=1}^{n}\beta_i(\lambda x_A\,.\,\vec{t_i})$ and $\vec{s} = \vec{v} = \sum_{j=1}^{m}\delta_j\vec{a_j}$ for $\vec{a_j} \in A$, then:

$$(\sum_{i=1}^{n}\beta_i(\lambda x_A\,.\,\vec{t_i}))(\alpha\vec{v}) \to \sum_{i=1}^{n}\beta_i\vec{t_i}\langle\alpha\vec{v}/x\rangle_A$$

$$= \sum_{i=1}^{n} \beta_i \sum_{j=1}^{m} \alpha\delta_j \vec{t_i}[\vec{a_j}/x]$$

On the other side:

$$\alpha((\sum_{i=1}^{n} \beta_i(\lambda x_A \,.\, \vec{t_i}))\vec{v}) \to \alpha(\sum_{i=1}^{n} \beta_i \vec{t_i}\langle\vec{v}/x\rangle_A)$$

$$= \alpha(\sum_{i=1}^{n} \beta_i \sum_{j=1}^{m} \delta_j \vec{t_i}[\vec{a_j}/x])$$

And we have that:

$$\sum_{i=1}^{n} \beta_i \sum_{j=1}^{m} \alpha\delta_j \vec{t_i}[\vec{a_j}/x] \equiv \alpha(\sum_{i=1}^{n} \beta_i \sum_{j=1}^{m} \delta_j \vec{t_i}[\vec{a_i}/x])$$

$(\alpha\vec{t})\vec{s} \equiv \alpha(\vec{t}\vec{s})$**:** If either $\vec{t}$ or $\vec{s}$ reduce, then merely reduce them on both sides of the congruence. If $\vec{t} = \sum_{i=1}^{n} \beta_i(\lambda x_A \,.\, \vec{t_i})$ and $\vec{s} = \vec{v}$ with $\vec{t_i}\langle\vec{v}/x\rangle_A$ defined, then:

$$(\alpha \sum_{i=1}^{n} \beta_i(\lambda x_A \,.\, \vec{t_i}))\vec{v} \to \alpha \sum_{i=1}^{n} \beta_i \vec{t_i}\langle\vec{v}/x\rangle_A$$

On the other side:

$$\alpha(\sum_{i=1}^{n} \beta_i(\lambda x_A \,.\, \vec{t_i})\vec{v}) \to \alpha \sum_{i=1}^{n} \beta_i \vec{t_i}\langle\vec{v}/x\rangle_A$$

$(\vec{t} + \vec{s})\vec{r} \equiv \vec{t}\vec{s} + \vec{t}\vec{r}$**:** There are two cases to consider. If there is a reduction on $\vec{t}, \vec{s}$ or $\vec{r}$, then we have to merely apply the same reduction on both sides of the congruence. If $\vec{t} = \sum_{i=1}^{n} \alpha_i(\lambda x_A \,.\, \vec{t_i})$, $\vec{s} = \sum_{j=1}^{m} \beta_j$ and $\vec{r} = \vec{v}$ then:

$$(\sum_{i=1}^{n} \alpha_i(\lambda x_A \,.\, \vec{t}) + \sum_{j=1}^{m} \beta_j(\lambda x_A \,.\, \vec{s_j}))\vec{v} \to$$

$$\sum_{i=1}^{n} \alpha_i \vec{t_i}\langle\vec{v}/x\rangle_A + \sum_{j=1}^{m} \beta_j \vec{s_j}\langle\vec{v}/x\rangle_A$$

On the other side:

$$(\sum_{i=1}^{n} \alpha_i(\lambda x_A \,.\, \vec{t}))\vec{v} + (\sum_{j=1}^{m} \beta_j(\lambda x_A \,.\, \vec{s_j}))\vec{v} \twoheadrightarrow$$

$$\sum_{i=1}^{n} \alpha_i \vec{t_i}\langle\vec{v}/x\rangle_A + \sum_{j=1}^{m} \beta_j \vec{s_j}\langle\vec{v}/x\rangle_A$$

$\vec{t}(\vec{s} + \vec{r}) \equiv \vec{t}\vec{s} + \vec{t}\vec{r}$**:** There are two cases to consider. If there is a reduction on $\vec{t}, \vec{s}$ or $\vec{r}$, then we have to merely apply the same reduction on both sides of the congruence. If $\vec{t} = \sum_{i=1}^{n} \alpha_i(\lambda x_A \,.\, \vec{t_i})$,

$\vec{s} = \vec{v} \equiv \sum_{j=1}^{m} \beta_j \vec{a_j}$ and $\vec{r} = \vec{w} \equiv \sum_{j=1}^{m} \delta_j \vec{a_j}$ with $\vec{a_j} \in A$. Then:

$$(\sum_{i=1}^{n} \alpha_i (\lambda x_A \,.\, \vec{t_i}))(\vec{v} + \vec{w}) \twoheadrightarrow \sum_{i=1}^{n} \alpha_i \vec{t_i} \langle \vec{v} + \vec{w} \rangle_A$$
$$= \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} (\beta_j + \delta_j) \vec{t_i} [\vec{a_j}/x]$$

On the other side:

$$(\sum_{i=1}^{n} \alpha_i (\lambda x_A \,.\, \vec{t_i}))\vec{v} + (\sum_{i=1}^{n} \alpha_i (\lambda x_A \,.\, \vec{t_i}))\vec{w} \twoheadrightarrow$$
$$\sum_{i=1}^{n} \alpha_i \vec{t_i} \langle \vec{v}/x \rangle_A + \sum_{i=1}^{n} \alpha_i \vec{t_i} \langle \vec{w}/x \rangle_A$$
$$= \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} \beta_j \vec{t_i} [\vec{a_j}/x] + \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} \delta_j \vec{t_i} [\vec{a_j}/x]$$

And we have that:

$$\sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} (\beta_j + \delta_j) \vec{t_i} [\vec{a_j}/x] \equiv$$
$$\sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} \beta_j \vec{t_i} [\vec{a_j}/x] + \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} \delta_j \vec{t_i} [\vec{a_j}/x]$$

$\mathsf{let}_{(A,B)}\ (x_1, x_2) = (\alpha \vec{t})\ \mathsf{in}\ \vec{s} \equiv \alpha(\mathsf{let}_{(A,B)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{s})$**:**

There are two cases to consider. If there is a reduction on $\vec{t}$, then we have to merely apply the same reduction on both sides of the congruence. If $\vec{t} = \vec{v} = (\vec{w}, \vec{u})$, then:

$$\mathsf{let}_{(A,B)}\ (x_1, x_2) = (\alpha \vec{v})\ \mathsf{in}\ \vec{s} \to \vec{s} \langle \alpha \vec{v}/x_1 \otimes x_2 \rangle_{A \otimes B}$$
$$= \alpha \vec{s} \langle \vec{w}/x_1 \rangle_A \langle \vec{u}/x_2 \rangle_B$$

On the other side:

$$\alpha(\mathsf{let}_{(A,B)}\ (x_1, x_2) = \vec{v}\ \mathsf{in}\ \vec{s}) \to \alpha(\vec{s} \langle \vec{v}/x_1 \otimes x_2 \rangle_{A \otimes B})$$
$$= \alpha(\vec{s} \langle \vec{w}/x_1 \rangle_A \langle \vec{u}/x_2 \rangle_B)$$

And we have that:

$$\alpha \vec{s} \langle \vec{w}/x_1 \rangle_A \langle \vec{u}/x_2 \rangle_B \equiv \alpha(\vec{s} \langle \vec{w}/x_1 \rangle_A \langle \vec{u}/x_2 \rangle_B)$$

$\mathsf{let}_{(A,B)}\ (x_1, x_2) = \vec{t} + \vec{s}\ \mathsf{in}\ \vec{r} \equiv$
$(\mathsf{let}_{(A,B)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{r}) + (\mathsf{let}_{(A,B)}\ (x_1, x_2) = \vec{s}\ \mathsf{in}\ \vec{r})$ **:**

There are two cases to consider. If there is a reduction on either $\vec{t}$ or $\vec{s}$, then we have to merely apply the same reduction on both

sides of the congruence. When $\vec{t} = \vec{u_1} \equiv \sum_{i=1}^{n} \alpha_i(\vec{v_i}, \vec{w_i})$, and $\vec{s} = \vec{u_2} \equiv \sum_{i=1}^{n} \beta_i(\vec{v_i}, \vec{w_i})$ where $\vec{v_i} \in A$ and $\vec{w_i} \in B$. Then:

$$\mathsf{let}_{(A,B)}\ (x_1, x_2) = \vec{u_1} + \vec{u_2}\ \mathsf{in}\ \vec{r} \to \vec{r}\langle \vec{u_1} + \vec{u_2}/x_1 \otimes x_2\rangle_{A\otimes B} \sum_{i=1}^{n}(\alpha_i + \beta_i)\vec{r}[\vec{v_i}/x_1][\vec{w_i}/x_2]$$

On the other side:

$$\begin{aligned}(\mathsf{let}_{(A,B)}\ (x_1, x_2) = \vec{u_1}\ \mathsf{in}\ \vec{r}) + (\mathsf{let}_{(A,B)}\ (x_1, x_2) = \vec{u_2}\ \mathsf{in}\ \vec{r}) &\twoheadrightarrow \vec{r}\langle \vec{u_1}/x_1 \otimes x_2\rangle_{A\otimes B}\langle \vec{u_2}/x_1 \otimes x_2\rangle_{A\otimes B} \\ &= \sum_{i=1}^{n} \alpha_i \vec{r}[\vec{v_i}/x_1][\vec{w_i}/x_2] + \sum_{i=1}^{n} \beta_i \vec{r}[\vec{v_i}/x_1][\vec{w_i}/x_2]\end{aligned}$$

And we have that:

$$\sum_{i=1}^{n}(\alpha_i + \beta_i)\vec{r}[\vec{v_i}/x_1][\vec{w_i}/x_2] \equiv \sum_{i=1}^{n} \alpha_i \vec{r}[\vec{v_i}/x_1][\vec{w_i}/x_2] + \sum_{i=1}^{n} \beta_i \vec{r}[\vec{v_i}/x_1][\vec{w_i}/x_2]$$

$\mathsf{case}\ \alpha\vec{t}\ \mathsf{of}\{\vec{v} \mapsto \vec{s_1} \mid \cdots \mid \vec{w} \mapsto \vec{s_n}\} \equiv \alpha(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v} \mapsto \vec{s_1} \mid \cdots \mid \vec{w} \mapsto \vec{s_n}\})$**:**
There are two cases to consider. If there is a reduction on $\vec{t}$ then we merely apply the same reduction on both sides of the congruence. If $\vec{t} = \vec{u} \equiv \sum_{i=1}^{n} \beta_i \vec{v_i}$, with $\sum_{i=1}^{n} |\beta_i|^2 = 1$, then:

$$\mathsf{case}\ \alpha\vec{u}\ \mathsf{of}\{\vec{v} \mapsto \vec{s_1} \mid \cdots \mid \vec{w} \mapsto \vec{s_2}\} \to \sum_{i=1}^{n} \alpha\beta_i \vec{s_i}$$

On the other side:

$$\alpha(\mathsf{case}\ \vec{u}\ \mathsf{of}\{\vec{v} \mapsto \vec{s_1} \mid \cdots \mid \vec{w} \mapsto \vec{s_2}\}) \to \alpha(\sum_{i=1}^{n} \beta_i \vec{s_i})$$

And we have that $\sum_{i=1}^{n} \alpha\beta_i \vec{s_i} \equiv \alpha(\sum_{i=1}^{n} \beta_i \vec{s_i})$.

$\mathsf{case}\ (\vec{t} + \vec{s})\ \mathsf{of}\{\vec{v} \mapsto \vec{r_1} \mid \cdots \mid \vec{w} \mapsto \vec{r_n}\} \equiv$
$\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{r_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{r_n}\} +$
$\mathsf{case}\ \vec{s}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{r_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{r_n}\}$**:**

There are two cases to consider. If there is a reduction on either $\vec{t}$ or $\vec{s}$ then we merely apply the same reduction on both sides of the congruence. If $\vec{t} = \vec{u_1} \equiv \sum_{i=1}^{n} \alpha_i \vec{v_1}$, $\vec{s} = \vec{u_2} \equiv \sum_{i=1}^{n} \beta_i \vec{v_i}$, then:

$$\mathsf{case}\ (\vec{u_1} + \vec{u_2})\ \mathsf{of}\{\vec{v_1} \mapsto \vec{r_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{r_n}\} \twoheadrightarrow \sum_{i=1}^{n} \alpha_i\beta_i\vec{r_i}$$

On the other side:

$$\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{r_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{r_n}\} +$$

$$\mathsf{case}\ \vec{s}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{r_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{r_n}\} \twoheadrightarrow \sum_{i=1}^{n} \alpha_i \vec{r_i} + \sum_{i=1}^{n} \beta_i \vec{r_i}$$

And we have that $\sum_{i=1}^{n} \alpha_i \beta_i \vec{r_i} \equiv \sum_{i=1}^{n} \alpha_i \vec{r_i} + \sum_{i=1}^{n} \beta_i \vec{r_i}$.

□

Convention 1. *With the previous result in mind, we will consider term distributions modulo the $\equiv$ congruence. This will not affect distributions under λ-abstractions or case conditionals which we only consider up to α-conversion.*

1.4 Realizability model

In this section, we present the type system corresponding to the untyped language introduced in the previous section, along with its realizability semantics.

1.4.1 Unitary Type Semantics

Given a deterministic machine, the next step to extract a language is to define the sets of values which will characterize its types. In order to achieve this we first need to identify the notion of what exactly constitutes a type.

Definition 4. *Unitary type We define a unitary type (or just type) as a notation A together with a set of unitary value distributions noted $[\![A]\!]$ called the unitary semantics of A.*

Our aim is to define types that are exclusively inhabited by values of norm equal to 1. The vectors that we wish to study all fall in the *unit sphere*. We will write $\mathcal{S}_1$ for the set $\mathcal{S}_1 := \{\vec{v} \in \vec{\mathbb{V}} \mid \|\vec{v}\| = 1\}$. This corresponds with the mathematical notion of representing quantum data as unit vectors in a Hilbert space.

We next move onto the type realizers. Since our aim is to extract a quantum lambda calculus, we wish to filter global phases of qubits at this level. Since the global phase of a quantum state has no physical significance, we wish to assign the same types to a term $\vec{t}$ and $e^{i\theta} \cdot \vec{t}$. This idea will guide the definition of type realizers.

Definition 5 (Type realizer)**.** *Given a type A and a term distribution $\vec{t}$, we say that $\vec{t}$ realizes A (noted $\vec{t} \Vdash A$), when there is a value distribution $\vec{v}$ such that:*

- $\vec{t} \twoheadrightarrow e^{i\theta} \cdot \vec{v}$
- $\vec{v} \in [\![A]\!]$

For each type A, we note the set of its realizers as $\{\Vdash A\}$.

With the notions of unitary types and its realizers we can start defining the specific approach for our previously defined language. We begin with a type grammar defined on Table 1.5 and build a simple algebra from the sets of values we aim to represent. From this point onwards we refer to $\mathbb{T}$ to the set of all types and $\mathbb{T}_B$ to the set of all bases.

$$T := B_X \mid T \to T \mid T \times T \mid \sharp T$$

$$[\![B_X]\!] := X \qquad \text{Where: } X \text{ is an orthonormal basis}$$

$$[\![A \times B]\!] := \Big\{(\vec{v}, \vec{w}) : \vec{v} \in [\![A]\!],\ \vec{w} \in [\![B]\!]\Big\}$$

$$[\![A \Rightarrow B]\!] := \Big\{\sum_{i=1}^{n} \alpha_i(\lambda x_B\,.\,\vec{t_i}) \in \mathcal{S}_1 : \forall \vec{w} \in [\![A]\!], (\sum_{i=1}^{n} \alpha_i \vec{t_i})\langle \vec{w}/x\rangle_A \Vdash B\Big\}$$

$$[\![\sharp A]\!] := ([\![A]\!]^{\perp})^{\perp}$$

$$\text{Where: } A^{\perp} = \{\vec{v} \in \mathcal{S}_1 \mid \langle \vec{v} \mid a\rangle = 0,\ \forall a \in A\}$$

Table 1.5: *Type notations and semantics*

The types B_X act as atomic types. They represent a finite set X of orthogonal vectors conforming an orthonormal basis. We can represent boolean values with a basis of size 2, but we are not limited to only one kind since there are infinite bases to choose from.

The type $A \times B$ represents the cartesian product of A and B. However, the syntax grammar only allows for pairs of pure values. So there is a small subtlety on the type depicted in the table. For every $\vec{v} = \sum_{i=1}^{n} \alpha_i v_i \in [\![A]\!]$ and $\vec{w} = \sum_{j=1}^{m} \beta_j w_j \in [\![B]\!]$ (With v_i and w_j pure values) when we filter out the notation for pairs, we get:

$$[\![A \times B]\!] := \Big\{\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j(v_i, w_j) : \vec{v} \in [\![A]\!],\ \vec{w} \in [\![B]\!]\Big\}$$

We stress this fact for rigorousness, but for ease of reading from this point onwards we will make use of the previously defined notation.

The arrow type $A \Rightarrow B$ is conformed by the distributions of lambda abstractions that take values from the interpretation of A to realizers of B. The last type $\sharp A$ takes the double orthogonal complement and intersects it with the unit sphere. In doing so, we are left with the $\mathbb{C}$-linear combinations of value distributions of type A, this will represent a superposition of values of type A.

The type grammar is standard except for type $\sharp A$. We use it to represent quantum data, i.e. linear resources, so terms of this type will not be able to be erased or duplicated. This can be thought as the opposite of the *bang* (!) modality in linear logic.

Intuitively, applying the sharp ($\sharp$) operator to a type A yields the span of the original type (Intersected with the unitary sphere). This describes the possible linear combinations of values of type A. The following proposition proves that characterization:

Proposition 1. *The type interpretation* $[\![\sharp A]\!]$ *contains the norm-*1 *linear combination of values in* $[\![A]\!]$.

$$[\![\sharp A]\!] = ([\![A]\!]^{\perp})^{\perp} = \mathsf{Span}([\![A]\!]) \cap \mathcal{S}_1$$

Proof. Proof by double inclusion.

$\mathsf{Span}(\llbracket A \rrbracket) \cap \mathcal{S}_1 \subseteq (\llbracket A \rrbracket^\perp)^\perp$**:** Let $\vec{v} \in \mathsf{Span}(\llbracket A \rrbracket) \cap \mathcal{S}_1$. Then $\vec{v}$ is of the form $\sum_{i=1}^{n} \alpha_i \vec{v_i}$ with $\vec{v_i} \in \llbracket A \rrbracket$. Taking $\vec{w} \in \llbracket A \rrbracket^\perp$, we examine the inner product:

$$\begin{aligned} \langle \vec{v} \mid \vec{w} \rangle &= \langle \sum_{i=1}^{n} \alpha_i \vec{v_i} \mid \vec{w} \rangle \\ &= \sum_{i=1}^{n} \overline{\alpha_i} \langle \vec{v_i} \mid \vec{w} \rangle = 0 \end{aligned}$$

Then $\vec{v} \in (\llbracket A \rrbracket^\perp)^\perp$.

$(\llbracket A \rrbracket^\perp)^\perp \subseteq \mathsf{Span}(\llbracket A \rrbracket) \cap \mathcal{S}_1$**:** Reasoning by contradiction, we assume that there is a $\vec{v} \in (\llbracket A \rrbracket^\perp)^\perp$ such that $v \notin \mathsf{Span}(\llbracket A \rrbracket) \cap \mathcal{S}_1$. Since $\vec{v} \notin \mathsf{Span}(\llbracket A \rrbracket)$, $\vec{v} = \vec{w_1} + \vec{w_2}$ such that $\vec{w_1} \in \mathsf{Span}\llbracket A \rrbracket$ and $\vec{w_2}$ is a non-null vector which cannot be written as a linear combination of elements of $\llbracket A \rrbracket$. In other words, $\vec{w_2} \in \llbracket A \rrbracket^\perp$. Taking the inner product:

$$\langle \vec{v} \mid \vec{w_2} \rangle = \langle \vec{w_1} + \vec{w_2} \mid \vec{w_2} \rangle = \|\vec{w_2}\| \neq 0$$

Then $\vec{v} \notin (\llbracket A \rrbracket^\perp)^\perp$. The contradiction stems from assuming $\vec{v} \notin \mathsf{Span}\llbracket A \rrbracket \cap \mathcal{S}_1$.

□

The following proposition shows that, as one would expect from the span, multiple applications of the sharp operator does not produce a different result beyond the first one.

Proposition 2. *The ♯ operator is idempotent, that is* $\llbracket \sharp A \rrbracket = \llbracket \sharp(\sharp A) \rrbracket$

Proof. We want to prove that $(((A^\perp)^\perp)^\perp)^\perp = (A^\perp)^\perp$. For ease of reading, we will write $A^{\perp^n}$ for n successive applications of the operation $\perp$.

$A \subseteq A^{\perp^2}$**:** Let $\vec{v} \in A$. Then, for all $\vec{w} \in A^\perp$, $\langle \vec{v} \mid \vec{w} \rangle = 0$. Then $\vec{v} \in A^{\perp^2}$. With this we have $A \subseteq A^{\perp^2}$.

$A^{\perp^3} \subseteq A^\perp$**:** Let $\vec{u} \in A^{\perp^3}$. Then, for all $\vec{v} \in A^{\perp^2}$, $\langle \vec{u} \mid \vec{v} \rangle = 0$. Since we have shown that $A \subseteq A^{\perp^2}$, we have that for all $\vec{w} \in A$, $\langle \vec{u} \mid \vec{w} \rangle = 0$. Then $\vec{u} \in A^\perp$. With this we have $A^{\perp^3} \subseteq A^\perp$.

With these two inclusions we have that $A^\perp = A^{\perp^3}$. So we conclude that: $\llbracket \sharp(\sharp A) \rrbracket = A^{\perp^4} = A^{\perp^2} = \llbracket \sharp A \rrbracket$ □

Remark 3. *A basis type B_X may be formed by value distributions of pairs and so might be written as the product type of smaller bases. For example, let $X = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, then $B_X = \mathbb{B} \times \mathbb{B}$. However, for the case of entangled bases this cannot be done. A clear example is the Bell basis:* $\mathsf{Bell} = \{\frac{|00\rangle + |11\rangle}{\sqrt{2}}, \frac{|00\rangle - |11\rangle}{\sqrt{2}}, \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \frac{|01\rangle - |10\rangle}{\sqrt{2}}\}$.

The only thing left would be to check that our type algebra captures sets of value distributions we wish to study. Proposition 3 states that every member of a type interpretation has norm 1.

Proposition 3. *For every type A,* $\llbracket A \rrbracket \subseteq \mathcal{S}_1$.

Proof. Proof by induction on the shape of A. Since by definition, $[\![B_X]\!]$, $[\![A \Rightarrow B]\!]$ and $[\![\sharp A]\!]$ are built from values in $\mathcal{S}_1$ the only case we need to examine is $A \times B$.

Let $\vec{v} = \sum_{i=0}^{n} \alpha_i v_i \in [\![A]\!]$ and $\vec{w} = \sum_{j=0}^{m} \beta_j w_j$ where every v_i are pairwise orthogonal, same for w_j. Then:

$$(\vec{v}, \vec{w}) = \sum_{i=0}^{n} \sum_{j=0}^{m} \alpha_i \beta_j (v_i, w_j)$$

So we have:

$$\|(\vec{v}, \vec{w})\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |\alpha_i \beta_j|^2} = \sqrt{\sum_{i=1}^{n} |\alpha_i|^2 \sum_{j=1}^{m} |\beta_j|^2}$$

Since both $\vec{v} \in [\![A]\!]$ and $\vec{w} \in [\![B]\!]$, by inductive hypothesis, we have that $\|\vec{v}\| = \|\vec{w}\| = 1$. Which is to say $\sum_{i=1}^{n} |\alpha_i|^2 = \sum_{j=1}^{m} |\beta_j| = 1$. So we conclude $\|(\vec{v}, \vec{w})\| = 1$.

□

Defining types as sets of values induces, at the same time, an intuitive way to define a subtyping relationship. We say a type A is subtype of a type B (Noted $A \leq B$) if the set of realizers of A is included in the set of realizers of B ($\{\Vdash A\} \subseteq \{\Vdash B\}$). If the sets coincide, we say that A is isomorphic to B (Noted $A \cong B$).

Example 1.1. *For example, for every type A, $A \leq \sharp A$. For bases, $B_{\mathbb{B}}$ and $B_{\mathbb{X}}$ we have that: neither $B_{\mathbb{B}} \not\leq B_{\mathbb{X}}$, nor $B_{\mathbb{B}} \not\leq B_{\mathbb{X}}$. However, $\sharp B_{\mathbb{B}} \cong \sharp B_{\mathbb{X}}$.*

Although, every type is defined by norm 1 value distributions, not every norm 1 distribution belongs to a type. Take for example the distribution $\frac{1}{\sqrt{2}}(|0\rangle + (|0\rangle, |0\rangle))$. Another case is a linear combination of abstractions with different bases. For example, the term:

$$\frac{1}{\sqrt{2}}(\lambda x_{\mathbb{B}} \,.\, \mathsf{NOT}\ x) + \frac{1}{\sqrt{2}}(\lambda x_{\mathbb{X}} \,.\, x)$$

Is not a member of an arrow type, since the bases decorating each abstraction do not match. However, it is computationally equivalent to the abstraction $(\lambda x_{\mathbb{B}} \,.\, |+\rangle)$ which belongs to the set $[\![B_{\mathbb{B}} \Rightarrow B_{\mathbb{X}}]\!]$.

1.4.2 Characterization of unitary operators

One of the main results of [1], is the characterization of $\mathbb{C}^2 \to \mathbb{C}^2$ unitary operators using values of type $\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B}$ [1, Theorem IV.12]. In this subsection we expand on this result. Our goal is to prove that abstractions of type $\sharp B_X \Rightarrow \sharp B_Y$ (both bases of size n) represent $\mathbb{C}^n \to \mathbb{C}^n$ unitary operators.

Unitary operators are the isomorphisms of Hilbert spaces since they preserve the basic structure of the space. With this in mind, the first step is to show that the members in $\sharp B_X \Rightarrow \sharp B_Y$ send basis vectors from B_X onto orthogonal vectors in $[\![\sharp B_Y]\!]$. In other words, these abstractions preserve both norm and orthogonality.

Lemma 3. *Given types B_X, B_Y of size n and a closed λ-abstraction $\lambda x_X \,.\, \vec{t}$ we have that $\lambda x_A \,.\, \vec{t} \in [\![\sharp B_X \Rightarrow \sharp B_Y]\!]$ if and only if there are value distributions $\vec{w_i} \in [\![\sharp B_Y]\!]$ such that $\forall \vec{v_i} \in [\![B_X]\!]$:*

$$\vec{t}[\vec{v_i}/x] \twoheadrightarrow \vec{w_i} \perp \vec{w_j} \twoheadleftarrow \vec{t}[\vec{v_j}/x] \qquad \text{if } i \neq j$$

Proof. The condition is necessary: Suppose that $\lambda x_X \,.\, \vec{t_k} \in [\![\sharp B_X \Rightarrow \sharp B_Y]\!]$, thus $\forall \vec{v_i} \in [\![\sharp B_X]\!]$, $\vec{t}\langle \vec{v_i}/x \rangle_X \twoheadrightarrow \vec{w_i} \in [\![\sharp B_Y]\!]$. It remains to be seen that $\vec{w_i} \perp \vec{w_j}$ if $i \neq j$. For that, we consider $\alpha_i \in \mathbb{C}$ such that $\sum_{i=1}^{n} |\alpha_i|^2 = 1$. By linear application on the basis X we observe that:

$$\begin{aligned}(\lambda x_X \,.\, \vec{t})(\sum_{i=1}^{n} \alpha_i \vec{v_i}) &\to \vec{t}\langle \sum_{i=1}^{n} \alpha_i \vec{v_i}/x \rangle_X \\ &= \sum_{i=1}^{n} \alpha_i \vec{t}[\vec{v_i}/x] \\ &\twoheadrightarrow \sum_{i=1}^{n} \alpha_i \vec{w_i}\end{aligned}$$

But since $\sum_{i=1}^{n} \alpha_i \vec{v_i} \in [\![\sharp A]\!]$, then $\sum_{i=1}^{n} \alpha_i \vec{w_i} \in [\![\sharp B]\!]$ too. Which implies $\| \sum_{i=1}^{n} \alpha_i \vec{w_i} \| = 1$. Therefore:

$$\begin{aligned}1 = \| \sum_{i=1}^{n} \alpha_i \vec{w_i} \| &= \langle \sum_{i=1}^{n} \alpha_i \vec{w_i} \mid \sum_{j=1}^{n} \alpha_j \vec{w_j} \rangle \\ &= \sum_{i=1}^{n} |\alpha_i|^2 \langle \vec{w_i} \mid \vec{w_i} \rangle + \sum_{i,j=1; i \neq j}^{n} \bar{\alpha_i} \alpha_j \langle \vec{w_i} \mid \vec{w_j} \rangle \\ &= \sum_{i=1}^{n} |\alpha_i|^2 \langle \vec{w_i} \mid \vec{w_i} \rangle + \sum_{i,j=1; i<j}^{n} 2 \ \mathsf{Re}(\bar{\alpha_i} \alpha_j \langle \tilde{w_i} \mid \tilde{w_j} \rangle) \\ &= \sum_{i=1}^{n} |\alpha_i|^2 \|\vec{w_i}\|^2 + 2 \sum_{i,j=1; i<j}^{n} \mathsf{Re}(\bar{\alpha_i} \alpha_j \langle \tilde{w_i} \mid \tilde{w_j} \rangle) \\ &= \sum_{i=1}^{n} |\alpha_i|^2 + 2 \sum_{i,j=1; i<j}^{n} \mathsf{Re}(\bar{\alpha_i} \alpha_j \langle \tilde{w_i} \mid \tilde{w_j} \rangle) \\ &= 1 + 2 \sum_{i,j=1; i<j}^{n} \mathsf{Re}(\bar{\alpha_i} \alpha_j \langle \tilde{w_i} \mid \tilde{w_j} \rangle)\end{aligned}$$

And thus we are left with $\sum_{i,j=1; i<j}^{n} \mathsf{Re}(\bar{\alpha_i} \alpha_j \langle \tilde{w_i} \mid \tilde{w_j} \rangle) = 0$. Taking $\alpha_{i'} = \alpha_{j'} = \frac{1}{\sqrt{2}}$ with 0 for the rest of coefficients, we have $\mathsf{Re}(\langle \tilde{w_{i'}} \mid \tilde{w_{j'}} \rangle) = 0$ for any two arbitrary i' and j'. In the same way, taking $\alpha_{i'} = \frac{1}{\sqrt{2}}$ and $\alpha_{j'} = \frac{i}{\sqrt{2}}$ with 0 for the rest of the coefficients, we have $\mathsf{Im}(\langle \tilde{w_{i'}} \mid \tilde{w_{j'}} \rangle) = 0$ for any two arbitrary i' and j'. Finally, we can conclude that $\langle \vec{w_i} \mid \vec{w_j} \rangle = 0$ if $i \neq j$.

The condition is sufficient: Suppose that there are $\vec{w_i} \in [\![\sharp B_Y]\!]$ such that for every $\vec{v_i} \in [\![B_X]\!]$:

$$\vec{t}[\vec{v_i}/x] \twoheadrightarrow \vec{w_i} \perp \vec{w_j} \twoheadleftarrow \vec{t}[\vec{v_j}/x] \qquad \text{If } i \neq j$$

Given any $\vec{u} \in [\![\sharp B_X]\!]$ we have that $\vec{u} = \sum_{i=1}^{n} \alpha_i \vec{v_i}$ with $\sum_{i=1}^{n} |\alpha_i|^2 = 1$ and $\vec{v_i} \in [\![B_X]\!]$. Then

$$(\lambda x_X \,.\, \vec{t})\vec{u} \to \vec{t_k}\langle \vec{u}/x \rangle_X = \sum_{i=1}^{n} \alpha_i \vec{t}[\vec{v_i}/x] \twoheadrightarrow \sum_{i=1}^{n} \alpha_i \vec{w_i}$$

We have that for each i, $\vec{w_i} \in [\![\sharp B_Y]\!]$. In order to show that $(\lambda x_A \,.\, \vec{t})\vec{u} \Vdash \sharp B_Y$ we still have to prove that $\| \sum_{i=1}^{n} \alpha_i \vec{w_i}\| = 1$

$$\begin{aligned}
\| \sum_{i=1}^{n} \alpha_i \vec{w_i}\|^2 &= \langle \sum_{i=1}^{n} \alpha_i \vec{w_i} \mid \sum_{j=1}^{n} \alpha_j \vec{w_j} \rangle \\
&= \sum_{i=1}^{n} |\alpha_i|^2 \langle \vec{w_i} \mid \vec{w_i} \rangle + \sum_{i,j=1; i \neq j}^{n} \bar{\alpha_i} \alpha_j \langle \vec{w_i} \mid \vec{w_j} \rangle \\
&= \sum_{i=1}^{n} |\alpha_i|^2 + 0 \\
&= 1
\end{aligned}$$

Then $\sum_{i=1}^{n} \alpha_i \vec{w_i} \in [\![\sharp(\sharp B_Y)]\!] = [\![\sharp B_Y]\!]$ by lemma 2. Since for every $\vec{u} \in [\![\sharp A]\!]$, $(\lambda x_A \,.\, \vec{t})\vec{u} \Vdash \sharp B$, we can conclude that $\lambda x_A \,.\, \vec{t} \in [\![\sharp A \to \sharp B]\!]$. □

Next, we need to bridge the gap between the values in the calculus with vectors in the space $\mathbb{C}^n$. In order to do this, we introduce a meta-language operation π_n which translates value distributions into vectors in $\mathbb{C}^n$. The operation simply writes the value in the canonical basis and takes the corresponding coefficients.

Definition 6. *Let B_X be an orthonormal basis of size n, then for every $\vec{v} \in [\![B_X]\!]$:*

$$\vec{v} \equiv \sum_{i=1}^{n} \alpha_i \, |i\rangle$$

Where $|i\rangle$ is the n-th dimensional product of $|0\rangle$ and $|1\rangle$ with i written in binary and $\sum_{i=1}^{n} |\alpha_i|^2 = 1$. We define $\pi_n : [\![B_X]\!] \to \mathbb{C}^n$ as (we omit the subscript when it can be deduced from the context):

$$\pi_n(\vec{v}) = (\alpha_1, \cdots, \alpha_n)$$

We say a λ-abstraction $(\lambda x_X \,.\, \vec{t})$ represents an operator $F : \mathbb{C}^n \to \mathbb{C}^n$ when:

$$(\lambda x_X \,.\, \vec{t})\vec{v} \twoheadrightarrow \vec{w} \iff F(\pi_n(\vec{v})) = \pi_n(\vec{w})$$

Basically, a lambda term represents a function $F : \mathbb{C}^n \to \mathbb{C}^n$ if it encodes the action of F on vectors. This definition, in conjunction with the previous lemma, allow us build a characterization of unitary operators as values of type $\sharp B_X \Rightarrow \sharp B_X$.

Theorem 2. *Let B_X, B_Y be orthonormal bases of size n. A closed λ-abstraction $(\lambda x_X \,.\, \vec{t})$ is a value of type $\sharp B_X \Rightarrow \sharp B_Y$ if and only if it represents a unitary operator $F : \mathbb{C}^n \to \mathbb{C}^n$.*

Proof. The condition is necessary: Suppose that $(\lambda x_X \,.\, \vec{t}) \in [\![\sharp B_X \Rightarrow \sharp B_Y]\!]$, then by lemma 3 we have that, for every $\vec{v_i} \in [\![B_X]\!]$ exist $\vec{w_i} \in [\![\sharp B_Y]\!]$ such that $\vec{t}[\vec{v_i}/x] \twoheadrightarrow \vec{w_i}$ and $\vec{w_i} \perp \vec{w_j}$ if $i \neq j$. Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be the operator defined as $F(\pi(\vec{v_i})) = \pi(\vec{w_i})$. From the linear application on X, it is clear that $(\lambda x_X \,.\, \vec{t})$ represents the operator F. Moreover, the operator F is unitary since $\|\pi(\vec{w_i})\|_{\mathbb{C}^n} = \|\pi(\vec{w_j})\|_{\mathbb{C}^n} = 1$ and $\langle \pi(\vec{w_i}) \mid \pi(\vec{w_j})\rangle_{\mathbb{C}^n} = 0$.

The condition is sufficient: Suppose that $(\lambda x_X \,.\, \vec{t})$ represents a unitary operator $F : \mathbb{C}^n \to \mathbb{C}^n$. From this we deduce that:

$$(\lambda x_X \,.\, \vec{t})\vec{v_i} \twoheadrightarrow \vec{w_i}$$

For some $\vec{v_i} \in [\![B_X]\!]$, $\vec{w_i} \in [\![B_Y]\!]$ such that $F(\pi(\vec{v_i})) = \pi(\vec{w_i})$. Then we have:

$$(\lambda x_X \,.\, \vec{t})\vec{v_i} \to \vec{t}\langle \vec{v_i}/x\rangle_X = \vec{t}[\vec{v_i}/x] \twoheadrightarrow \vec{w_i} \in [\![\sharp B_Y]\!],$$

since $\|\vec{w_i}\| = \|F(\pi(\vec{v_i}))\|_{\mathbb{C}^n} = 1$, we can deduce from lemma 3, that $(\lambda x_X \,.\, \vec{t}) \in [\![\sharp B_X \Rightarrow \sharp B_Y]\!]$. Then:

$$\langle \vec{w_i} \mid \vec{w_j}\rangle = \langle F(\pi(\vec{v_i})) \mid F(\pi(\vec{v_j}))\rangle_{\mathbb{C}^n} = 0$$

□

These results can be extended to unitary distributions of lambda abstractions, since $(\lambda x_X \,.\, \sum_{i=1}^n \alpha_i \vec{t_i})$ is syntactically different but computationally equivalent to $\sum_{i=1}^n \alpha_i (\lambda x_X \,.\, \vec{t_i})$. Ultimately, we generalized one of the main theorems in [1]. The inclusion of the basis type in our system allow us to reason more easily about the action of the operators and translate the proof onto a more general case.

1.4.3 Typing rules

Our focus in this section is to enumerate and prove the validity of various typing rules. The objective being to extract a reasonable set of rules to constitute a type system. We first need to lay the groundwork to properly define what does it mean for a typing rule to be valid.

Definition 7. *A context (Denoted by capital Greek letters Γ, Δ) is a mapping $\Gamma : \mathsf{Var} \to \mathbb{T} \times \mathbb{T}_B$ assigning a type and basis to each variable in its domain. We note the mapping $\Gamma(x_i) \mapsto (A_i, B_{X_i})$ as:*

$$\Gamma = x_{1\,B_{X_1}} : A_1, \cdots, x_{n\,B_{X_n}} : A_n$$

As usual with typing judgements, the context will keep track of the type of free variables of a term. However, since the substitution operation depends on a basis we also wish to include that information. This is not strictly necessary, since the basis a variable is interpreted should not impact on the type. For example the result of the substitution:

$$(\lambda x_{\mathbb{B}} \,.\, (x, y))\langle |0\rangle \,/y\rangle_{B_{\mathbb{B}}} = (\lambda x_{\mathbb{B}} \,.\, (x, |0\rangle))$$

And the substitution:

$$(\lambda x_{\mathbb{B}} \,.\, (x, y))\langle |0\rangle \,/y\rangle_{B_{\mathbb{X}}} = \frac{1}{\sqrt{2}}((\lambda x_{\mathbb{B}} \,.\, (x, |+\rangle)) + (\lambda x_{\mathbb{B}} \,.\, (x, |-\rangle)))$$

Are not syntactically equivalent, but at the same time, they are equivalent under elimination contexts. Therefore, since typing via realizability captures computational behaviour, the types will match. We will however keep basis information on the contexts to later simplify our proofs. With this, we can define which substitutions validate a context.

Definition 8. *Given a context* Γ *we call the unitary semantics of* Γ*, noted* $[\![\Gamma]\!]$*, to the set of substitutions such that:*

$$[\![\Gamma]\!] := \{\sigma \textit{ substitution} \quad | \quad \mathrm{dom}(\sigma) = \mathrm{dom}(\Gamma) \textit{ and } \forall x_i \in \mathrm{dom}(\Gamma),$$
$$\Gamma(x_i) = (A_i, B_{X_i}) \Rightarrow \sigma(x_i) = \langle \vec{v_i}/x_i\rangle_{B_{X_i}} \wedge \vec{v_i} \in [\![A_i]\!]\}$$

In order for the calculus to be correct we need to ensure that qubits are treated linearly. The first step is to identify which variables in the context represent quantum data, those will be the ones associated with a type of the form $\sharp A$. We call the subset of Γ composed by these variables, its *strict domain*.

Definition 9. *We define the strict domain of a context* Γ*, noted* $\mathrm{dom}^\sharp(\Gamma)$*, as:*

$$\mathrm{dom}^\sharp(\Gamma) := \{x \in \mathrm{dom}(\Gamma) \mid [\![\Gamma(x)]\!] = [\![\sharp(\Gamma(x))]\!]\}$$

Here we make use of the idempotence of $\sharp$ (Proposition 2) to define the strict domain.

In order for a typing judgement $\Gamma \vdash \vec{t} : A$ to be valid, it needs to comply with two conditions. First, every free variable in the term $\vec{t}$ must be in the domain of the context Γ and every variable in the strict context $\mathrm{dom}^\sharp(\Gamma)$ must appear in the term $\vec{t}$. This ensures there is no erasure of information and every variable is accounted. Linear treatment of quantum data is enforced by the substitution.

Second, every substitution in the unitary semantics of Γ, when applied to the term $\vec{t}$, must yield a term which reduces to a realizer of type A. This condition matches the computational behaviour of the term and context to the type. To put it more precisely:

Definition 10. *We say that a typing judgement* $\Gamma \vdash \vec{t} : A$ *is valid when:*

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathrm{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$
- *For all* $\sigma \in [\![\Gamma]\!]$, $\vec{t}\langle\sigma\rangle \Vdash A$

With this definition in mind, we consider a typing rule to be valid, when starting from judgements we assume to be valid we reach a valid conclusion. In table 1.6 we enumerate several of these rules. One important thing to note is that there are infinite valid rules, we limit ourselves to listing a subset which could constitute a reasonable typing system for a typed calculus.

We are also interested on *orthogonal terms*, that is, terms which reduce to orthogonal values. Naturally, unless these terms are closed, we need to take the context into consideration. We define orthogonality judgements in the following manner:

Definition 11. *We say that an orthogonality judgement* $\Gamma \vdash (\Delta_1 \vdash \vec{t}) \perp (\Delta_2 \vdash \vec{s}) : A$ *is valid when:*

- *The judgement* $\Gamma, \Delta_1 \vdash \vec{t} : A$ *is valid.*
- *The judgement* $\Gamma, \Delta_2 \vdash \vec{s} : A$ *is valid.*
- *For every* $\sigma \in [\![\Gamma, \Delta_1]\!], \tau \in [\![\Gamma, \Delta_2]\!]$ *there are value distributions* $\vec{v}, \vec{w}$ *such that* $\vec{t}\langle\sigma\rangle \twoheadrightarrow \vec{v}, \vec{s}\langle\tau\rangle \twoheadrightarrow \vec{w}$ *and* $\vec{v} \perp \vec{w}$.

If both Δ_1 and Δ_2 are empty, we will note the judgement as $\Gamma \vdash \vec{t} \perp \vec{s} : A$. We will be mostly interested in these cases.

The main result of this section, is the proof of validity of each of the rules presented on table 1.6.

Theorem 3. *The rules in table 1.6 are valid.*

Proof. For each typing rule in Table 1.6 we have to show the typing judgement is valid starting from the premises:

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

Sub Trivial since $\{\Vdash A\} \subseteq \{\Vdash A'\}$.

UnitLam If the hypothesis is valid, $\mathrm{dom}^\sharp(\Gamma, x_A : A) \subseteq \mathrm{FV}(\sum_{i=1}^n \alpha_i \vec{t_i}) \subseteq \mathrm{dom}(\Gamma, x_A : A)$. It follows that $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathrm{FV}(\sum_{i=1}^n \alpha_i \lambda x_A \,.\, \vec{t_i}) \subseteq \mathrm{dom}(\Gamma)$. Given $\sigma \in [\![\Gamma]\!]$, we want to show that $(\sum_{i=1}^n \alpha_i \lambda x_A \,.\, \vec{t_i})\langle\sigma\rangle \Vdash A \Rightarrow B$. Let $\vec{v} \in [\![A]\!]$, then:

$$
\begin{aligned}
(\sum_{i=1}^{n} (\lambda x_A \,.\, \vec{t_i}))\langle\sigma\rangle\vec{v} &= (\sum_{j=1}^{m} \beta_j (\sum_{i=1}^{n} \alpha_i (\lambda x_A \,.\, \vec{t_i})[\sigma_i]))\vec{v} \\
&= (\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j (\lambda x_A \,.\, \vec{t_i}[\sigma_j]))\vec{v} \\
&\to \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j \vec{t_i}[\sigma_j]\langle\vec{v}/x\rangle_A \\
&= \sum_{i=1}^{n} \alpha_i \vec{t_i}\langle\sigma\rangle\langle\vec{v}/x\rangle_A \\
&= (\sum_{i=1}^{n} \alpha_i \vec{t_i})\langle\sigma\rangle\langle\vec{v}/x\rangle_A \qquad \text{By lemma 1}
\end{aligned}
$$

$$\frac{B_X \leq A \vee X = \mathcal{P}}{x_X : A \vdash x : A}\ \text{(Axiom)} \qquad \frac{\Gamma \vdash \vec{t} : A \quad A \leq A'}{\Gamma \vdash \vec{t} : A'}\ \text{(Sub)}$$

$$\frac{\Gamma, x_A : A \vdash \sum_{i=1}^n \alpha_i \vec{t_i} : B}{\Gamma \vdash \sum_{i=1}^n \alpha_i \lambda x_A \,.\, \vec{t_i} : A \Rightarrow B}\ \text{(UnitLam)}$$

$$\frac{\Gamma \vdash \vec{s} : A \Rightarrow B \quad \Delta \vdash \vec{t} : A}{\Gamma, \Delta \vdash \vec{s}\vec{t} : B}\ \text{(App)} \qquad \frac{\Gamma \vdash \vec{t} : A}{\Gamma \vdash e^{i\theta} \cdot \vec{t} : A}\ \text{(GlobalPhase)}$$

$$\frac{\Gamma \vdash \vec{t} : A \quad \Delta \vdash \vec{s} : B}{\Gamma, \Delta \vdash (\vec{t}, \vec{s}) : A \times B}\ \text{(Pair)} \qquad \frac{\Gamma \vdash \vec{t} : B \quad \flat A \quad A \leq B}{\Gamma, x_A : B \vdash \vec{t} : C}\ \text{(Weak)}$$

$$\frac{\Gamma \vdash \vec{t} : A_1 \times A_2 \quad \Delta, x_{B_1} : A_1, y_{B_2} : A_2 \vdash \vec{s} : C}{\Gamma, \Delta \vdash \mathsf{let}_{(B_1,B_2)}\ (x, y) = \vec{t}\ \mathsf{in}\ \vec{s} : C}\ \text{(LetPair)}$$

$$\frac{\Gamma \vdash \vec{t} : \sharp(A_1 \times A_2) \quad \Delta, x_{B_1} : \sharp A_1, y_{B_2} : \sharp A_2 \vdash \vec{s} : C}{\Gamma, \Delta \vdash \mathsf{let}_{(B_1,B_2)}\ (x, y) = \vec{t}\ \mathsf{in}\ \vec{s} : \sharp C}\ \text{(LetTens)}$$

$$\frac{\Gamma \vdash \vec{t} : B_{\{\vec{v_i}\}_{i=1}^n} \quad \forall i,\ \Delta \vdash \vec{s_i} : A}{\Gamma, \Delta \vdash \mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\} : A}\ \text{(Case)}$$

$$\frac{\Gamma \vdash \vec{t} : \sharp B_{\{\vec{v_i}\}_{i=1}^n} \quad \forall i \neq j,\ \Delta \vdash \vec{s_i} \perp \vec{s_j} : A}{\Gamma, \Delta \vdash \mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\} : \sharp A}\ \text{(UnitCase)}$$

$$\frac{\forall i \neq j, \Gamma \vdash \vec{t_i} \perp \vec{t_j} : A \quad \sum_{i=1}^n |\alpha_i|^2 = 1}{\Gamma \vdash \sum_{i=1}^n \vec{t_i} : \sharp A}\ \text{(Sum)}$$

$$\frac{\Gamma, x_A : A, y_A : A \vdash \vec{t} : B \quad \flat A}{\Gamma, x_A : A \vdash \vec{t}[y := x] : B}\ \text{(Contr)} \qquad \frac{\Gamma \vdash \vec{t} : A \quad \vec{t} \equiv \vec{s}}{\Gamma \vdash \vec{s} : A}\ \text{(Equiv)}$$

Where the property $\flat$ is defined as:

$$\flat X \iff \forall \vec{v}, \vec{w} \in [\![X]\!],\ \vec{v} \neq \vec{w} \Rightarrow \langle \vec{v} \mid \vec{w} \rangle = 0$$

Table 1.6: *Some valid typing rules*

Considering that $\langle\sigma\rangle \in [\![\Gamma]\!]$, then we have that $\langle\sigma\rangle\langle\vec{v}/x\rangle_A \in [\![\Gamma, x_A : A]\!]$. Since we assume $\Gamma, x_A : A \vdash \sum_{i=1}^{n} \alpha_i \vec{t_i} : B$, then $\vec{t_i}\langle\sigma\rangle\langle\vec{v}/x\rangle_A \Vdash B$. Finally, we can conclude that the distribution: $\sum_{i=1}^{n} \alpha_i \lambda x_A \,.\, \vec{t_i} \in [\![A \Rightarrow B]\!]$.

App If the hypotheses are valid, then:

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

From this, we can conclude that $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathrm{FV}(\vec{s}\vec{t}) \subseteq \mathrm{dom}(\Gamma, \Delta)$. Given $\sigma \in [\![\Gamma, \Delta]\!]$, we can observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. Then we have:

$$
\begin{aligned}
(\vec{t}\vec{s})\langle\sigma\rangle &= (\vec{t}\vec{s})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \\
&= (\sum_{i=i}^{n} \alpha_i (\vec{t}\vec{s})[\sigma_{\Gamma i}])\langle\sigma_\Delta\rangle \\
&= \sum_{j=1}^{m} \beta_j (\sum_{i=1}^{n} \alpha_i (\vec{t}\vec{s})[\sigma_{\Gamma i}])[\sigma_{\Delta j}] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j \vec{t}[\sigma_{\Gamma i}][\sigma_{\Delta j}]\vec{s}[\sigma_{\Gamma i}][\sigma_{\Delta j}] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j \vec{t}[\sigma_{\Gamma i}]\vec{s}[\sigma_{\Delta j}] \\
&\equiv (\sum_{i=1}^{n} \alpha_i \vec{t}[\sigma_{\Gamma i}])(\sum_{j=1}^{m} \beta_j \vec{s}[\sigma_{\Delta j}]) \\
&= \vec{t}\langle\sigma_\Gamma\rangle\vec{s}\langle\sigma_\Delta\rangle \\
&\twoheadrightarrow (e^{i\theta_1}\vec{v})(e^{i\theta_2}\vec{w}) \qquad \text{Where: } \vec{v} \in [\![A \Rightarrow B]\!], \vec{w} \in [\![A]\!] \\
&\equiv e^{i\theta}(\vec{v}\vec{w}) \qquad \text{With: } \theta = \theta_1 + \theta_2 \\
&\to e^{i\theta}\vec{r} \qquad \text{Where: } \vec{r} \Vdash B
\end{aligned}
$$

Then we can conclude that $(\vec{t}\vec{s})\langle\sigma\rangle \Vdash B$.

Pair If the hypotheses are valid, then:

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

From this, we can conclude that $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathrm{FV}((\vec{s}, \vec{t})) \subseteq \mathrm{dom}(\Gamma, \Delta)$. Given $\sigma \in [\![\Gamma, \Delta]\!]$, we can observe that $\sigma = \sigma_\Gamma, \sigma_\Delta$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. Then we have:

$$
\begin{aligned}
(\vec{t}, \vec{s})\langle\sigma\rangle &= (\vec{t}, \vec{s})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \\
&= \sum_{j=1}^{m} \beta_j (\sum_{i=1}^{n} \alpha_i (\vec{t}, \vec{s})[\sigma_{\Gamma i}])[\sigma_{\Delta j}]
\end{aligned}
$$

$$
\begin{aligned}
&\equiv \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j(\vec{t}[\sigma_{\Gamma i}][\sigma_{\Delta j}], \vec{s}[\sigma_{\Gamma i}][\sigma_{\Delta j}]) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j(\vec{t}[\sigma_{\Gamma i}], \vec{s}[\sigma_{\Delta j}]) \\
&= (\sum_{i=1}^{n} \alpha_i\vec{t}[\sigma_{\Gamma i}], \sum_{j=1}^{m} \beta_j\vec{s}[\sigma_{\Delta j}]) \\
&= (\vec{t}\langle\sigma_\Gamma\rangle, \vec{s}\langle\sigma_\Delta\rangle) \\
&\twoheadrightarrow (e^{i\theta_1}\cdot\vec{v}, e^{i\theta_2}\cdot\vec{w}) \qquad \text{Where: } \vec{v}\in[\![A]\!], \vec{w}\in[\![B]\!] \\
&= e^{i\theta}(\vec{v},\vec{w}) \qquad \text{Where: } \vec{v}\in[\![A]\!], \vec{w}\in[\![B]\!]
\end{aligned}
$$

From this we can conclude that $(\vec{t},\vec{s})\langle\sigma\rangle \Vdash A\times B$. Finally, $\Gamma,\Delta\vdash(\vec{t},\vec{s}) : A\times B$

LetPair If the hypotheses are valid, then:

- $\mathrm{dom}^\sharp(\Gamma)\subseteq \mathrm{FV}(\vec{t})\subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma_\Gamma\rangle \Vdash A\times B\ \forall\sigma_\Gamma\in[\![\Gamma]\!]$
- $\mathrm{dom}^\sharp(\Delta, x_1{}_{B_1} : A_1, x_2{}_{B_2} : A_2)\subseteq \mathrm{FV}(\vec{s})$
- $\mathrm{FV}(\vec{s})\subseteq \mathrm{dom}(\Delta, x_1{}_{B_1} : A_1, x_2{}_{B_2} : A_2)$
- $\vec{s}\langle\sigma_\Delta\rangle \Vdash C\ \forall\sigma_\Delta\in[\![\Delta, x_1{}_{B_1} : A_1, x_2{}_{B_2} : A_2]\!]$

From this, we can conclude that:

- $\mathrm{dom}^\sharp(\Gamma,\Delta)\subseteq \mathrm{FV}(\mathsf{let}_{(B_1,B_2)}\ (x,y)=\vec{s}\ \mathsf{in}\ \vec{t})$
- $\mathrm{FV}(\mathsf{let}_{(B_1,B_2)}\ (x,y)=\vec{s}\ \mathsf{in}\ \vec{t})\subseteq \mathrm{dom}(\Gamma,\Delta)$

Given $\sigma\in[\![\Gamma,\Delta]\!]$, we have that $\langle\sigma\rangle = \langle\sigma_\Gamma\rangle, \langle\sigma_\Delta\rangle$ for some $\sigma_\Gamma\in[\![\Gamma]\!]$ and $\sigma_\Delta\in[\![\Delta]\!]$. Then we have:

$$
\begin{aligned}
&(\mathsf{let}_{(B_1,B_2)}\ (x,y)=\vec{t}\ \mathsf{in}\ \vec{s})\langle\sigma\rangle = \\
&(\mathsf{let}_{(B_1,B_2)}\ (x,y)=\vec{t}\ \mathsf{in}\ \vec{s})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j(\mathsf{let}_{(B_1,B_2)}\ (x,y)=\vec{t}\ \mathsf{in}\ \vec{s})[\sigma_{\Gamma i}][\sigma_{\Delta j}] \\
&\equiv \mathsf{let}_{(B_1,B_2)}\ (x,y)=\sum_{i=1}^{n}\alpha_i[\sigma_{\Gamma i}]\vec{t}\ \mathsf{in}\ \sum_{j=1}^{m}\beta_j\vec{s}[\sigma_{\Delta j}] \\
&= \mathsf{let}_{(B_1,B_2)}\ (x,y)=\vec{t}\langle\sigma_\Gamma\rangle\ \mathsf{in}\ \vec{s}\langle\sigma_\Delta\rangle \\
&\twoheadrightarrow \mathsf{let}_{(B_1,B_2)}\ (x,y)=e^{i\theta}\cdot(\vec{v},\vec{w})\ \mathsf{in}\ \vec{s}\langle\sigma_\Delta\rangle \\
&\qquad\qquad \text{Where: } \vec{v}\in[\![A]\!], \vec{w}\in[\![B]\!] \\
&\to e^{i\theta_1}\cdot(\vec{s}\langle\sigma_\Delta\rangle\langle(\vec{v},\vec{w})/x_1\otimes x_2\rangle_{B_1\otimes B_2}) \\
&= e^{i\theta_1}\cdot(\vec{s}\langle\sigma_\Delta\rangle\langle\vec{v}/x_1\rangle_{B_1}\langle\vec{w}/x_2\rangle_{B_2}) \\
&\twoheadrightarrow e^{i\theta_1}\cdot(e^{i\theta_2}\cdot\vec{u}) \qquad \text{Where: } \vec{u}\in[\![C]\!] \\
&\equiv e^{i\theta}\cdot\vec{u} \qquad \text{Where: } \theta=\theta_1+\theta_2
\end{aligned}
$$

Since $\langle\sigma_\Delta\rangle\langle\vec{v}/x_1\rangle_{B_1}\langle\vec{w}/x_2\rangle_{B_2}\in[\![\Delta, x_1{}_{B_1} : A_1, x_2{}_{B_2} : A_2]\!]$, then we can conclude that $(\mathsf{let}_{(B_1,B_2)}\ (x,y)=\vec{t}\ \mathsf{in}\ \vec{s})\langle\sigma\rangle \Vdash C$.

LetTens If the hypotheses are valid then:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathrm{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$ and $\vec{t}\langle\sigma\rangle \Vdash A \otimes B \ \forall \sigma \in [\![\Gamma]\!]$
- $\mathrm{dom}^\sharp(\Delta, x_{1\,B_1} : \sharp A_1, x_{2\,B_2} : \sharp A_2) \subseteq \mathrm{FV}(\vec{s})$
- $\subseteq \mathrm{dom}(\Delta, x_1 : B_1, x_{2\,B_2} : A_2)$
- $\vec{s}\langle\sigma\rangle \Vdash \sharp C \ \forall \sigma \in [\![\Delta, x_{1\,B_1} : \sharp A_1, x_{2\,B_2} : \sharp A_2]\!]$

From this we can conclude that:

- $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathrm{FV}(\mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{s})$
- $\mathrm{FV}(\mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{s}) \subseteq \mathrm{dom}(\Gamma, \Delta)$

Given $\sigma \in [\![\Gamma, \Delta]\!]$, we have that $\langle\sigma\rangle = \langle\sigma_\Gamma\rangle, \langle\sigma_\Delta\rangle$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. Using the first hypothesis we have that, $\vec{t}\langle\sigma_\Gamma\rangle \Vdash \sharp(A_1 \times A_2)$, from 1 we have that:

$$\vec{t}\langle\sigma_\Gamma\rangle \twoheadrightarrow e^{i\theta_1} \cdot \vec{u} = e^{i\theta_1} \cdot (\sum_{k=1}^{l} \gamma_k(\vec{v}_k, \vec{u}_k))$$

With:

- $\sum_{k=1}^{l} |\gamma_k|^2 = 1$
- $\forall k,\ \vec{v}_k \in [\![A_1]\!],\ \vec{u}_k \in [\![A_2]\!]$
- $\forall k \neq l, \langle(\vec{v_k}, \vec{u_k}) \mid (\vec{v_l}, \vec{u_l})\rangle = 0$

Then:

$$\begin{aligned}
&(\mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{s})\langle\sigma\rangle \\
&= \mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{s}\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} \mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{s}\ [\sigma_{\Gamma i}][\sigma_{\Delta j}] \\
&\equiv \mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = \sum_{i=1}^{n} \alpha_i \vec{t}\ [\sigma_{\Gamma i}]\ \mathsf{in}\ \sum_{j=1}^{m} \beta_j \vec{s}\ [\sigma_{\Delta j}] \\
&= \mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = \vec{t}\langle\sigma_\Gamma\rangle\ \mathsf{in}\ \vec{s}\langle\sigma_\Delta\rangle \\
&\twoheadrightarrow \mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = e^{i\theta_1} \cdot \vec{u}\ \mathsf{in}\ \vec{s}\langle\sigma_\Delta\rangle \\
&\to e^{i\theta_1} \cdot (\vec{s}\langle\sigma_\Delta\rangle\langle\vec{u}/x_1 \otimes x_2\rangle_{B_1\otimes B_2}) \\
&= e^{i\theta_1} \cdot (\sum_{k=1}^{l} \gamma_k \vec{s}\langle\sigma_\Delta\rangle\langle\vec{v_k}/x\rangle_{B_1}\langle\vec{u_k}/y\rangle_{B_2}) \\
&\twoheadrightarrow e^{i\theta_1} \cdot (\sum_{k=1}^{l} \gamma_k e^{i\rho_k} \vec{w_k})
\end{aligned}$$

Since $\vec{s}\langle\sigma_\Delta\rangle\langle\vec{v_k}/x\rangle_{B_1}\langle\vec{u_k}/y\rangle_{B_2} \in [\![\Delta, x_{B_1} : \sharp A_1, y_{B2} : \sharp A_2]\!]$, for every k, then $\vec{w_k} \in [\![C]\!]$. It remains to be seen that the term has norm-1, $\| \sum_{k=1}^{l} \gamma_k e^{i\rho_k} \vec{w_k} \| = 1$. For that, we observe:

$$\| \sum_{k=1}^{l} \gamma_k e^{i\rho_k} \vec{w_k} \|$$

$$
\begin{aligned}
&= \langle \sum_{k=1}^{l} \alpha_i e^{i\rho_k} \vec{w_k} \mid \sum_{k'=1}^{l} \gamma_{k'} e^{i\rho_{k'}} w_{\vec{k'}} \rangle \\
&= \sum_{k=1}^{l} \sum_{k'}^{l} \overline{\gamma_k e^{i\rho_k}}\ \gamma_{k'} e^{i\rho_{k'}} \langle \vec{w_k} \mid w_{\vec{k'}} \rangle \\
&= \sum_{k=1}^{l} \sum_{k'=1}^{l} \overline{\gamma_k e^{i\rho_k}}\ \gamma_{k'} e^{i\rho_{k'}} \langle \vec{v_k} \mid v_{\vec{k'}} \rangle \langle \vec{u_k} \mid u_{\vec{k'}} \rangle \quad \text{(from Lemma ??)} \\
&= \sum_{k=1}^{k} \sum_{k'=1}^{l} \overline{\gamma_k e^{i\rho_k}}\ \gamma_{k'} e^{i\rho_{k'}} \langle (\vec{u_k}, \vec{v_k}) \mid (u_{\vec{k'}}, v_{\vec{k'}}) \rangle \quad \text{(from Prop ??)} \\
&= \sum_{k=1}^{n} \overline{\gamma_k e^{i\rho_k}}\ \gamma_k e^{i\rho_k} \langle (\vec{v_k}, \vec{u_k}) \mid (\vec{v_k}, \vec{u_k}) \rangle \\
&\quad + \sum_{k,k'=1;k\neq k'}^{n} \overline{\gamma_k e^{i\rho_k}}\ \gamma_{k'} e^{i\rho_{k'}} \langle (\vec{v_k}, \vec{u_k}) \mid (v_{\vec{k'}}, u_{\vec{k'}}) \rangle \\
&= \sum_{k=1}^{n} \overline{\gamma_k e^{i\rho_k}}\ \gamma_k e^{i\rho_k} + 0 \\
&= \sum_{k=1}^{l} |\gamma_k|^2 |e^{i\rho_k}|^2 = 1
\end{aligned}
$$

Then $\sum_{i=1}^{n} \alpha_i \vec{w_i} \in [\![\sharp C]\!]$. Finally, we can conclude that:

$$(\mathsf{let}_{(B_1,B_2)}\ (x_1, x_2) = \vec{t}\ \mathsf{in}\ \vec{s})\langle\sigma\rangle \Vdash \sharp C$$

Case If the hypotheses are valid then:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathrm{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$
- For every $\sigma_\Gamma \in [\![\Gamma]\!]$, $\vec{t}\langle\sigma_\Gamma\rangle \Vdash B_{\{\vec{v_i}\}_{i=1}^n}$
- For every $i \in \{0, \cdots, n\}, \mathrm{dom}^\sharp(\Delta) \subseteq \mathrm{FV}(\vec{s_i}) \subseteq \mathrm{dom}(\Delta)$
- For every $i \in \{0, \cdots, n\}, \sigma_\Delta \in [\![\Delta]\!]$, $\vec{s_i}\langle\sigma_\Delta\rangle \Vdash A$

From this we can conclude that:

- $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathrm{FV}(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})$
- $\mathrm{FV}(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\}) \subseteq \mathrm{dom}(\Gamma, \Delta)$

Then, given $\sigma \in [\![\Gamma, \Delta]\!]$, we have that $\langle\sigma\rangle = \langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. Using the first hypothesis we have that, $\vec{t}\langle\sigma_\Gamma\rangle \twoheadrightarrow e^{i\theta_1} \cdot \vec{v_k}$ for some $k \in \{1, \cdots, n\}$. From the second hypothesis we have that $\vec{s_i}\langle\sigma_\Delta\rangle \twoheadrightarrow e^{i\rho_i} \cdot \vec{u_i} \in [\![A]\!]$ for $i \in \{1, \cdots, n\}$. Therefore:

$$
\begin{aligned}
&(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v}_1 \mapsto \vec{s_1} \mid \cdots \mid \vec{v}_n \mapsto \vec{s_n}\})\langle\sigma\rangle \\
&\quad = (\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v}_1 \mapsto \vec{s_1} \mid \cdots \mid \vec{v}_n \mapsto \vec{s_n}\})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle
\end{aligned}
$$

$$
\begin{aligned}
&= (\sum_{i=1}^{n} \alpha_i \mathsf{case}\ \vec{t}[\sigma_{\Gamma i}]\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma_\Delta\rangle \\
&\equiv (\mathsf{case}\ \sum_{i=1}^{n} \alpha_i \vec{t}[\sigma_{\Gamma i}]\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma_\Delta\rangle \\
&= (\mathsf{case}\ \vec{t}\langle\sigma_\Gamma\rangle\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma_\Delta\rangle \\
&\twoheadrightarrow (\mathsf{case}\ e^{i\theta_1} \cdot \vec{v_k}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma_\Delta\rangle \\
&\to e^{i\theta_1} \cdot (\vec{s_k}\langle\sigma_\Delta\rangle) \\
&\twoheadrightarrow e^{i\theta_1} \cdot (e^{i\rho_k} \cdot \vec{u_k}) \qquad \text{Where: } \vec{u_k} \in [\![A]\!] \\
&\equiv e^{i\theta} \cdot \vec{u_k} \qquad \text{With: } \theta = \theta_1 + \theta_2
\end{aligned}
$$

Since we pose no restriction on k, we can conclude that:

$$(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma\rangle \Vdash A$$

UnitCase If the hypotheses are valid, then:

- $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathrm{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma)$
- For every $\sigma_\Gamma \in [\![\Gamma]\!]$, $\vec{t}\langle\sigma_\Gamma\rangle \Vdash \sharp B_{\{\vec{v_i}\}_{i=1}^n}$
- For every i, $\mathrm{dom}^\sharp(\Delta) \subseteq \mathrm{FV}(\vec{s_i}) \subseteq \mathrm{dom}(\Delta)$
- For every $i \in \{0, \cdots, n\}, \sigma_\Delta \in [\![\Delta]\!]$, $\vec{s_i}\langle\sigma_\Delta\rangle \Vdash A$

From this we can conclude that:

- $\mathrm{dom}^\sharp(\Gamma, \Delta) \subseteq \mathrm{FV}(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v}_1 \mapsto \vec{s_1} \mid \cdots \mid \vec{v}_n \mapsto \vec{s_n}\})$
- $\mathrm{FV}(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v}_1 \mapsto \vec{s_1} \mid \cdots \mid \vec{v}_n \mapsto \vec{s_n}\}) \subseteq \mathrm{dom}(\Gamma, \Delta)$

Then, given $\sigma \in [\![\Gamma, \Delta]\!]$, we have that $\langle\sigma\rangle = \langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\sigma_\Delta \in [\![\Delta]\!]$. Using the first hypothesis we have that, $\vec{t}\langle\sigma_\Gamma\rangle \Vdash \sharp B_{\{\vec{v_i}\}_{i=1}^n}$, then $\vec{t}\langle\sigma_\Gamma\rangle \twoheadrightarrow e^{i\theta_1} \cdot \vec{u} = e^{i\theta_1} \cdot (\sum_{i=1}^n \beta_i \vec{v_i})$ where $\sum_{i=1}^n |\beta_i|^2$. From the second hypothesis we have that $\vec{s_i}\langle\sigma_\Delta\rangle \twoheadrightarrow e^{i\rho_i} \cdot \vec{u_i} \in [\![A]\!]$ for $i \in \{1, \cdots, n\}$ and $u_i \perp u_j$ if $i \neq j$. Therefore:

$$
\begin{aligned}
&(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma\rangle \\
&\quad= (\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma_\Gamma\rangle\langle\sigma_\Delta\rangle \\
&\quad= (\sum_{i=1}^{n} \alpha_i \mathsf{case}\ \vec{t}[\sigma_{\Gamma i}]\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma_\Delta\rangle \\
&\quad\equiv (\mathsf{case}\ \sum_{i=1}^{n} \alpha_i \vec{t}[\sigma_{\Gamma i}]\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma_\Delta\rangle \\
&\quad= (\mathsf{case}\ \vec{t}\langle\sigma_\Gamma\rangle\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma_\Delta\rangle \\
&\quad\twoheadrightarrow (\mathsf{case}\ e^{i\theta_1} \cdot \vec{u}\ \mathsf{of}\{\vec{v} \mapsto \vec{s_1} \mid \cdots \mid \vec{w} \mapsto \vec{s_2}\})\langle\sigma_\Delta\rangle \\
&\quad\to e^{i\theta_1} \cdot (\sum_{i=1}^{n} \beta_i s_i)\langle\sigma_\Delta\rangle
\end{aligned}
$$

$$
\begin{aligned}
&= e^{i\theta_1} \cdot (\sum_{j=1}^{n} \delta_j (\sum_{i=1}^{n} \beta_i \vec{s_i})[\sigma_{\Delta_j}]) \\
&= e^{i\theta_1} \cdot (\sum_{j=1}^{n} \delta_j (\sum_{i=1}^{n} \beta_i \vec{s_i}[[\sigma_{\Delta_j}]])) \\
&\equiv e^{i\theta_1} \cdot (\sum_{i,j=1}^{n} \beta_i \delta_j \vec{s_i}[\sigma_{\Delta_j}]) \\
&= e^{i\theta_1} \cdot (\sum_{i=1}^{n} \beta_i \vec{s_i} \langle \sigma_\Delta \rangle) \\
&\twoheadrightarrow e^{i\theta_1} \cdot (\sum_{i=1}^{n} \beta_i e^{i\rho_i} \cdot \vec{u_i})
\end{aligned}
$$

It remains to be seen that: $\| \sum_{i=1}^{n} \beta_i e^{i\rho_i} \cdot \vec{u_i} \| = 1$:

$$
\begin{aligned}
\| \sum_{i=1}^{n} \beta_i e^{i\rho_i} \cdot \vec{u_i} \| &= \langle \sum_{i=1}^{n} \beta_i e^{i\rho_i} \cdot \vec{u_i} \mid \sum_{i=1}^{n} \beta_i e^{i\rho_i} \cdot \vec{u_i} \rangle \\
&= \sum_{i,j=1}^{n} \overline{\beta_i e^{i\rho_i}} \beta_j e^{i\rho_j} \langle \vec{u_i} \mid \vec{u_j} \rangle \\
&= \sum_{i=1}^{n} \overline{\beta_i e^{i\rho_i}} \beta_i e^{i\rho_i} \langle \vec{u_i} \mid \vec{u_i} \rangle \\
&\quad + \sum_{i,j=1; i \neq j}^{n} \overline{\beta_i e^{i\rho_i}} \beta_j e^{i\rho_j} \langle \vec{u_i} \mid \vec{u_j} \rangle \\
&= \sum_{i=1}^{n} |\beta_i|^2 |e^{i\rho_i}|^2 + 0 \\
&= \sum_{i=1}^{n} |\beta_i|^2 = 1
\end{aligned}
$$

Then we can conclude that $\sum_{i=1}^{n} \beta_i e^{i\rho_i} \vec{u_i} \in [\![\sharp A]\!]$ and finally:

$$(\mathsf{case}\ \vec{t}\ \mathsf{of}\{\vec{v_1} \mapsto \vec{s_1} \mid \cdots \mid \vec{v_n} \mapsto \vec{s_n}\})\langle\sigma\rangle \Vdash \sharp A$$

Sum If the hypothesis is valid then for every i, $\mathrm{dom}^\sharp(\Gamma) \subseteq \mathrm{FV}(\vec{t_i}) \subseteq \mathrm{dom}(\Gamma)$.

From this we can conclude that $\mathrm{dom}^\sharp(\Gamma) \subseteq \sum_{i=1}^{n} \alpha_i \vec{t_i} \subseteq \mathrm{dom}(\Gamma)$. Given $\sigma \in [\![\Gamma]\!]$, we have for every i, $\vec{t_i}\langle\sigma\rangle \twoheadrightarrow e^{i\rho_i} \cdot \vec{v_i}$ where $\vec{v_i} \in [\![A]\!]$. Moreover, for every $i \neq j$, $\vec{v_i} \perp \vec{v_j}$ and $\sum_{i=1}^{n} |\alpha_i|^2 = 1$. Then:

$$(\sum_{i=1}^{n} \alpha_i \vec{t_i})\langle\sigma\rangle = \sum_{j=1}^{m} \beta_j (\sum_{i=1}^{n} \alpha_i \vec{t_i})[\sigma_j]$$

$$\begin{aligned}
&\equiv \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} \beta_j \vec{t_i}[\sigma_j] \\
&= \sum_{i=1}^{n} \alpha_i \vec{t_i}\langle\sigma\rangle \\
&\twoheadrightarrow \sum_{i=1}^{n} \alpha_i e^{i\rho_i} \vec{v_i}
\end{aligned}$$

It remains to be seen that $\| \sum_{i=1}^{n} \alpha_i e^{i\rho_i} \vec{v_i} \| = 1$. But:

$$\begin{aligned}
&\| \sum_{i=1}^{n} \alpha_i e^{i\rho_i} \vec{v_i} \| \\
&= \langle \sum_{i=1}^{n} \alpha_i e^{i\rho_i} \vec{v_i} \mid \sum_{i=1}^{n} \alpha_i e^{i\rho_i} \vec{v_i} \rangle \\
&= \sum_{i=i}^{n} \sum_{j=1}^{n} \overline{\alpha_i e^{i\rho_i}} \alpha_j e^{i\rho_j} \langle \vec{v_i} \mid \vec{v_j} \rangle \\
&= \sum_{i=1}^{n} \overline{\alpha_i e^{i\rho_i}} \alpha_i e^{i\rho_i} \langle \vec{v_i} \mid \vec{v_i} \rangle + \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \overline{\alpha_i e^{i\rho_i}} \alpha_j e^{i\rho_j} \langle \vec{v_i} \mid \vec{v_j} \rangle \\
&= \sum_{i=1}^{n} |\alpha_i|^2 |e^{i\rho_i}|^2 + 0 \\
&= \sum_{i=1}^{n} |\alpha_i|^2 = 1
\end{aligned}$$

Then we can conclude that $\sum_{i=1}^{n} \alpha_i e^{i\rho_i} \vec{v_i} \in [\![\sharp A]\!]$ and finally $(\sum_{i=1}^{n} \alpha_i \vec{t_i})\langle\sigma\rangle \Vdash \sharp A$.

Weak Given $\sigma \in [\![\Gamma, x_A : B]\!]$, we observe that $\langle\sigma\rangle = \langle\sigma_\Gamma\rangle\langle\vec{v}/x\rangle_A$ for some $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\vec{v} \in [\![B]\!]$. Using the first hypothesis, we know that $\vec{t}\langle\sigma_\Gamma\rangle \twoheadrightarrow e^{i\theta}\vec{w}$ where $\vec{w} \in [\![B]\!]$. Then we have:

$$\vec{t}\langle\sigma\rangle = \vec{t}\langle\sigma_\Gamma\rangle\langle\vec{v}/x\rangle_A \twoheadrightarrow e^{i\theta}\vec{w}\langle\vec{v}/x\rangle_A$$

Since $\vec{v} \in [\![A]\!]$, $\vec{w}\langle\vec{v}/x\rangle_A = \vec{w}[\vec{v}/x] = \vec{w}$ and $\vec{w} \in [\![B]\!]$, we can finally conclude that $\vec{t}\langle\sigma\rangle \Vdash B$.

Contr If the hypothesis is valid, we have that $\mathrm{dom}^\sharp(\Gamma, x_A : A, y_A : A) \subseteq \mathrm{FV}(\vec{t}) \subseteq \mathrm{dom}(\Gamma, x_A : A, y_A : A)$ and given $\sigma \in [\![\Gamma, x_A : A, y_A : A]\!]$, then $\vec{t}\langle\sigma\rangle \in [\![B]\!]$. Since we assume $\flat A$, we have that $\mathrm{dom}^\sharp(\Gamma, x_A : A, y_A : A) = \mathrm{dom}^\sharp(\Gamma, x_A : A)$. Therefore:

$$\mathrm{dom}^\sharp(\Gamma, x_A : A) \subseteq \mathrm{FV}(\vec{t})[x/y] \subseteq \mathrm{dom}(\Gamma, x_A : A)$$

Given $\sigma \in [\![\Gamma, x_A : A]\!]$, we observe that $\langle\sigma\rangle = \langle\vec{v}/x\rangle_A\langle\sigma_\Gamma\rangle$ with $\sigma_\Gamma \in [\![\Gamma]\!]$ and $\vec{v} \in [\![A]\!]$. Since $\vec{v} \in [\![A]\!]$, we know that

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : B_{\vec{v}} \vdash x : B_{\vec{v}}}\,(\text{Ax}) \quad \dfrac{}{y : B_{\vec{v}} \vdash y : B_{\vec{v}}}\,(\text{Ax})}{x : B_{\vec{v}}, y : B_{\vec{v}} \vdash (x, y) : B_{\vec{v}} \times B_{\vec{v}}}\,(\text{Pair})}{x : B_{\vec{v}} \vdash (x, x) : B_{\vec{v}} \times B_{\vec{v}}}\,(\text{Contr})}{\vdash \lambda x^{\vec{v}}.(x, x) : B_{\vec{v}} \Rightarrow (B_{\vec{v}} \times B_{\vec{v}})}\,(\text{Lam}) \qquad \dfrac{\vec{v} \in \{\vec{v}\}}{\vdash \vec{v} : B_{\vec{v}}}\,(\text{Ax})}{\vdash (\lambda x^{\vec{v}}.(x, x))\ \vec{v} : B_{\vec{v}} \times B_{\vec{v}}}\,(\text{App})$$

Table 1.7: *Duplication of a vector $\vec{v}$*

$\vec{t}[\vec{v}/z] = \vec{t}\langle \vec{v}/z \rangle_A$ for any variable z. Then we have:

$$\begin{aligned}
\vec{t}[x/y]\langle\sigma\rangle &= \vec{t}[x/y]\langle \vec{v}/x \rangle_A \langle \sigma_\Gamma \rangle \\
&= \vec{t}[x/y][\vec{v}/x]\langle \sigma_\Gamma \rangle \\
&= \vec{t}[\vec{v}/y][\vec{v}/x]\langle \sigma_\Gamma \rangle \\
&= \vec{t}\langle \vec{v}/y \rangle_A \langle \vec{v}/x \rangle_A \langle \sigma_\Gamma \rangle
\end{aligned}$$

Since $\langle \vec{v}/y \rangle_A \langle \vec{v}/x \rangle_A \langle \sigma \rangle \in [\![\Gamma, x_A : A, y_A : A]\!]$, we get:

$$\vec{t}\langle \vec{v}/y \rangle \langle \vec{v}/x \rangle_A \langle \sigma_\Gamma \rangle \twoheadrightarrow e^{i\theta}\vec{w} \in [\![B]\!]$$

Then we can finally conclude that $\vec{t}[x/y]\langle\sigma\rangle \Vdash B$.

Equiv It follows from definition and the fact that the reduction commutes with the congruence relation.

GlobalPhase It follows from the definition of type realizers.

□

1.4.4 Discussion: Towards a specification system

RELEO ESTA SUBSECCIÓN Y NO ESTOY SEGURO DE LA RELEVANCIA Y SI INCLUIRLA. ME GUSTARÍA CHARLARLO.

We found a somewhat interesting extreme when we drop the condition of bases to have size n, and we consider singletons. The resulting system replicates the term reduction inside the strict typing rules and forms a sort of specification system. In short, sequents take the following form: $x_1 : \vec{v}_1, \cdots, x_n : \vec{v}_n \vdash \vec{t} : \vec{w}$ (We omit the brackets in singleton sets). And so, we can read the previous sequent as: "Every substitution σ such that $\sigma(x_i) = \vec{v_i}$ validates that $\sigma(\vec{t})$, reduces to $\vec{w}$". With this in mind, we can design inference rules that only use singleton sets as types. Let us take for example any non-abstraction value distribution $\vec{v}$, we describe a specification with tight typing in table 1.7.

This result is not surprising. When dealing with classic computation, regardless of the basis, if we restrict the possible inputs to only one option, we can statically determine the output without having to reduce the term.

However, this changes when entangled qubits are involved. Entangled quantum states are states where a qubit cannot be described precisely and independently of another qubit. So for example the state $\mathsf{Bell} = \frac{1}{\sqrt{2}}((|0\rangle, |0\rangle) + (|1\rangle, |1\rangle))$ cannot be described as a pair of two separate values $\vec{v}$ and $\vec{w}$. We can however use the let construct to

destroy the pair, and in that case, we will have to type two separate variables. The typing judgement is thus:

$$\dfrac{\dfrac{\dfrac{\mathsf{Bell} \in \{\mathsf{Bell}\}}{\vdash \mathsf{Bell} : B_{\mathsf{Bell}}}\,(\text{Basis}) \qquad B_{\mathsf{Bell}} \leq \sharp(\mathbb{B}\times\mathbb{B})}{\vdash \mathsf{Bell} : \sharp(\mathbb{B}\times\mathbb{B})}\,(\text{Sub}) \qquad \dfrac{\dfrac{\dfrac{}{x : \sharp\mathbb{B} \vdash x : \sharp\mathbb{B}}\,(\text{Ax}) \qquad \dfrac{}{y : \sharp\mathbb{B} \vdash y : \sharp\mathbb{B}}\,(\text{Ax})}{x : \sharp\mathbb{B}, y : \sharp\mathbb{B} \vdash (x,y) : \sharp\mathbb{B}\times\sharp\mathbb{B}}\,(\text{Pair}) \qquad \sharp\mathbb{B}\times\sharp\mathbb{B} \leq \sharp(\mathbb{B}\times\mathbb{B})}{x : \sharp\mathbb{B}, y : \sharp\mathbb{B} \vdash (x,y) : \mathbb{B}\otimes\mathbb{B}}\,(\text{Sub})}{\vdash \mathsf{let}_{(\mathbb{B},\mathbb{B})}\ (x,y) = \mathsf{Bell}\ \mathsf{in}\ (x,y) : \sharp(\mathbb{B}\times\mathbb{B})}\,(\text{LetTens})$$

Despite knowing exactly which state we are referring to, there is an inherent loss of precision, and this itself is reflected in the type we are able to give. While this might seem as a limitation of the language, it is the desired outcome. This is due to the physical nature of the state, which cannot be described with two independent variables.

This mechanism outlines a possible implementation of a specification system similar to Hoare logic via realizability. It could be useful to reason about correctness in quantum programs. At the same time, we identify some limitations when analyzing terms which deal with entangled states. We let this development for future research.

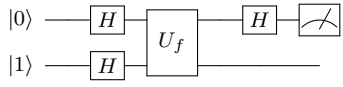
1.5 Examples

In this chapter we examine two use cases for the λ_B calculus. First, taking advantage of the basis types defined in the type algebra, we are able to give a more expressive type to the term encoding Deutsch's algorithm. Second, we make use of the deferred measurement principle and pattern matching from the case constructor to write a descriptive term encoding the quantum teleportation protocol.

1.5.1 Deutsch's algorithm

The Deutsch-Josza algorithm is a small example designed to showcase a problem which is solved exponentially faster by a quantum computer over a classical one. In it, we take as input a black box oracle which encodes a function $f : \{0,1\}^n \to \{0,1\}$. This function can be either *constant* or *balanced* (It outputs 0 for exactly half the inputs and 1 for the other half). The task to solve is to determine under which of the two classes the oracle falls.

In this section we will focus on the case where $n = 1$, the original formulation of Deutsch's algorithm. However, this results can be generalized to any arbitrary n. The quantum circuit implementing the algorithm is the following:



For a detailed discussion on the logic and operation of the algorithm, see (CITAR). We will do a comparison between Deutsch's algorithm written in different bases and see what information we can glean from the typing of the terms.

We first define the terms for the algorithm. The top level Deutsch abstraction, takes an oracle U_f which inputs two qubits $|xy\rangle$ and outputs $|x(y \oplus f(x))\rangle$ where $\oplus$ denotes addition modulo 2. The circuit will output $|0\rangle$ on the first qubit if the function f is balanced

On table 1.9 we note the four possible oracles. D_1 and D_4 correspond to the oracles encoding the 0 and 1 constant functions and D_2, D_3 to the identity and bit-flip respectively.

$$\mathsf{Deutsch} := (\lambda f_{\mathcal{P}} \,.\, \mathsf{let}_{(\mathbb{B},\mathbb{B})}\ (x,y) = (f(\mathbb{H}\,|0\rangle)(\mathbb{H}\,|1\rangle))\ \mathsf{in}\ ((\mathbb{H}x),y))$$
$$\mathbb{H} := \lambda x_{\mathbb{B}} \,.\, \mathsf{case}\ x\ \mathsf{of}\{|0\rangle \mapsto |+\rangle \mid |1\rangle \mapsto |-\rangle\}$$

Table 1.8: *Deutsch algorithm term*

$$\begin{aligned}
D_1 &:= \lambda x_{\mathbb{B}} \,.\, \lambda y_{\mathbb{B}} \,.\, (x,y)\\
D_2 &:= \lambda x_{\mathbb{B}} \,.\, \lambda y_{\mathbb{B}} \,.\, \mathsf{CNOT}\ x\ y\\
D_3 &:= \lambda x_{\mathbb{B}} \,.\, \lambda y_{\mathbb{B}} \,.\, \mathsf{CNOT}\ x\ (\mathsf{NOT}\ y)\\
D_4 &:= \lambda x_{\mathbb{B}} \,.\, \lambda y_{\mathbb{B}} \,.\, (x, (\mathsf{NOT}\ y))\\
&\text{Where:}\\
\mathsf{CNOT} &:= \lambda x_{\mathbb{B}} \,.\, \lambda y_{\mathbb{B}} \,.\, \mathsf{case}\ x\ \mathsf{of}\{|0\rangle \mapsto (|0\rangle\,, y) \mid |1\rangle \mapsto (|1\rangle\,, \mathsf{NOT}\ y)\}\\
\mathsf{NOT} &:= \lambda x_{\mathbb{B}} \,.\, \mathsf{case}\ x\ \mathsf{of}\{|0\rangle \mapsto |1\rangle \mid |1\rangle \mapsto |0\rangle\}
\end{aligned}$$

Table 1.9: *Oracles implementing the four possible functions* $f : \{0,1\} \mapsto \{0,1\}$

Each of these oracles can be typed as $\mathbb{B} \to \mathbb{B} \to (\mathbb{B} \times \mathbb{B})$. But since we are passing $|+\rangle$ and $|-\rangle$ as arguments, the typing we would be able to assign is: $\sharp\mathbb{B} \to \sharp\mathbb{B} \to (\mathbb{B} \otimes \mathbb{B})$. This means that the final typing for Deutsch would be: $\vdash \mathsf{Deutsch} : (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B} \Rightarrow (\sharp\mathbb{B} \Rightarrow \sharp\mathbb{B})) \Rightarrow \sharp(\mathbb{B} \times \mathbb{B})$. This would seem to suggest that the result of the computation is a superposition of pairs of booleans.

However, this approach underutilizes the amount of information we have available. We know that the oracle will receive specifically the state $|+-\rangle$, and so we can rewrite the intervening terms taking this information into account. In table 1.10 we restate the terms, but this time the abstractions and conditional cases are written in the basis $\mathbb{X} = \{|+\rangle\,, |-\rangle\}$.

In this case, for each of the oracles we can assign the type $\mathbb{X} \to \mathbb{X} \to \mathbb{X} \times \mathbb{X}$ and type the term Deutsch as $(\mathbb{X} \to \mathbb{X} \to \mathbb{X} \times \mathbb{X}) \to \mathbb{B}$. There is a key difference here, the type of the oracles assure us that the result will be in the basis state $\mathbb{X} \times \mathbb{X}$. In other words, the result can either be a pair of $|+\rangle$ or $|-\rangle$ (up to a global phase). Since we know this fact, we can manipulate the result of f as we would with classical bits, and discard the second component.

Both functions are equivalent on an operational point of view. But reframing the into a different basis, allow us to give a more tight typing to the terms and more insight on how the algorithm works. If we analize the typing judgements, we observe that none of the variables has a $\sharp$ type. This has two consequences, first we can safely discard the second qubit and second, the Hadamard transform guarantees that the first qubit will yield a boolean. This correlates with the fact that Deutsch's algorithm is deterministic and, we can statically ensure the result will be a basis vector.

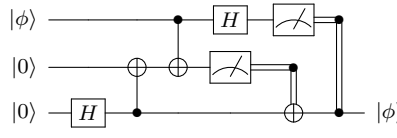
$$
\begin{array}{rcl}
\mathsf{Deutsch} & := & (\lambda U_{f_{\mathcal{P}}} \,.\, \mathsf{let}_{(\mathbb{X},\mathbb{X})}\ (x,y) = (U_f\ |+\rangle\ |-\rangle)\ \mathsf{in} \\
 & & \mathsf{case}\ x\ \mathsf{of}\{|+\rangle \mapsto |0\rangle \mid |-\rangle \mapsto |1\rangle\}) \\
D_1 & := & \lambda x_{\mathbb{X}} \,.\, \lambda y_{\mathbb{X}} \,.\, (x,y) \\
D_2 & := & \lambda x_{\mathbb{X}} \,.\, \lambda y_{\mathbb{X}} \,.\, \mathsf{CNOT}_{\mathbb{X}}\ x\ y \\
D_3 & := & \lambda x_{\mathbb{X}} \,.\, \lambda y_{\mathbb{X}} \,.\, \mathsf{CNOT}_{\mathbb{X}}\ x\ (\mathsf{NOT}_{\mathbb{X}}\ y) \\
D_4 & := & \lambda x_{\mathbb{X}} \,.\, \lambda y_{\mathbb{X}} \,.\, (x, (\mathsf{NOT}_{\mathbb{X}}\ y)) \\
\text{Where:} & & \\
\mathsf{CNOT}_{\mathbb{X}} & := & \lambda x_{\mathbb{X}} \,.\, \lambda y_{\mathbb{X}} \,.\, \mathsf{case}\ y\ \mathsf{of}\{ \\
 & & |+\rangle \mapsto (x, |+\rangle) \mid \\
 & & |-\rangle \mapsto (\mathsf{Z}_{\mathbb{X}} x, |-\rangle) \\
 & & \} \\
\mathsf{Z}_{\mathbb{X}} & := & \lambda x_{\mathbb{X}} \,.\, \mathsf{case}\ x\ \mathsf{of}\{|+\rangle \mapsto |-\rangle \mid |-\rangle \mapsto |+\rangle\} \\
\mathsf{NOT}_{\mathbb{X}} & := & \lambda x_{\mathbb{X}} \,.\, \mathsf{case}\ x\ \mathsf{of}\{|+\rangle \mapsto |+\rangle \mid |-\rangle \mapsto (-1)\cdot|-\rangle\}
\end{array}
$$

Table 1.10: *Deutsch term and oracles in the shifted Hadamard basis.*

1.5.2 Quantum teleportation

The *principle of deferred measurement* states that any quantum circuit involving the measurement of some qubit followed by a gate controlled by the outcome of this measurement is equivalent to another circuit in which no gates are controlled by previous measurement results. Instead, all gates are controlled by a quantum state. λ_B does not implement a mechanism to measure states, but using the case constructor is possible to simulate these quantum controlled gates.

A notable example of an algorithm which makes use of classical controlled gates is the *quantum teleportation*. In it, two agents (usually called Alice and Bob) share two parts of a Bell state and make use of the entanglement to move a quantum state from a qubit owned by Alice to a qubit owned by Bob. The quantum circuit representation of the algorithm is the following:



The algorithm first encodes the Bell state Φ^+ onto the second and third qubit and then performs a Bell basis measurement on the first and second qubit. In order to do this, it first decomposes applying a CNOT gate followed by a Hadamard gate on the first qubit (The adjoint of the Bell state generation). Then the first and second qubit are measured, which informs the correction needed for the third qubit to recover the state ϕ.

We can simulate the operation of the algorithm, via a λ-term which instead of outright measuring, describes the steps to take in each of the possible outcomes. A possible implementation is:

$$(\lambda x_{\mathbb{B}} \,.\, \mathsf{let}_{(\mathbb{B},\mathbb{B})}\ (y_1,y_2) = \Phi^+\ \mathsf{in}\ \ \mathsf{case}(x,y_1)\ \mathsf{of}\ \{\Phi^+ \mapsto (\Phi^+, y_2)$$

$$\Phi^- \mapsto (\Phi^-, Zy_2)$$
$$\Psi^+ \mapsto (\Psi^+, Xy_2)$$
$$\Psi^- \mapsto (\Psi^-, ZXy_2)$$
$$\})$$

The λ-term takes the state $|\phi\rangle$ as an argument, then matches the first qubit of the Bell pair and the $|\phi\rangle$ qubit, with the vectors of the Bell basis. In each branch, corrects the third qubit to recover the original $|\phi\rangle$ state. This is akin to controlling the correction with each of the Bell basis vectors.

The λ_B calculus provides syntax which allows the abstraction of the steps encoding and decoding on the Bell basis. This technique makes full use of the deferred measurement principle and can be applied to measurements on arbitrary bases. The final type of the term is $\sharp B_{\mathbb{B}} \Rightarrow \sharp B_{\mathsf{Bell}} \times \sharp B_{\mathbb{B}}$.

BUSCAR OTROS EJEMPLOS DE ALGORITMOS QUE HAGAN MEDICIONES EN BASES NO COMPUTACIONALES.

1.6 Conclusion

In this chapter we explore a quantum-data/quantum-control λ-calculus, with the additional feature of framing the abstraction in different bases besides the canonical one.

The mechanism needed to implement this idea is the decoration in λ-terms and let constructors. Along with a new substitution which dictates the decomposition of the value distribution onto different bases. These changes do not add expressive power to the original calculus it is based from, however they provide a different point of view when writing programs.

The reduction system itself orchestrates the computation and makes use of the syntax and substitution previously defined. The main point to note is that the evaluation commutes with the congruence relationship, ensuring that interpreting a vector in a different basis does not alter the result of the computation. And in turn, allowing to consider value distributions modulo this congruence.

The previous work pays its dividends when considering the realizability model. The inclusion of the atomic types B_X, is used to characterize the abstractions that represent unitary functions. This is the main result of the section and is a generalization of the characterization found in [1]. Here, the use of basis types gives way to a simpler proof.

The other main result of the chapter is the validity of the several typing rules described in table 1.6. Extracting them via the realizability technique, ensures their correctness and can later form the foundation of the type system for a programming language.

Finally, we present two examples that showcase the advantage of the typing system and syntax. First Deutsch's algorithm, which exhibits a more expressive type and in turn, allows to treat the result classically. Second, the case for quantum teleportation, where we are able to gates controlled by a Bell basis measurement as branches on a pattern matching case.

There are a few remaining lines of research that stem from this work. A natural progression would be to provide a categorical model

to study the calculus through a different lens and relate it to other well studied systems.

On the other side, we could also try to give a translation into an intermediate language like ZX alongside the lines of the second chapter. Proving that, despite the programs being detached from the circuitry, they can still be implemented concretely.

Lastly, we found that the realizability technique could prove to be a promising foundation for a specification system. In short, there are several opportunities to expand and make use of this work.

List of Figures

List of Tables

Bibliography

[1] Alejandro Díaz-Caro, Mauricio Guillermo, Alexandre Miquel, and Benoît Valiron. Realizability in the unitary sphere. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2019)*, pages 1–13, 2019. 1.1, 1.4.2, 1.4.2, 1.6