University of Minho
School of Engineering

Bruna Filipa Martins Salgado

# A Metric Equational System for Quantum Computation

August 2024

**University of Minho**
School of Engineering

Bruna Filipa Martins Salgado

# A Metric Equational System for Quantum Computation

Master's Dissertation
Master in Physics Engineering

Work carried out under the supervision of
**Renato Jorge Araújo Neves**

August 2024

*cos. Esta licença costuma ser comparada com as licenças de software livre e de código aberto «copyleft». Todos os trabalhos novos baseados no seu terão a mesma licença, portanto quaisquer trabalhos derivados também permitirão o uso comercial. Esta é a licença usada pela Wikipédia e é recomendada para materiais que seriam beneficiados com a incorporação de conteúdos da Wikipédia e de outros projetos com licenciamento semelhante.]*

**CC BY-ND**

https://creativecommons.org/licenses/by-nd/4.0/ *[Esta licença permite que outras pessoas usem o seu trabalho para qualquer fim, incluindo para fins comerciais. Contudo, o trabalho, na forma adaptada, não poderá ser partilhado com outras pessoas e têm que lhe ser atribuídos os devidos créditos.]*

**CC BY-NC**

https://creativecommons.org/licenses/by-nc/4.0/ *[Esta licença permite que outros remisturem, adaptem e criem a partir do seu trabalho para fins não comerciais, e embora os novos trabalhos tenham de lhe atribuir o devido crédito e não possam ser usados para fins comerciais, eles não têm de licenciar esses trabalhos derivados ao abrigo dos mesmos termos.]*

**CC BY-NC-SA**

https://creativecommons.org/licenses/by-nc-sa/4.0/ *[Esta licença permite que outros remisturem, adaptem e criem a partir do seu trabalho para fins não comerciais, desde que lhe atribuam a si o devido crédito e que licenciem as novas criações ao abrigo de termos idênticos.]*

**CC BY-NC-ND**

https://creativecommons.org/licenses/by-nc-nd/4.0/ *[Esta é a mais restritiva das nossas seis licenças principais, só permitindo que outros façam download dos seus trabalhos e os compartilhem desde que lhe sejam atribuídos a si os devidos créditos, mas sem que possam alterá- los de nenhuma forma ou utilizá-los para fins comerciais.]*

# Acknowledgements

Write your acknowledgements here. Do not forget to mention the projects and grants that you have benefited from while doing your research, if any. Ask your supervisor about the specific textual format to use. (Funding agencies are quite strict about this.)

# Statement of Integrity

I hereby declare having conducted this academic work with integrity.
I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.
I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, Braga, August 2024

Bruna Filipa Martins Salgado

# Abstract

Noisy intermediate-scale quantum (NISQ) computers are expected to operate with severely limited hardware resources. Precisely controlling qubits in these systems comes at a high cost, is susceptible to errors, and faces scarcity challenges. Therefore, error analysis is indispensable for the design, optimization, and assessment of NISQ computing. Nevertheless, the analysis of errors in quantum programs poses a significant challenge. The overarching goal of the M.Sc. project is to provide a fully-fledged quantum programming language on which to study metric program equivalence in various scenarios, such as in quantum algorithmics and quantum information theory.

**Keywords** approximate equivalence, $\lambda$-calculus, metric equations

# Resumo


Escrever aqui o resumo (pt)



**Palavras-chave** palavras, chave, aqui, separadas, por, vírgulas

# Contents


**Acronyms** xv

**Notation** xvii

**1 Introduction** 1

1.1 Motivation and Context . . . . . 1
1.2 Goals . . . . . 4
1.3 Document Structure . . . . . 4

**2 Metric Lambda Calculus** 5

2.1 The Lambda Calculus . . . . . 5
2.2 Syntax . . . . . 7
2.2.1 Type system . . . . . 7
2.2.2 (Raw)Terms . . . . . 8
2.2.3 Free and Bound Variables . . . . . 8
2.2.4 Term formation rules . . . . . 9
2.2.5 $\alpha$-equivalence . . . . . 11
2.2.6 Substitution . . . . . 12
2.2.7 Properties . . . . . 13
2.2.8 Equations-in-context . . . . . 14
2.3 Metric equational system . . . . . 15

**3 Quantum Meets Lambda Calculus** 17

3.1 Mathematical Preliminaries . . . . . 17
3.1.1 Complex vector spaces . . . . . 18
3.1.2 Linear operators . . . . . 20
3.1.3 Inner product . . . . . 20

3.1.4 Norm and normed spaces . . . 21
3.1.5 Important classes of operators/matrices . . . 22
3.1.6 Useful norms . . . 23
3.1.7 Eigenvectors, eigenvalues and the spectral theorem . . . 24
3.1.8 Tensor Products and Direct Sums of Complex spaces . . . 25
3.2 Quantum Computing Preliminaries . . . 25
3.2.1 The 2-Dimensional Hilbert Space . . . 26
3.2.2 Multi-qubit States . . . 27
3.3 Quantum Lambda Calculus:Interpretation . . . 30

**4 Conditionals** **33**
4.1 Measurements . . . 33
4.1.1 Example: Deutsch's Algorithm . . . 33
4.2 Conditionals . . . 37
4.2.1 Syntax . . . 37
4.2.2 Interpretation . . . 43
4.2.3 $\beta$ and $\eta$ Equations . . . 48
4.2.4 Metric equations . . . 52
4.2.5 Quantum Teleportation . . . 60
4.2.6 Ilustration: Noisy Quantum Teleportation . . . 63
4.3 Discard Operation . . . 75
4.3.1 Example: Proving an equivalence using the discard equation-in-context 75
4.3.2 Ilustration: A malicious attack on the quantum teleportation protocol 75

**5 Graded modalities** **83**
5.1 Syntax . . . 83
5.2 Interpretation . . . 84
5.3 Quantum State Discrimination . . . 84

**6 Conclusions and future work** **85**
6.1 Conclusions . . . 85
6.2 Prospect for future work . . . 85

# List of Figures


1 Term formation rules of affine lambda calculus. . . . . . . . . . . . . . . . . . 10
2 Equations-in-context for affine lambda calculus . . . . . . . . . . . . . . . . . 14
3 Metric equational system . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

4 Bloch sphere representation of a qubit . . . . . . . . . . . . . . . . . . . . . 27
5 Judgment interpretation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
6 Judgment interpretation of the operations in quantum lambda calculus. . . . 32

7 Quantum circuit implementing Deutsch's algorithm . . . . . . . . . . . . . . . 34
8 Term formation rules for conditionals . . . . . . . . . . . . . . . . . . . . . . 37
9 Judgment interpretation for conditionals . . . . . . . . . . . . . . . . . . . . 43
10 Metric equational system for condicionals . . . . . . . . . . . . . . . . . . . 52
11 Quantum Teleportation Protocol . . . . . . . . . . . . . . . . . . . . . . . . 60
12 Quantum Teleportation Protocol: Dephasing with probability $p$ after EPR pair creation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
13 Quantum Teleportation Protocol: Amplitude Dampling with probability $\gamma$ after Correction. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
14 Quantum Teleportation Protocol: Erroneous implementation of the Hadamard gate. $H^\epsilon$ is regarded as the composition $R_y(\frac{\pi}{2}) \cdot P(\pi + \epsilon)$. . . . . . . . . . . 69
15 T operation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
16 Quantum Teleportation Protocol: Bit flip with 50% probability before measurement. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77

17 Term formation rules of graded lambda calculus. . . . . . . . . . . . . . . . . 83

# Acronyms

**NISQ** Noisy Intermediate-Scale Quantum 1, 2, 3

**BNF** Backus-Naur Form 7, 8

# Notation

$FV(v)$ Set of free variables of a term $v$. 8

$v : \mathbb{A}$ Typed term. 9

$\Gamma, \Delta, E$ Typical names for typing contexts. 9

$\Gamma \triangleright v : \mathbb{A}$ Typing judgement. 9

$v[w/x]$ Substitution of a variable $x$ for a term $w$ in a term $v$. 12

$\Gamma \triangleright v = w : \mathbb{A}$ Equation-in-context. 14

$t =_\epsilon s$ Metric equation. 15

$\mathbb{C}^n$ n-dimensional complex space. 18

$\mathbb{C}^{n \times m}$ Space of complex matrices of dimension $n \times m$. 18

$R, V, W$ Typical names for vector spaces. 18

$\langle \cdot, \cdot \rangle$ Inner product. 20

$(-)^*$ Complex conjugate operation. 21

$|| \cdot ||$ Norm of an arbitrary vector. 21

$(-)^\dagger$ Adjoint operation. 22

$U$ Typical name for a unitary operator. 23

$\rho, \sigma$ Typical designations for density matrices. 23

$||.||_2$ Euclidean norm. 23

$||.||_2$ Trace norm. 23

$\|.\|_\sigma$ Operator norm. 23

$(-)^{\otimes n}$ N-fold tensor product. 25

$|\psi\rangle$ Quantum state. Also known as ket. 25

$\langle\psi|$ $|\psi\rangle^\dagger$. Also known as bra. 25

$\langle\psi|\phi\rangle$ Inner product between states $|\psi\rangle$ and $|\phi\rangle$. 25

$|\psi\rangle \otimes |\phi\rangle$ Tensor product of states $|\psi\rangle$ and $|\phi\rangle$. 27

$|\psi\rangle\,|\phi\rangle$ Tensor product of states $|\psi\rangle$ and $|\phi\rangle$. 27

$|\psi\phi\rangle$ Tensor product of states $|\psi\rangle$ and $|\phi\rangle$. 27

# Chapter 1

# Introduction

## 1.1 Motivation and Context

Quantum computing dates back to 1982 when Nobel laureate Richard Feynman proposed the idea of constructing computers based on quantum mechanics principles to efficiently simulate quantum phenomena Feynman [1982].

The field has since evolved into a multidisciplinary research area that combines quantum mechanics, computer science, and information theory. Quantum information theory, in particular, is based on the idea that if there are new physics laws, there should be new ways to process and transmit information. In classical information theory, all systems (computers, communication channels, etc.) are fundamentally equivalent, meaning they adhere to consistent scaling laws. These laws, therefore, govern the ultimate limits of such systems. For instance, if the time required to solve a particular problem, such as the factorization of a large number, increases exponentially with the size of the problem, this scaling behavior remains true irrespective of the computational power available. Such a problem, growing exponentially with the size of the object, is known as a ”difficult problem”. However, as demonstrated by Peter Shor, the use of a quantum computer with a sufficient number of quantum bits (qubits) could significantly accelerate the factorization of large numbers Shor [1994]. This advancement poses a significant threat to the security of confidential data transmitted over the Internet, as the RSA algorithm is based on the computational difficulty of factorizing large numbers.

The quantum computing paradigm holds immense promise, as evidenced by this compelling result in computational complexity theory. While hardware advancements have brought the scientific community closer to realizing this potential, the ultimate goal is yet to be accomplished. A **Noisy Intermediate-Scale Quantum** (**NISQ**) computer equipped with 50-100 qubits may surpass the capabilities of current classical computers, yet the impact of quantum noise, such as decoherence in entangled states, imposes limitations on the size of quantum circuits that can be executed reliably Preskill [2018]. Unfortunately, general-purpose error correction techniques Calderbank and Shor [1996]; Gottesman [1997]; Steane [1996] consume a substantial number of qubits, making it difficult for **NISQ** devices to make use of them in the near term. For instance, the implementation of a single logical qubit may require between $10^3$ and $10^4$ physical qubits Fowler et al. [2012]. As a result, it is unreasonable to expect that the idealized quantum algorithm will run perfectly on a quantum device, instead only a mere approximation will be observed.

To reconcile quantum computation with **NISQ** computers, quantum compilers perform transformations for error mitigation Wallman and Emerson [2016] and noise-adaptive optimization Murali et al. [2019]. Additionally, current quantum computers only support a restricted, albeit universal, set of quantum operations. As a result, nonnative operations must be decomposed into sequences of native operations before execution Harrow et al. [2002], Burgholzer and Wille [2020]. In general, perfect computational universality is not sought, but only the ability to approximate any quantum algorithm, with a preference for minimizing the use of additional gates beyond the original requirements. The assessment of these compiler transformations necessitates a comparison of the error bounds between the source and compiled quantum programs. Additionally, in quantum information theory, it is essential to account for errors arising from malicious attacks or noisy channels Watrous [2018].

This suggests the development of appropriate notions of approximate program equivalence, *in lieu* of the classical program equivalence and underlying theories that typically hinge on the idea that equivalence is binary, *i.e.* two programs are either equivalent or they are not Winskel [1993].

As previously noted, Shor's algorithm has played a pivotal role in sparking heightened interest within the scientific community toward quantum computing research. Several quantum programming languages have surfaced over the past 25 years Zhao [2020]; Serrano et al. [2022]. These include imperative languages such as Qiskit Qiskit contributors [2023] and Silq Bichsel et al. [2020], as well as functional languages such as Quipper Green et al. [2013] and Q# Svore et al. [2018]. On one hand, the design of quantum programming languages is strongly oriented towards implementing quantum algorithms. On the other hand, the definition of functional paradigmatic languages or functional calculi serves as a valuable tool for delving into theoretical aspects of quantum computing, particularly exploring the foundational basis of quantum computation Zorzi [2016].

Most of the current research on algorithms and programming languages assumes that addressing the challenge of noise during program execution will be resolved either by the hardware or through the implementation of fault-tolerant protocols designed independently of any specific application Chong et al. [2017]. As previously stated, this assumption is not realistic in the **NISQ** era. Nonetheless, there have been efforts to address the challenge of approximate program equivalence in the quantum setting.

Hung et al. [2019] and Tao et al. [2021] reason about the issue of noise in a quantum while-language by developing a deductive system to determine how similar a quantum program is from its idealised, noise-free version. The former introduces the ($Q$,$\lambda$)-diamond norm which analyzes the output error given that the input quantum state satisfies some quantum predicate $Q$ to degree $\lambda$. However, it does not specify any practical method for obtaining non-trivial quantum predicates. In fact, the methods used in [Hung et al. [2019]] cannot produce any post conditions other than $(I, 0)$ (*i.e.*, the identity matrix $I$ to degree 0, analogous to a "true" predicate) for large quantum programs. The latter specifically addresses and delves into this aspect.

An alternative approach was explored in Dahlqvist and Neves [2023a], using linear $\lambda$-calculus as basis – *i.e* programs are written as linear $\lambda$-terms – which has deep connections to both logic and category theory Girard et al. [1995], Benton [1994]. A notion of approximate equivalence is then integrated in the calculus via the so-called diamond norm , which induces a metric (roughly, a distance function) on the space of quantum programs (seen semantically as completely positive trace-preserving super-operators) [Watrous [2018]]. The authors argue that their deductive system allows to compute an approximate distance between two quantum programs easily as opposed to computing an exact distance "semantically" which tends to involve quite complex operators. Some positive results were achieved in this setting, but much remains to be done.

Agora falta falar das copias : existe uma variedade de problemas in inf quantica que envolvem copias de estados, (discriminação, quantum hyp testing, metrology), e portanto o nosso framework deve ser capaz de lidar com isso.

## 1.2 Goals

## 1.3 Document Structure

# Chapter 2

# Metric Lambda Calculus

The Lambda Calculus, developed by Church and Curry in the 1930s, serves as a formal language capturing the key attribute of higher-order functional languages, treating functions as first-class citizens, allowing them to be passed as arguments Barendregt et al. [1984]. Moreover, lambda calculus has been proven to be universal in the sense that any computable function can be represented as an expression within the language Bernays [1936] . Beyond its foundational aspects, this calculus incorporates extensions for modeling side effects, including probabilistic or non-deterministic behaviors and shared memory. Higher-order functions form a pivotal abstraction in practical programming languages such as LISP, Scheme, ML, and Haskell.

This chapter introduces the metric lambda calculus as presented in Dahlqvist and Neves [2023a]. The metric lambda calculus integrates notions of approximation into the equational system of affine lambda calculus, a variant of lambda calculus that restricts each variable to being used at most once. The metric lambda calculus incorporates a metric equational system, enabling reasoning about approximate program equivalence. This chapter offers a brief insight into lambda calculus and an overview of the syntax and metric equational system of the metric lambda calculus. For a more detailed study of lambda calculus theory, the reader is referred to Barendregt et al. [1984].

## 2.1 The Lambda Calculus

The concept of a function takes a central role in the lambda calculus. But what exactly is a function? In most mathematics, the "functions as graphs" paradigm the "functions as graphs" paradigm is the most elegant and appropriate framework for understanding functions. Within this paradigm, each function $f$ has a fixed domain $X$ and a fixed codomain $Y$.

The function $f$ is then a subset of $X \times Y$ that satisfies the property that for each $x \in X$ there is a unique $y \in Y$ such that $(x, y) \in f$. Two functions $f$ and $g$ are equal if they yield the same output on each input, that is if $f(x) = g(x)$ for all $x \in X$. This perspective is known as the extensional view of functions, as it emphasizes that the only observable property of a function is how it maps inputs to outputs.

On the other hand, the "functions as rules" paradigm is more appropriate within computer science. In this context, defining a function involves specifying a rule or procedure for computing the function. Such a rule is often expressed in the form of a formula, for example, $f(x) = x^2$. As with the mathematical paradigm, two functions are considered extensionally equal if they exhibit the same input-output behavior. However, this view also introduces the notion of intensional equality: two functions are intensionally equal if they are defined by (essentially) the same formula.

In the lambda calculus, functions are described explicitly as formulae. The function $f : x \mapsto f(x)$ is represented as $\lambda x.f(x)$. Applying a function to an argument is done by juxtaposing the two expressions. For instance consider the function $f : x \mapsto x + 1$, to compute $f(2)$ one writes $(\lambda x.x + 1)(2)$.

The expression of higher-order functions - functions whose inputs and/or outputs are themselves functions- in a simple manner is an essential feature of lambda calculus. For example, the composition operator $f, g \mapsto f \circ g$ is written as $\lambda f.\lambda g.\lambda x.f(g(x))$. Considering the functions $f : x \mapsto x^2$ and $g : x \mapsto x + 1$, to compute $(f \circ g)(2)$ one writes

$$(\lambda f.\lambda g.\lambda x.f(g(x)))(\lambda x.x^2)(\lambda x.x + 1)(2).$$

As mentioned above, within the "functions as rules" paradigm, is not always necessary to specify the domain and codomain of a function in advance. For instance, the identity function $f : x \mapsto x$, can have any set $X$ as its domain and codomain, provided that the domain and codomain are the same. In this case, one says that $f$ has type $X \to X$. In the case of the composition operator, $h = \lambda f.\lambda g.\lambda x.f(g(x))$, the domain and codomain of the functions $f$ and $g$ must match. Specifically, $f$ can have any set $X$ as its domain and any set $Y$ as its codomain, provided that $Y$ is the domain of $g$. Similarly, $g$ can have any set $Z$ as its codomain. Thus, $h$ has type

$$(X \to Y) \to (Y \to Z) \to (X \to Z).$$

This flexibility regarding domains and codomains enables operations on functions that are not possible in ordinary mathematics. For instance, if $f = \lambda x.x$ is the identity function, then one has that $f(x) = x$ for any $x$. In particular, by substituting $f$ for $x$, one obtains $f(f) = (\lambda x.x)(f) = f$. Note that the equation $f(f) = f$ is not valid in conventional mathematics, as it is not permissible, due to set-theoretic constraints, for a function to belong to its own domain.

Nevertheless, this remarkable aspect of lambda calculus, this work focuses on a more restricted version of the lambda calculus, known as the simply-typed lambda calculus. Here, each expression is always assigned a type, which is very similar to the situation in mathematics. A function may only be applied to an argument if the argument's type aligns with the function's expected domain. Consequently, terms such as $f(f)$ are not allowed, even if $f$ represents the identity function.

## 2.2 Syntax

The grammar and term formation rules of the affine lambda calculus, discussed in Dahlqvist and Neves [2023a], are presented in this subsection.

### 2.2.1 Type system

As previously mentioned, this work focuses on the simply-typed lambda calculus, this work focuses on the simply-typed lambda calculus, where each lambda term is assigned a *type*. Unlike sets, types are *syntactic* objects, meaning they can be discussed independently of their elements. One can conceptualize types as names or labels for set. The definition of the grammar of types for affine lambda calculus is as follows, where $G$ represents a set of ground types, is given by the following **Backus-Naur Form** (Backus et al. [1960]).

$$\mathbb{A} ::= X \in G \mid \mathbb{I} \mid \mathbb{A} \otimes \mathbb{A} \mid \mathbb{A} \multimap \mathbb{A} \tag{2.1}$$

Note that this is an inductive definition. Ground types are things such as booleans, integers, and so forth. The type $\mathbb{I}$ is the unit/empty type, which has only one element. The type $\mathbb{A} \otimes \mathbb{A}$ correponds to the tensor of two types, while the type $\mathbb{A} \multimap \mathbb{B}$ is the type of linear maps one type to another.

### 2.2.2 (Raw)Terms

The expressions of the lambda calculus are called lambda terms. In the simply-typed lambda calculus, each lambda term is assigned a type. The terms without the specification of a type are called *raw typed lambda terms*. The grammar of *raw typed lambda terms* is given by the **BNF** below.

$$
\begin{aligned}
v, v_1, \ldots, v_n, w \quad ::= \quad & x \mid f(v_1, \ldots, v_n) \mid \ * \ \mid (\lambda x : \mathbb{A}.v) \mid (vw) \mid v \otimes w \mid \\
& \text{pm } v \text{ to } x \otimes y.w \mid v \ \text{ to } \ * .w \mid \text{dis}(v)
\end{aligned}
$$

Here $x$ ranges over an infinite set of variables. $f \in \Sigma$, where $\Sigma$ corresponds to a class of sorted operation symbols, and $f(v_1, \ldots, v_n)$ corresponds to the aplication of the function $f$ to the arguments $v_1, \ldots, v_n$. The symbol $*$ is the unit element of the type $\mathbb{I}$. The term $(\lambda x : \mathbb{A}.v)$ is the lambda abstraction term, which represents a function that takes an argument of type $\mathbb{A}$ and returns the value of $v$. The term $(vw)$ is the application term, which applies the function $v$ to the argument $w$. The term $v \otimes w$ is the tensor product of $v$ and $w$. The term pm $v$ to $x \otimes y.w$ is the pattern-matching construct, which is used to deconstruct a tensor product into components $x$ and $y$. The term $v$ to $*.w$ is used to discard a variable $v$ of the unit type. The term dis$(v)$ is the discard term, which is used to discard a term $v$.

Ver o que por antes do ::= porque v1,..., vn tb são termos

### 2.2.3 Free and Bound Variables

An occurrence of a variable $x$ within a term of the form $\lambda x.v$ is referred to as *bound*. Similarly, the variables $x$ and $y$ in the term pm $v$ to $x \otimes y.w$ are also bound. A variable occurrence that is not bound is said to be *free*. For example, in the term $\lambda x.xy$, the variable $y$ is free, whereas the variable $x$ is bound.

The set of free variables of a term $v$ is denoted by $FV(v)$, and is defined inductively as follows:

$$
\begin{aligned}
FV(x) &= \{x\}, & FV(*) &= \emptyset, \\
FV(f(v_1, \ldots, v_n)), &= FV(v_1) \cup \ldots \cup FV(v_n) & FV(\lambda x : \mathbb{A}.v) &= FV(v) \backslash \{x\}, \\
FV(vw) &= FV(v) \cup FV(w), & FV(v \otimes w) &= FV(v) \cup FV(w), \\
FV(\text{pm } v \text{ to } x \otimes y.w), &= FV(v) \cup (FV(w) \backslash \{x, y\}) & FV(\text{dis}(v)) &= FV(v), \\
FV(v \text{ to } *.w) &= FV(v) \cup FV(w).
\end{aligned}
$$

### 2.2.4 Term formation rules

To prevent the formation of nonsensical terms within the context of lambda calculus, such as $(v \otimes w)(u)$, the *typing rules* are imposed.

A *typed* term is a pair consisting of a term and its corresponding type. The notation $v : \mathbb{A}$ denotes that the term $v$ has type $\mathbb{A}$. Typing rules are formulated using *typing judgments*. A typing judgment is an expression of the form $x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n \triangleright v : \mathbb{A}$ (where $n \geq 1$), which asserts that the term $v$ is a well-typed term of type $\mathbb{A}$ under the assumption that each variable variable $x_i$ has type $\mathbb{A}_i$, for $1 \leq i \leq n$. The list $x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n$ of typed variables is called the *typing context* of the judgment, and it might be empty. Each variable $x_i$ (where $1 \leq i \leq n$) must occur at most once in $x_1, \ldots, x_n$. The typing contexts are denoted by Greek letters $\Gamma, \Delta, E$, and from now on, when referring to an abstract judgment, the notation $\Gamma \triangleright v : \mathbb{A}$ will be employed. The empty context is denoted by $-$. Note that in the affine lambda calculus, different contexts do not share variables. For example, if $\Gamma = x : \mathbb{A}, y : \mathbb{B}$ none of these variables can appear in any other context.

The concept of *shuffling* is employed to construct a linear typing system that ensures the admissibility of the exchange rule and enables unambiguous reference to judgment's denotation $[\![\Gamma \triangleright v : \mathbb{A}]\!]$. An admissible rule is not explicitly included in the formal definition of type theory, but its validity can be proven by demonstrating that whenever the premises can be derived, it is possible to construct a derivation of its conclusion. Shuffling is defined as a permutation of typed variables in a sequence of contexts, $\Gamma_1, \ldots, \Gamma_n$, preserving the relative order of variables within each $\Gamma_i$ Shulman [2021]. For instance, if $\Gamma_1 = x : \mathbb{A}, y : \mathbb{B}$ and $\Gamma_2 = z : \mathbb{C}$, then $z : \mathbb{C}, x : \mathbb{A}, y : \mathbb{B}$ is a valid shuffle of $\Gamma_1, \Gamma_2$. On the other hand, $y : \mathbb{B}, x : \mathbb{A}, z : \mathbb{C}$ is not a shuffle because it alters the occurrence order of $x$ and $y$ in $\Gamma_1$. The set of shuffles in $\Gamma_1, \ldots, \Gamma_n$ is denoted as $\mathsf{Sf}(\Gamma_1, \ldots, \Gamma_n)$. A valid typing derivation is constructed using the inductive rules shown in Figure 1.

$$\frac{\Gamma_i \triangleright v_i : \mathbb{A}_i \quad f : \mathbb{A}_1, \ldots, \mathbb{A}_n \to \mathbb{A} \in \Sigma \quad E \in \mathsf{Sf}(\Gamma_1; \ldots; \Gamma_n)}{E \triangleright f(v_1, \ldots, v_n) : \mathbb{A}}\,(\mathsf{ax}) \qquad \frac{}{x : \mathbb{A} \triangleright x : \mathbb{A}}\,(\mathsf{hyp})$$

$$\frac{}{- \triangleright * : \mathbb{I}}\,(\mathbb{I}_i) \quad \frac{\Gamma \triangleright v : \mathbb{A} \otimes \mathbb{B} \quad \Delta, x : \mathbb{A}, y : \mathbb{B} \triangleright w : \mathbb{C} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright \mathsf{pm}\ v\ \mathsf{to}\ x \otimes y.w : \mathbb{D}}\,(\otimes_e) \quad \frac{\Gamma \triangleright v : \mathbb{A}}{\Gamma \triangleright \mathsf{dis}(v) : \mathbb{I}}\,(\mathsf{dis})$$

$$\frac{\Gamma \triangleright v : \mathbb{A} \quad \Delta \triangleright w : \mathbb{B} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright v \otimes w : \mathbb{A} \otimes \mathbb{B}}\,(\otimes_i) \qquad \frac{\Gamma \triangleright v : \mathbb{I} \quad \Delta \triangleright w : \mathbb{A} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright v\ \mathsf{to}\ *.w : \mathbb{A}}\,(\mathbb{I}_e)$$

$$\frac{\Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}}{\Gamma \triangleright \lambda x : \mathbb{A}.v : \mathbb{A} \multimap \mathbb{B}}\,(\multimap_i) \qquad \frac{\Gamma \triangleright v : \mathbb{A} \multimap \mathbb{B} \quad \Delta \triangleright w : \mathbb{A} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright vw : \mathbb{B}}\,(\multimap_e)$$

Figure 1: Term formation rules of affine lambda calculus.

The rule (ax) states that if there is a function $f \in \Sigma$ that has type $\mathbb{A}_1, \ldots, \mathbb{A}_n \to \mathbb{A}$ and a set of variables $v_1, \ldots, v_n$ whose types match the type of the arguments of $f$, then if that function is applied to $v_1, ..., v_n$ the respective result is of type $\mathbb{A}$. The rule (hyp) is a tautology: under the assumption that $x$ has type $\mathbb{A}$, $x$ has type $\mathbb{A}$. The rule ($\mathbb{I}_i$) asserts that the unit element $*$ always has type $\mathbb{I}$. The rule ($\multimap_i$) expresses that if $v$ is a term of type $\mathbb{B}$ with a variable $x$ of type $\mathbb{A}$, then $\lambda x : \mathbb{A}.v$ is a function of type $\mathbb{A} \multimap \mathbb{B}$. The rule $(\multimap_e)$ states that a function of type $\mathbb{A} \multimap \mathbb{B}$ can be applied to an argument of type $\mathbb{A}$ to produce a result of type $\mathbb{B}$. The rule $(\otimes_i)$ asserts that if there is a term $v$ of type $\mathbb{A}$ and a term $w$ of type $\mathbb{B}$, then the tensor of these terms is of type $\mathbb{A} \otimes \mathbb{B}$. The rule $(\otimes_e)$ expresses if there is a term $w$ of type $\mathbb{D}$ with variables $x$ and $y$ of types $\mathbb{A}$ and $\mathbb{B}$, respectively, and a term $v$ of type $\mathbb{A} \otimes \mathbb{B}$, then $v$ can be deconstructed into $x \otimes y$. The rule $(\mathbb{I}_e)$ states that if there is a term $w$ of type $\mathbb{A}$ and a term $v$ of type $\mathbb{I}$, then $v$ can be discarded, and only the term $w$ remains. Finally, the rule $(\mathsf{dis})$ asserts that a term $v$ of type $\mathbb{A}$ can be discarded, resulting in a term of type $\mathbb{I}$.

For a better understanding of the rules, a few straightforward programming examples are provided. For instance, the program that swaps the elements of a tensor product can be written as follows:

$$x : \mathbb{A}, y : \mathbb{B} \triangleright \mathsf{pm}\ x \otimes y\ \mathsf{to}\ a \otimes b.b \otimes a : \mathbb{B} \otimes \mathbb{A}$$

Now, to prove that this program is well-typed one can write the following typing derivation:

$$
\begin{array}{lll}
1 & x : \mathbb{A} \triangleright x : \mathbb{A} & (\textsf{hyp}) \\
2 & y : \mathbb{B} \triangleright y : \mathbb{B} & (\textsf{hyp}) \\
3 & x : \mathbb{A}, y : \mathbb{B} \triangleright x \otimes y : \mathbb{A} \otimes \mathbb{B} & (1, 2, \otimes_i) \\
4 & b : \mathbb{B} \triangleright b : \mathbb{B} & (\textsf{hyp}) \\
5 & a : \mathbb{A} \triangleright a : \mathbb{A} & (\textsf{hyp}) \\
6 & b : \mathbb{B}, a : \mathbb{A} \triangleright b \otimes a : \mathbb{B} \otimes \mathbb{A} & (4, 5, \otimes_i) \\
7 & x : \mathbb{A}, y : \mathbb{B} \triangleright \textsf{pm } x \otimes y \textsf{ to } a \otimes b . b \otimes a : \mathbb{B} \otimes \mathbb{A} & (3, 6, \otimes_e)
\end{array}
$$

Observe that in the notation of the third column, the numbers correspond to the premises utilized in the application of the rule.

Another example is the function that recieves a tensor product and returns first element and discards the second:

$$- \triangleright \lambda x : \mathbb{A} \otimes \mathbb{B} . \textsf{pm } x \textsf{ to } a \otimes b . \textsf{dis}(b) \ \textsf{ to } \ * . a : \mathbb{A}$$

To prove that this program is well-typed one can write the following typing derivation:

$$
\begin{array}{lll}
1 & b : \mathbb{B} \triangleright b : \mathbb{B} & (\textsf{hyp}) \\
2 & b : \mathbb{B} \triangleright \textsf{dis}(b) : \mathbb{I} & (1, \textsf{dis}) \\
3 & a : \mathbb{A} \triangleright a : \mathbb{A} & (\textsf{hyp}) \\
4 & a : \mathbb{A}, b : \mathbb{B} \triangleright \textsf{dis}(b) \ \textsf{ to } \ * . a & (2, 3, \mathbb{I}_e) \\
5 & x : \mathbb{A} \otimes \mathbb{B} \triangleright x : \mathbb{A} \otimes \mathbb{B} & (\textsf{hyp}) \\
6 & x : \mathbb{A} \otimes \mathbb{B} \triangleright \textsf{pm } x \textsf{ to } a \otimes b . \textsf{dis}(b) \ \textsf{ to } \ * . a : \mathbb{A} & (4, 5, \otimes I_e) \\
7 & - \triangleright \lambda x : \mathbb{A} \otimes \mathbb{B} . \textsf{pm } x \textsf{ to } a \otimes b . \textsf{dis}(b) \ \textsf{ to } \ * . a : \mathbb{A} & (6, \multimap_i)
\end{array}
$$

Also fala-se de Type inference algorithm? Tipo existe...

### 2.2.5 $\alpha$-equivalence

A natural notion of equivalence definition stems from the fact that terms that differ only in the names of their bound variables represent the same program. For instance, the functions $\lambda x : \mathbb{A}.x$ and $\lambda y : \mathbb{A}.y$ have the same input-output behavior, despite being represented by different lambda terms. This equivalence is called *$\alpha$-equivalence*.

**Definition 2.2.1.** The $\alpha$-equivalence is an equivalence relation on lambda terms that is used to rename bound variables. To rename a variable $x$ as $y$ in a term $v$, denoted by $v\{y/x\}$, is to replace all occurrences of $x$ in $v$ by $y$. Two terms $v$ and $w$ are $\alpha$-equivalent, written $=_\alpha$, if one can be derived from the other by a series of changes of bound variables

**Convention 2.2.1.** Terms are considered up to $\alpha$-equivalence from now on.

### 2.2.6 Substitution

The substitution of a variable $x$ for a term $w$ in a term $v$ is denoted by $v[w/x]$. It is only permitted to replace free variables. For instance, $\lambda x.x\,[v/x]$ is $\lambda x.x$ and not $\lambda x.v$ . Moreover, it is necessary to avoid the unintended binding of free variables. For example,

$$(\text{pm } x \otimes y \text{ to } a \otimes b.b \otimes a \otimes z)\,[z/\text{pm } c \otimes d \text{ to } e \otimes f.f \otimes e \otimes a]$$

is not the same as

$$\text{pm } x \otimes y \text{ to } a \otimes b.b \otimes a \otimes (\text{pm } c \otimes d \text{ to } e \otimes f.f \otimes e \otimes a).$$

Instead, the bounded variable $a$ must be renamed before the substitution, and in this case, the proper substitution is

$$(\text{pm } x \otimes y \text{ to } t \otimes b.b \otimes t \otimes z)\,[z/\text{pm } c \otimes d \text{ to } e \otimes f.f \otimes e \otimes a]$$

which is equal to

$$\text{pm } x \otimes y \text{ to } t \otimes b.b \otimes t \otimes (\text{pm } c \otimes d \text{ to } e \otimes f.f \otimes e \otimes a).$$

Note that a simple way of ensuring these restrictions are satisfied is not allowing the variable $x$ to occur in the context of $w$ in $v[w/x]$. Since $x$ is in the context of $v$, this is always the case in the affine lambda calculus.

**Definition 2.2.2.** Given the typings judgments $\Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}$ and $\Delta \triangleright w : \mathbb{A}$, the substitution

$\Gamma, \Delta \triangleright v[w/x] : \mathbb{B}$ is defined below. The types of judgments are omitted as no ambiguity arises.

$$\Gamma, \Delta \triangleright y[w/x] = \Gamma, \Delta \triangleright y,$$

$$\Delta \triangleright *[w/x] = \Delta \triangleright *,$$

$$\Gamma, \Delta \triangleright (\lambda y : \mathbb{B}.v)[w/x] = \Gamma, \Delta \triangleright \lambda y : \mathbb{B}.v[w/x],$$

$$(\mathsf{dis}(v))[w/x] = \mathsf{dis}(v[w/x]),$$

In the next three cases, $\Gamma, x : \mathbb{A} \in \mathsf{Sf}(\Gamma_1, \ldots, \Gamma_i, \ldots, \Gamma_n)$ and $\Gamma_i \triangleright v_i$

$$\Gamma, \Delta \triangleright (f(v_1, \ldots, v_n))[w/x] = \Gamma, \Delta \triangleright f(v_1[w/x], \ldots, v_n), \qquad (\text{ if } x : \mathbb{A} \in \Gamma_1)$$

$$\Gamma, \Delta \triangleright (f(v_1, \ldots, v_i, \ldots, v_n))[w/x] = \Gamma, \Delta \triangleright f(v_1, \ldots, v_i[w/x], \ldots, v_n), \qquad (\text{ if } x : \mathbb{A} \in T_i)$$

$$\Gamma, \Delta \triangleright (f(v_1, \ldots, v_n))[w/x] = \Gamma, \Delta \triangleright f(v_1, \ldots, v_n[w/x]), \qquad (\text{ if } x : \mathbb{A} \in \Gamma_n)$$

In the next two cases, $\Gamma, x : \mathbb{A} \in \mathsf{Sf}(\Gamma_1, \Gamma_2), \Gamma_1 \triangleright v$, and $\Gamma_2 \triangleright u$

$$\Gamma, \Delta \triangleright (vu)[w/x] = \Gamma, \Delta \triangleright (v[w/x]u), \qquad (\text{ if } x : \mathbb{A} \in \Gamma_1)$$

$$\Gamma, \Delta \triangleright (vu)[w/x] = \Gamma, \Delta \triangleright (vu[w/x]), \qquad (\text{ if } x : \mathbb{A} \in \Gamma_2)$$

In the next two cases, $\Gamma, x : \mathbb{A} \in \mathsf{Sf}(\Gamma_1, \Gamma_2), \Gamma_1 \triangleright v$, and $\Gamma_2 \triangleright u$

$$\Gamma, \Delta \triangleright (v \otimes u)[w/x] = \Gamma, \Delta \triangleright v[w/x] \otimes u, \qquad (\text{ if } x : \mathbb{A} \in \Gamma_1)$$

$$\Gamma, \Delta \triangleright (v \otimes u)[w/x] = \Gamma, \Delta \triangleright v \otimes u[w/x], \qquad (\text{ if } x : \mathbb{A} \in \Gamma_2)$$

In the next two cases, $\Gamma, x : \mathbb{A} \in \mathsf{Sf}(\Gamma_1, \Gamma_2), \Gamma_1 \triangleright v$, and $\Gamma_2, y : \mathbb{D}, z : \mathbb{E} \triangleright u$

$$\Gamma, \Delta \triangleright (\mathsf{pm}\ v\ \mathsf{to}\ y \otimes z.u)[w/x] = \Gamma, \Delta \triangleright \mathsf{pm}\ v[w/x]\ \mathsf{to}\ y \otimes z.u, \qquad (\text{ if } x : \mathbb{A} \in \Gamma_1)$$

$$\Gamma, \Delta \triangleright (\mathsf{pm}\ v\ \mathsf{to}\ y \otimes z.u)[w/x] = \Gamma, \Delta \triangleright \mathsf{pm}\ v\ \mathsf{to}\ y \otimes z.u[w/x], \qquad (\text{ if } x : \mathbb{A} \in \Gamma_2)$$

In the next two cases, $\Gamma, x : \mathbb{A} \in \mathsf{Sf}(\Gamma_1, \Gamma_2), \Gamma_1 \triangleright v$, and $\Gamma_2 \triangleright u$

$$\Gamma, \Delta \triangleright (v\ \mathsf{to}\ *.u)[w/x] = \Gamma, \Delta \triangleright v[w/x]\ \mathsf{to}\ *.u \qquad (\text{ if } x : \mathbb{A} \in \Gamma_1),$$

$$\Gamma, \Delta \triangleright (v\ \mathsf{to}\ *.u)[w/x] = \Gamma, \Delta \triangleright v\ \mathsf{to}\ *.u[w/x] \qquad (\text{ if } x : \mathbb{A} \in \Gamma_2).$$

The sequential substitutions $M[M_i/x_i] \ldots [M_n/x_n]$ are writen as $M[M_i/x_i, \ldots, M_n/x_n]$.

### 2.2.7 Properties

The calculus defined in Figure 1 possesses several desirable properties, which are listed below. Before proceeding, it is necessary to introduce some auxiliary notation. Given a context $\Gamma$, $te(\Gamma)$ denotes context $\Gamma$ with all types erased. The expression $\Gamma \simeq_\pi \Gamma'$ denotes that the contexts $\Gamma$ is a permutation of context $\Gamma'$.This notation also applies to non-repetitive lists of untyped variables $te(\Gamma)$. Additionally, a judgment $\Gamma \triangleright v : \mathbb{A}$ will often be abbreviated into $\Gamma \triangleright v$ or even just $v$ when no ambiguities arise.

The properties are as follows:

1. for all judgements $\Gamma \triangleright v$ and $\Gamma' \triangleright v$, $\mathrm{te}(\Gamma) \simeq_\pi \mathrm{te}(\Gamma')$;
2. additionally if $\Gamma \triangleright v : \mathbb{A}$, $\Gamma' \triangleright v : \mathbb{A}'$, and $\Gamma \simeq_\pi \Gamma'$, then $\mathbb{A}$ must be equal to $\mathbb{A}'$;
3. all judgements $\Gamma \triangleright v : \mathbb{A}$ have a unique derivation.
4. (exchange) For every judgement $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright v : \mathbb{D}$ it is possible to derive $\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright v : \mathbb{D}$.
5. (substitution) For all judgements $\Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}$ and $\Delta \triangleright w : \mathbb{A}$ it is possible to derive $\Gamma, \Delta \triangleright v[w/x] : \mathbb{B}$.

### 2.2.8 Equations-in-context

The simply typed lambda calculus is a formal language that captures operations like the application of a function to an argument and the elimination of variables. To express these operations there is a set of rules known as reduction rules. These rules fall into two primary categories: the $\beta$-*reductions*, which perform operations and enforce the implicit meaning of the term, and $\eta$-*reductions*, which simplify terms by exploiting the extensionality of functions. There is also a secondary class of reductions known as *commuting conversions*, which serve to disambiguate terms that, while equivalent, have different representations. As a result, affine $\lambda$-calculus comes equipped with the so-called equations-in-context $\Gamma \triangleright v = w : \mathbb{A}$, depicted in Figure 2.

| | | | |
|---|---|---|---|
| $(\beta)$ | $\Gamma, \Delta \triangleright (\lambda x : \mathbb{A}.\, v) w = v[w/x] : \mathbb{B}$ | $(\eta)$ | $\Gamma \triangleright \lambda x : \mathbb{A}.(v x) = v : \mathbb{A} \multimap \mathbb{B}$ |
| $(\beta_{\mathbb{I}_e})$ | $\Gamma \triangleright * \;\mathsf{to}\; *.\, v = v : \mathbb{A}$ | $(\eta_{\mathbb{I}_e})$ | $\Delta, \Gamma \triangleright v \;\mathsf{to}\; *.\, w[*/z] = w[v/z] : \mathbb{A}$ |
| $(\beta_{\otimes_e})$ | $E, \Gamma, \Delta \triangleright \mathsf{pm}\; v \otimes w \;\mathsf{to}\; x \otimes y.\, u = u[v/x, w/y] : \mathbb{A}$ | | |
| $(\eta_{\otimes_e})$ | $\Delta, \Gamma \triangleright \mathsf{pm}\; v \;\mathsf{to}\; x \otimes y.\, u[x \otimes y/z] = u[v/z] : \mathbb{A}$ | | |
| $(c_{\mathbb{I}_e})$ | $\Delta, \Gamma, E \triangleright u[v \;\mathsf{to}\; *.w/z] = v \;\mathsf{to}\; *.u[w/z] : \mathbb{A}$ | | |
| $(c_{\otimes_e})$ | $\Delta, \Gamma, E \triangleright u[\mathsf{pm}\; v \;\mathsf{to}\; x \otimes y.\, w/z] = \mathsf{pm}\; v \;\mathsf{to}\; x \otimes y.\, u[w/z] : \mathbb{A}$ | | |
| $(\eta_{\mathsf{dis}})$ | $x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n \triangleright v = \mathsf{dis}(x_1) \;\mathsf{to}\; * \ldots .\, \mathsf{dis}(x_{n-1}) \;\mathsf{to}\; *\; \mathsf{dis}(x_n) : \mathbb{I}$ | | |

Figure 2: Equations-in-context for affine lambda calculus

It is evident that, for example, equation $(\beta)$ enforces the meaning of $(\lambda x : \mathbb{A}.\ v)w$, which is interpreted as "$v$ with $w$ in place of $x$". The equation $(\eta)$, on the other hand, is a simplification rule that states that a function that applies another function $v$ to an argument $x$ can be simplified to the function $v$ itself. The remaining $\beta$ e $\eta$ equations follow similar reasoning. The commuting conversion $(c_{\mathbb{I}_e})$ expresses that substituting a variable $z$ by a term that maps a term $v$ to the unit element $*$ in a term $w$ is equivalent to mapping a term $v$ to the unit element $*$ and then replacing $z$ by $w$. The other commuting conversion has a similar interpretation.

## 2.3 Metric equational system

Mete-se contexto e tipo nas eqs métricas?

*Metric equations* Mardare et al. [2016], Mardare et al. [2017] are a strong candidate for reasoning about approximate program equivalence. These equations take the form of $t =_\epsilon s$, where $\epsilon$ is a non-negative rational representing the "maximum distance" between the two terms $t$ and $s$. The metric equational system for linear lambda calculus is depicted in Figure 3.

$$\frac{}{v =_0 v}\ \text{(refl)} \qquad \frac{v =_q w \quad w =_r u}{v =_{q+r} u}\ \text{(trans)} \qquad \frac{v =_q w \quad r \geq q}{v =_r w}\ \text{(weak)}$$

$$\frac{\forall r > q.\ v =_r w}{v =_q w}\ \text{(arch)} \qquad \frac{\forall i \leq n.\ v =_{q_i} w}{v =_{\wedge q_i} w}\ \text{(join)} \qquad \frac{v =_q w}{w =_q v}\ \text{(sym)}$$

$$\frac{v =_q w \quad v' =_r w'}{v \otimes v' =_{q+r} w \otimes w'} \qquad \frac{\forall i \leq n.\ v_i =_{q_i} w_i}{f(v_1, ..., v_n) =_{\Sigma q_i} f(w_1, ..., , w_n)} \qquad \frac{v =_q w}{\lambda x : \mathbb{A}.\ v =_q \lambda x : \mathbb{A}.\ w}$$

$$\frac{v =_q w \quad v' =_r w'}{\text{pm } v \text{ to } x \otimes y.\ v' =_{q+r} \text{pm } w \text{ to } x \otimes y.\ w'} \qquad \frac{v =_q w}{\text{dis}(v) =_q \text{dis}(w)} \qquad \frac{v =_q w \quad v' =_r w'}{vv' =_{q+r} ww'}$$

$$\frac{\Gamma \triangleright v =_q w : \mathbb{A} \quad \Delta \in \text{perm}(\Gamma)}{\Delta \triangleright v =_q w : \mathbb{A}} \qquad \frac{v =_q w \quad v' =_r w'}{v \text{ to } *.v' =_{q+r} w \text{ to } *.w'} \qquad \frac{v =_q w \quad v' =_r w'}{v[v'/x] =_{q+r} w[w'/x]}$$

Figure 3: Metric equational system

Here, perm$(\Gamma)$ denotes the set of possible permutions of context $\Gamma$. The rules (refl), (trans), and (sym) generalize the properties of reflexivity, transitivity, and symmetry of equality.

Rule (weak) asserts that if two terms are at a maximum distance $q$ from each other, then they are also separated by any $r \geq q$. Rule (arch) states that if $v =_r w$ for all approximations $r$ of $q$, then it necessarily follows that $v =_q w$. The rule (join) expresses that if several maximum distances between two terms are known, the actual maximum distance between them is the minimum of these distances. The rule that follows conveys that if the maximum distance between two terms $v$ and $w$ is $q$, and the maximum distance between terms $v'$ and $w'$ is $r$, then the maximum distance between the tensor products $v \otimes v'$ and $w \otimes w'$ is $q + r$. The remaining rules follow similar reasoning.

To ilustrate the usefulness of these equations, consider the program $P$ that recieves a tensor product, swaps its elements and then applies a function $f$ to to the new second element of the tensor pair:

$$P = x : \mathbb{A}, y : \mathbb{B} \triangleright \text{pm } x \otimes y \text{ to } a \otimes b.b \otimes f(a) : \mathbb{D} \otimes \mathbb{A}$$

Now, consider the case where $f$ is an idealized version of function $f^\epsilon$ mapping $a$ to $f(a)^\epsilon$. The program that applies the "real" function $f$ to the first element of the tensor pair is $P^\epsilon$:

$$P^\epsilon = x : \mathbb{A}, y : \mathbb{B} \triangleright \text{pm } x \otimes y \text{ to } a \otimes b.b \otimes f(a)^\epsilon : \mathbb{D} \otimes \mathbb{A}$$

Knowing that $f(a)^\epsilon =_\epsilon f(a)$, it is possible to show that $P^\epsilon =_\epsilon P$ using the metric equational system. The prove is as follows. The types and contexts are omitted for brevity as no ambiguity arises.

| | | |
|---|---|---|
| 1 | $f(a)^\epsilon =_\epsilon f(a)$ | |
| 2 | $b =_0 b$ | (refl) |
| 3 | $b \otimes f(a)^\epsilon =_\epsilon b \otimes f(a)$ | $(1, 2, \otimes_i)$ |
| 4 | $x \otimes y =_0 x \otimes y$ | (refl) |
| 5 | $\text{pm } x \otimes y \text{ to } a \otimes b.b \otimes f(a)^\epsilon =_\epsilon \text{pm } x \otimes y \text{ to } a \otimes b.b \otimes f(a)$ | $(3, 4, \otimes_e)$ |

## Chapter 3

# Quantum Meets Lambda Calculus

Quantum lambda calculus integrates quantum computation with higher-order functions, thereby emerging as a powerful tool for formal reasoning about quantum programs within a functional programming framework. This functional paradigm, with a static type system, offers the significant advantage of ensuring the absence of run-time errors, *i.e.*, potential errors can be detected at compile-time, when the program is written, rather than during execution.
The principal distinction between the quantum lambda calculus introduced in this section and the formulation proposed by Selinger Selinger and Valiron [2006]; Selinger et al. [2009] lies in the handling of data duplication. In this aproach, as dictated by the type system in Figure 1, duplication of any data is strictly prohibited. In contrast, Selinger's approach permits the duplication of classical data while strictly forbidding the duplication of quantum data. Nonetheless, the controlled duplication of both classical and quantum data will be adressesd in chapter 5.

Escrever intro ao capítulo: o que é que se vai abordar e tal e é baseada nos livros x e y

The first two sections of this chapter present the mathematical and quantum computing preliminaries necessary for understanding theory of quantum computation. The introduction to quantum computing is primarily based on Nielsen and Chuang [2010]; Watrous [2018], while the mathematical foundations are also based on Rudin [1991] and Guide [2006] .

## 3.1 Mathematical Preliminaries

It is impossible to present the theory of quantum computation without introducing some concepts of linear algebra within finite-dimensional spaces. This section provides a brief overview of the aspects of linear algebra that are most pertinent to the study of quantum computation.

### 3.1.1 Complex vector spaces

The basic objects of linear algebra are vector spaces. The vector spaces of interest in this work are the complex vector spaces, $\mathbb{C}^n$ and $\mathbb{C}^{n\times m}$, therefore, the following definition is given in the context of complex vector spaces.

**Definition 3.1.1.** A *vector space* (over $\mathbb{C}$) consists of a set $V$ whose elements are called vectors, together with two operations:

- An operation called vector addition that takes two vectors $v, w \in V$, and results in a third vector, written $v + w \in V$;
- An operation called scalar multiplication that takes a scalar $a \in \mathbb{C}$ and a vector $v \in V$, and results in a new vector, written $r \cdot v \in V$

which satisfy the following conditions (called axioms), for all $u, v, w \in V$ and $a_1, a_2 \in \mathbb{C}$:

1. Vector addition is commutative: $u + v = v + u$;
2. Vector addition is associative: $(u + v) + w = u + (v + w)$;
3. There is a zero vector $0 \in V$ such that $v + 0 = v$ for all $v \in V$;
4. Each $v \in V$ has an additive inverse $w \in V$ such that $w + v = 0$;
5. Scalar multiplication distributes over scalar addition, $(a_1 + a_2) \cdot v = a_1 \cdot v + a_2 \cdot v$;
6. Scalar multiplication distributes over vector addition, $a_1 \cdot (v + w) = a_1 \cdot v + a_1 \cdot w$;
7. Ordinary multiplication of scalars associates with scalar multiplication, $(a_1 a_2) \cdot v = a_1 \cdot (a_2 \cdot v)$;
8. Multiplication by the scalar 1 is the identity operation, $1 \cdot v = v$.

The letters $R, V, W$ will often be used to refer to vector spaces. All vector spaces are assumed to be finite dimensional, unless otherwise noted.

In the case of the vector space $\mathbb{C}^n$, the space of all n-tuples of complex numbers, $(a_1, \ldots, a_n)$, addition and scalar multiplication are defined in the following standard way:

- Addition: for vectors $u = (a_1, \ldots, a_n), v = (b_1, \ldots, b_n) \in \mathbb{C}^n$, the vector $u + v \in \mathbb{C}^n$ is defined by the equation $(u + v) = (a_1 + b_1, \ldots, a_n + b_n)$.

- Scalar multiplication: for a scalar $a \in \mathbb{C}$ and a vector $v = (b_1, \ldots, b_n) \in \mathbb{C}^n$, the vector $a \cdot v \in \mathbb{C}^n$ is defined by the equation $a \cdot v = (a \cdot b_1, \ldots, a \cdot b_n)$.

The zero vector in $\mathbb{C}^n$ is the vector with all entries equal to zero.
Sometimes, the column matrix notation is used to represent vectors in $\mathbb{C}^n$, that is, a vector $v \in \mathbb{C}^n$ is written as a column matrix

$$v = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}.$$

Other times for readability the format $(a_1, \ldots, a_n)$ is used. The latter should be interpreted as a shorthand for a column vector.
$\mathbb{C}^{n \times m}$ is a vector space when addition is defined as matrix addition and scalar multiplication is defined as multiplication of each component by the scalar. The zero vector is the zero matrix, *i.e.*, the matrix with all entries equal to zero.

**Definition 3.1.2.** A *vector subspace* of a vector space $V$ is a subset $W$ of $V$ such that $W$ is also a vector space, that is, $W$ must be closed under scalar multiplication and addition.

**Definition 3.1.3.** A *spanning set* of a vector space is a set of vectors $v_1, \ldots, v_n$ such that any vector $v$ in the vector space can be written as a linear combination $v = \sum_i a_i v_i$ of vectors in that set, where $a_i$ are scalars.

**Definition 3.1.4.** A set of non-zero vectors $v_1, \ldots, v_n$ are *linearly dependent* if there exists a set of complex numbers $a_1, \ldots, a_n$ with $a_i \neq 0$ for at least one value of $i$, such that

$$a_1 v_1 + a_2 v_2 + \ldots a_n v_n = 0.$$

**Definition 3.1.5.** A set of vectors $v_1, \ldots, v_n$ are *linearly independent* if they are not linearly dependent.

**Definition 3.1.6.** A *basis* for a vector space is a sequence of vectors that is linearly independent and that spans the space.

**Definition 3.1.7.** The number of elements in the basis is defined to be the *dimension* of $V$.

### 3.1.2 Linear operators

A *linear operator* between vector spaces $V$ and $W$ is defined to be any function $A : V \to W$ which is linear in its inputs, *i.e.*

$$A\left(\sum_i a_i v_i\right) = \sum_i a_i A(v_i)$$

Usually $A(v)$ is just denoted $Av$.

Suppose $V, W,$ and $R$ are vector spaces, and $A : V \to R$ and $B : W \to R$ are linear operators. Then the notation $BA$ is used to denote the *composition* of $B$ with $A$, defined by $(BA)(v) \equiv B(A(v))$. Once again, $(BA)(v).$ is abbreviated as $BAv$.

For any choice of complex spaces $V \in \mathbb{C}^n$ and $W \in \mathbb{C}^m$, there is a bijective linear correspondence between the set of operators from $V$ to $W$ and the set of $n \times m$ matrices. The claim that the matrix $A \in \mathbb{C}^{m\times n}$ is a linear operator just means that

$$A\left(\sum_i a_i v_i\right) = \sum_i a_i A(v_i)$$

is true as an equation where the operation is matrix multiplication of $A$ by a collumn vector in $\mathbb{C}^n$. Clearly, this is true! On the other hand, suppose $A : V \to W$ is a linear operator between vector spaces $V$ and $W$, such that $V \in \mathbb{C}^n$ and $W \in \mathbb{C}^m$. Suppose $v_1, \ldots, v_n$ is a basis for $V$ and $w_1, \ldots, w_n$ is a basis for $W$. Then for each $j$ in the range $1, \ldots, m$, there exist complex numbers $A_{1j}$ through $A_{nj}$ such that

$$Av_j = \sum_i A_{ij} w_i.$$

The matrix whose entries are the values $A_{ij}$ is said to form a *matrix representation* of the operator $A$. This matrix representation of $A$ is completely equivalent to the operator $A$. As a result, when considering operators on vector spaces of the form $\mathbb{C}^n$ it is common to refer to the operator $A$ and its matrix representation interchangeably.

### 3.1.3 Inner product

**Definition 3.1.8.** The *inner product* $\langle\cdot,\cdot\rangle$ is a function from a vector space $V$ to the complex numbers, $\langle\cdot,\cdot\rangle : V \times V \to \mathbb{C}$, that satisfies the following properties for all $v, w \in V$ and $r \in \mathbb{C}$:

1. Linearity in the second argument,

$$\left\langle v, \sum_i a_i w_i \right\rangle = \sum_i a_i \langle v, w_i \rangle.$$

2. $\langle v, w \rangle = \langle w, v \rangle^\dagger$, where $(-)^*$ is the complex conjugate operation.

3. $\langle v, w \rangle \geq 0$ with equality if and only if $v = 0$.

The inner product $\langle v, w \rangle$ of two vectors $v = (a_1, \ldots, a_n), w = (b_1, \ldots, b_n) \in \mathbb{C}^n$ is defined as

$$\langle v, w \rangle = \sum_i a_i^* b_i.$$

#### Trace

In order to define inner product of a matrix, it is necessary to first define the trace of a matrix. The trace of a square matrix $A \in \mathbb{C}^{n \times n}$ defined to be the sum of its diagonal elements,

$$\text{tr}(A) = \sum_i A_{ii}.$$

The trace is *cyclic*, that is, $\text{tr}(AB) = \text{tr}(BA)$, and *linear*, $\text{tr}(A+B) = \text{tr}(A)+\text{tr}(B)$, $\text{tr}(a \cdot A) = a \cdot \text{tr}(A)$, where matrices $A, B \in \mathbb{C}^{n \times n}$, and $a$ is a complex number.
By means of the trace, one defines the inner product of two operators $A, B \in \mathbb{C}^{m \times n}$ as follows

$$\langle A, B \rangle = \text{tr}(A^\dagger B).$$

In the *finite* dimensional complex vector spaces that come up in quantum computation and quantum information, a *Hilbert space* is exactly the same thing as an inner product space. As a result, both $\mathbb{C}^n$ and $\mathbb{C}^{n \times m}$ are Hilbert spaces.

### 3.1.4 Norm and normed spaces

**Definition 3.1.9.** A *norm* $\|\cdot\|$ is a function that associates an element of a vector space $V$ with a non-negative real number, such that the following properties hold:

1. Positive definiteness: $\|v\| \geq 0$ for all $v \in V$, with $\|v\| = 0$ if and only if $v = 0$;

2. Positive scalability: $\|av\| = |a|\|v\|$ for all $v \in V$ and $a$ is a scalar;

3. The triangle inequality: $\|v + w\| \leq \|v\| + \|w\|$ for all $v, w \in V$.

**Definition 3.1.10.** A vector space together with a norm is called a *normed vector space*.

Every normed space may be regarded as a metric space, in which the distance $d(x, y)$ between vectors $x$ and $y$ is $\|x - y\|$ . The relevant properties of $d$ are

1. $0 < d(x, y) < \infty$ for all $x$ and $y$,
2. $d(x, y) = 0$ if and only if $x = y$,
3. $d(x, y) = d(y, x)$ for all $x$ and $y$,
4. $d(x, z) < d(x, y) + d(y, z)$ for all $x, y, z$.

Every inner product space is a normed space, where the norm of a vector $v$ is defined as $\|v\| = \sqrt{\langle v, v\rangle}$.

**Definition 3.1.11.** Two vector $u, v$ are said to be *orthogonal* if $\langle v, u\rangle$. An *orthogonal set* is a set of orthogonal vectors of the same vector space.

**Definition 3.1.12.** A *unit vector* is a vector $v$ such that $\|v\| = 1$. It is also said that $v$ is *normalized* if $\|v\| = 1$

**Definition 3.1.13.** An orthogonal set of unit vectors is called an *orthonormal set*, and when such a set forms a basis it is called an *orthonormal basis*.

### 3.1.5 Important classes of operators/matrices

Linear operators mapping a complex space $\mathbb{C}^n$ or $\mathbb{C}^{n\times n}$ to itself will be called *square operators* due to the fact that their matrix representations are square matrices. Therefore, those definitions given in the context of square operators are also valid for square matrices.
The following classes of operators are of particular interest in quantum information theory.

**Definition 3.1.14.** *Normal operators.* A square operator $A$ is *normal* if $AA^\dagger = A^\dagger A$, where $(-)^\dagger$ is the adjoint operation.

**Definition 3.1.15.** *Hermitian operators.* A square operator $A$ is *hermitian* if $A = A^\dagger$. Every Hermitian operator is a normal operator.

**Definition 3.1.16.** *Positive (semidefinite) operators*. A square operator $A$ is *positive*, denoted $A \geq 0$, if $\langle v, Av\rangle \geq 0$ for all $v \in \mathbb{C}^n$.

**Definition 3.1.17.** *Unitary operators.* A square operator $U$ is *unitary* if $U^\dagger U = UU^\dagger = I$, where $I$ is the identity operator. The letter $U$ will often be used to refer to unitary operators.

Geometrically, unitary operators are important because they preserve inner products between vectors, $\langle Uv, Uw\rangle = \langle v, w\rangle$ for any two vectors $v$ and $w$. Let $S_1 = \{v_i\}$ be any orthonormal basis set. Define $s_2 = \{w_i\} = \{Uv_i | v_i \in S\}$, so $S_2$ is also an orthonormal basis set, since unitary operators preserve inner products. Note that $U = \sum_i w_i v_i^\dagger$. Conversely, if $\{v_i\}$ are any two orthonormal bases, then it is easily checked that the operator $U$ defined by $U = \sum_i w_i^\dagger v_i$ is unitary.

**Definition 3.1.18.** *Density operator.* Positive semidefinite operators that have trace equal to 1 are called *density operators*. Lowercase Greek letters, such as $\rho, \sigma$ are conventionally used to denote density operators.

**Definition 3.1.19.** *Isometries.* An operator $A : \mathbb{C}^n \to \mathbb{C}^m$ is as isometry if $\|Av\| = \|v\|$ for all elements all elements $v \in \mathbb{C}^n$.

**Definition 3.1.20.** *Diagonal operators.* A square operator $A$ is *diagonal* if $A_{ij} = 0$ for all $i \neq j$.

### 3.1.6 Useful norms

**Definition 3.1.21.** The *euclidean norm*, $\|.\|_2$, of a vector $v \in \mathbb{C}^n$ is defined as:

$$\|v\|_2 = \sqrt{\langle v, v\rangle}.$$

**Definition 3.1.22.** The *trace norm*, $\|.\|_2$, is defined as:

$$\|A\|_1 = \text{tr}\sqrt{A^\dagger A}$$

This norm is also known as the Schatten 1-norm. The trace norm induces a metric on the set of density matrices which is defined by $d(\rho, \sigma) = \|\rho - \sigma\|_1$.

**Definition 3.1.23.** Given vector spaces $V$ and $W$, the operator, $\|.\|_\sigma$, norm for an operator $E : V \to W$ is defined as

$$\|E\|_\sigma = \text{sup}\{\|E(v)\| \mid \|v\| = 1.\}$$

### 3.1.7 Eigenvectors, eigenvalues and the spectral theorem

**Definition 3.1.24.** An *n-permutation* is a function on the first $n$ positive integers $\pi = \{1, \ldots, n\} \to \{1, \ldots, n\}$ that is one-to-one and onto. In a permutation each number $1, \ldots, n$ appears as output for one and only one input.

**Definition 3.1.25.** The *sign* of a permutation sgn$(\pi)$ is $-1$ if the number of inversions in $\pi$ is odd and is $+1$ if the number of inversions is even.

**Definition 3.1.26.** The *determinant* of a square matrix $A \in \mathbb{C}^{n\times n}$ is defined as

$$\det(A) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{i=1}^{n} A_{i\sigma(i)},$$

Here $S_n$ is the set of all $n$-permutations $\pi = \{1, \ldots, n\} \to \{1, \ldots, n\}$, and sgn$(\pi)$ denotes the sign of the permutation $\pi$.

**Definition 3.1.27.** An *eigenvector* of a linear operator $A$ on a vector space is a non-zero vector $v$ such that $Av = \lambda v$, where $\lambda$ is a complex number known as the *eigenvalue* of $A$ corresponding to $v$.

The *characteristic polynomial* of a square operator $A$ is the polynomial $p(\lambda) = \det(A - \lambda I)$, where $I$ is the identity operator. It can be shown that the characteristic function depends only upon the operator A, and not on the specific matrix representation used for $A$. By the fundamental theorem of algebra, every polynomial has at least one complex root, so every operator $A$ has at least one eigenvalue, and a corresponding eigenvector. The solutions of the *characteristic equation* $c(\lambda) = 0$ are the eigenvalues of the operator $A$.

**Theorem 3.1.1.** Nielsen and Chuang *[2010] Every normal operator $A \in \mathbb{C}^{n\times n}$ can be expressed as a linear combination $\sum_i \lambda_i b_i b_i^\dagger$ where the set $\{b_i, \ldots, bn\}$ is an orthonormal basis on $\mathbb{C}^n$.*

Using this last result any function $f : \mathbb{C} \to \mathbb{C}$, can be extended to normal matrices via,

$$f(A) = \sum_i f(\lambda_i) b_i b_i^\dagger \tag{3.1}$$

where $A = \sum_i \lambda_i b_i b_i^\dagger$ is the spectral decomposition of $A$.

### 3.1.8 Tensor Products and Direct Sums of Complex spaces

**Definition 3.1.28.** Consider two finite complex spaces spaces $V$ and $W$ with respective bases $(e_1, \ldots, e_n)$ and $(f_1, \ldots, f_k)$. The tensor product of $V$ and $W$, denoted $V \otimes W$, is defined as the space generated by the basis of syntactic symbols:

$$(e_1 \otimes f_1, \ldots, e_1 \otimes f_k, \ldots, e_n \otimes f_1, \ldots, e_n \otimes f_k).$$

The tensor product of two elements $v = \sum_i \lambda_i \cdot e_i$ and w = $\sum_j \mu_j \cdot f_j$ is:

$$v \otimes w = \sum_{i,j} \lambda_i \mu_j \cdot e_i \otimes f_j.$$

The inner product in $V \otimes W$ is defined as follows

$$\langle e_i \otimes f_j, e_k \otimes f_l \rangle = \langle e_i, e_k \rangle \langle f_j, f_l \rangle.$$

The tensor product of two operators $A \in \mathbb{C}^{n \times m}$ and $B \in \mathbb{C}^{p \times q}$ is defined as the operator $A \otimes B \in \mathbb{C}^{np \times mq}$ such that

$$(A \otimes B)(v \otimes w) = Av \otimes Bw.$$

The notation $(-)^{\otimes n}$ will be used to denote the tensor product of a space, vector, or operator with itself $n$ times.

**Definition 3.1.29.** The *direct sum* of two vector spaces $V$ and $W$, denoted $V \oplus W$, is the space of all pairs $(v, w)$ where $v \in V$ and $w \in W$.

The inner product in $V \oplus W$ is defined as follows

$$\langle (v_1, w_1), (v_2, w_2) \rangle = \langle v_1, v_2 \rangle + \langle w_1, w_2 \rangle.$$

## 3.2 Quantum Computing Preliminaries

The basic unit of information in quantum computation is a quantum bit or qubit Perdrix [2008]. Just as classical bit, which can be in one of two states, 0 or 1, a qubit also has a state. Qubits are represented using *Dirac notation*, where the ket symbol $|\psi\rangle$ is used to denote a quantum state $\psi$. The corresponding bra symbol $\langle\psi|$ is used to denote the conjugate transpose of the state $\psi$. In this setting, the inner product of two states $|\psi\rangle$ and $|\phi\rangle$ is denoted $\langle\psi|\phi\rangle$.

### 3.2.1 The 2-Dimensional Hilbert Space

The state of a single qubit is described by a normalized vector of the 2-dimensional Hilbert space $\mathbb{C}^2$. States $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ are equivalent to the classical states 0 and 1, respectively. These states, known as the *computational basis* states, form an orthonormal basis for this vector space. Unlike classical bits, a qubit is not restricted to the states $|0\rangle$ and $|1\rangle$; it can exist in a linear combination of these states, commonly referred to as a *superposition*. Consequently, a general qubit state can be written as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{3.2}$$

$\alpha$ and $\beta$ are known as *amplitudes*. $|\alpha|^2$ and $|\beta|^2$ can be seen as the probabilities of measuring each state. Because $|\alpha|^2 + |\beta|^2 = 1$, Equation 3.2 and be rewritten as

$$|\psi\rangle = e^{i\gamma}\left(\cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\right), \tag{3.3}$$

where $\theta$, $\phi$ and $\gamma$ are real numbers. $e^{i\gamma}$ is known as a *global phase* and is often ignored because it has no observable effects, *i.e.*, it does not affect the probabilities of measurement outcomes. When disregarding the global phase, the quantum state can $|\psi\rangle$ be represented as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{3.4}$$

which corresponds to a point in the unit sphere where $\theta$ marks the latitude (*i.e.* the polar angle) and $\phi$ marks the longitude (*i.e.* the azimuthal angle). This representation is traditionally called the Bloch sphere representation. A point in the latter representation corresponds to the vector in $\mathbb{R}^3$ defined by $(\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$ and often called Bloch vector.

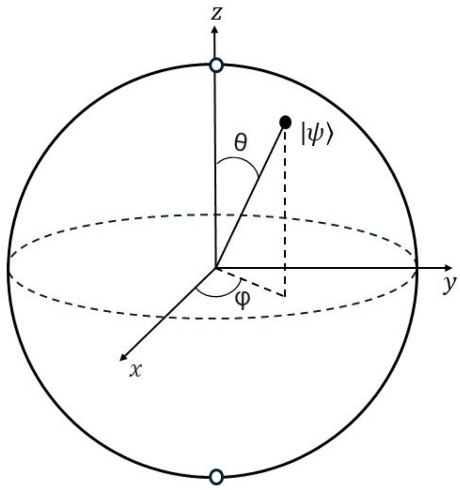


Figure 4: Bloch sphere representation of a qubit

The distance between two quantum states $|\psi\rangle$ and $|\psi'\rangle$ is their Euclidean distance in the Bloch sphere Wallman and Emerson [2016]; Nielsen and Chuang [2010].
There are infinite points in the Bloch sphere, which might suggest the possibility of encoding an infinite amount of information in the infinite binary expansion of the angle $\theta$. However, when a qubit is measured, it collapses to one of the basis states, so only one bit of information can be extracted from a qubit. To accurately determine the amplitudes $\alpha$ and $\beta$, an infinite number of identical qubit copies would need to be measured. Nevertheless, it is still conceptually valid to think of these amplitudes as "hidden information". One could say that quantum computation is the art of manipulating this hidden information using phenomena such as interference and superposition to perform tasks that would be impossible or inefficient with classical computers.

### 3.2.2 Multi-qubit States

The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. As a result, an $n$-qubit state can be represented by a unit vector in $2^n$-dimensional Hilbert space $\mathbb{C}^{2^n}$. The notations $|\psi\rangle \otimes |\phi\rangle$, $|\psi\rangle\,|\phi\rangle$, and $|\psi\phi\rangle$ are used to denote the tensor product of two states $|\psi\rangle$ and $|\phi\rangle$. The computational basis states of an $n$-qubit system are of the form $|x_1 \ldots x_n\rangle$ and so a quantum state of such a system is specified by $2^n$ amplitudes. For instance, a two-qubit state can be written as

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle\,.$$

It should be noted that unfortunately, no simple generalization of the Bloch sphere known for multiple qubits.

### Entanglement

An $n$-qubit mixed state can be represented by a density operator $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$, whose matrix representation is $\rho = \Sigma_i\, p_i |\phi_i\rangle\langle\phi_i|$. A density operator encodes uncertainty about the current state of the quantum system at hand. For example, a mixed state with half probability of $|0\rangle$ and $|1\rangle$ can be represented by $\frac{|0\rangle\langle 0|+|1\rangle\langle 1|}{2} = I/2$, where $I$ is the identity matrix. One usually denotes density matrices by the greek letters $\rho$, $\sigma$, and so forth. The set of density operators is denoted by $\mathcal{D}_n \subseteq \mathbb{C}^{2^{n\times n}}$.

Measurements extract classical information from quantum states. If a measurement $M_m$ is performed on a state $\rho$, the outcome $m$ is observed with probability $p_m = \text{Tr}(M_m \rho M_m^\dagger)$ for each $m$. Moreover, after a measurement yielding outcome $m$, the state collapses to $M_m \rho M_m^\dagger / p_m$.

Operations on quantum systems can be described using unitary operators. An operator, $U$, is unitary if its Hermitian conjugate is its own inverse, *i.e.*, $U^\dagger U = UU^\dagger = I$. For a pure state $|\psi\rangle$, a unitary operator $U$ describes an evolution from $|\psi\rangle$ to $U|\psi\rangle$. Similarly, for a density operator $\rho$, the corresponding evolution is $\rho \mapsto U\rho U^\dagger$. For example, the bit flip gate $X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ maps $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$. On the other hand, the Hadamard gate $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ maps $|0\rangle$ to $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ (denoted as $|+\rangle$) and $|1\rangle$ to $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ (denoted as $|-\rangle$). There are also multi-qubit gates, such as $CNOT$, which leaves the states $|00\rangle$ and $|01\rangle$ unchanged, and maps $|10\rangle$ and $|11\rangle$ to each other.

More broadly, the evolution of a quantum system can be defined by a super-operator $E$, which is a completely-positive and trace-preserving linear map from $\mathcal{D}(n)$ to $\mathcal{D}(m)$. A super-operator $E$ is called positive if it sends positive matrices to positive matrices, *i.e.* $A \geq 0 \Rightarrow EA \geq 0$. A super-operator is said to be completely positive if, for any positive integer $k$ and any $k$-dimensional Hilbert space $\mathbb{C}^{2^k}$, the super-operator $E \otimes I_{\mathbb{C}^{2^k}}$ is a positive map on $\mathcal{D}(n \times k)$. Finally, a super-operator $E$ is called trace-preserving if $\text{Tr}\, EA = \text{Tr} A$ Watrous [2018]. Completely-positive, trace-preserving super-operators are traditionally called quantum channels.

For every super-operator $E : \mathcal{D}(n) \to \mathcal{D}(m)$, there exists a set of Kraus operators $\{\epsilon_k\}_k$ such that $E(\rho) = \Sigma_k\, \epsilon_k\, \rho\, \epsilon_k^\dagger$ for any input $\rho \in \mathcal{D}(n)$. Note that the set of Kraus operators is finite if the Hilbert space is finite-dimensional. The Kraus form of $E$ is written as $E = \Sigma_k\, \epsilon_k \circ \epsilon_k^\dagger$.

A matrix $A \in \mathbb{C}^{n\times n}$ is Hermitian if $A = A^\dagger$. A matrix $A \in \mathbb{C}^{n\times n}$ is said to be normal if $AA^\dagger = A^\dagger A$. Clearly every Hermitian matrix is normal. Note also that for every matrix $A \in C^{n\times n}$ the matrix $A^\dagger A$ is Hermitian. Next, it is well-known that by appealing to the spectral theorem [NC16], every normal matrix $A \in \mathbb{C}^{n\times n}$ can be expressed as a linear combination $\sum_i \lambda_i b_i b_i^\dagger$ where the set $\{b_i, \ldots, bn\}$ is an orthonormal basis of $\mathbb{C}^n$. Using this last result we can extend any function $f : \mathbb{C} \to \mathbb{C}$, to normal matrices via,

$$f(A) = \sum_i f(\lambda_i) b_i b_i^\dagger \tag{3.5}$$

. . . The Bloch vector is given by

$$r_\mu = \mathrm{Tr}(\rho\sigma_\mu) \tag{3.6}$$

add trace, partial trace, reduced density matrix, and respective Bloch Vector, put the last paragragh in the right place and rewrite it

In the quantum paradigm, a potential notion of approximate equivalence arises from the so-called diamond norm Watrous [2018], which induces a metric (roughly, a distance function) on the space of quantum programs (seen semantically as completely positive trace-preserving super-operators). This norm relies on another norm known as the trace norm. The $\|\|_1$ latter is defined by $\|A\|_1 = \mathrm{Tr}\sqrt{A^\dagger A}$ for matrices $A \in \mathbb{C}^{n\times n}$. The trace norm induces a metric on the set of density matrices which is defined by $d(\rho, \sigma) = \|\rho - \sigma\|_1$. The trace distance between two super-operators $E, E' : \mathbb{C}^{n\times n} \to \mathbb{C}^{m\times m}$, denoted as $T(E, E')$, is defined as follows:

$$T(E, E') = \max\{\|(E - E')A\|_1 \mid \|A\|_1 = 1\} \tag{3.7}$$

Unfortunately, this norm is not stable under tensoring Watrous [2018], and consequently, the diamond norm, which is based on the trace norm, is used instead. The diamond norm between two super-operators $E, E' : \mathbb{C}^{n\times n} \to \mathbb{C}^{m\times m}$ is defined as:

$$\|E - E'\|_\diamond = T(E \otimes I_n, E' \otimes I_n) \tag{3.8}$$

where $I_n$ is the identity super-operator over the space $\mathbb{C}^{n\times n}$.

Consider an operator r : $(\mathbb{C}^n \to \mathbb{C}^m) \to (\mathbb{C}^{n\times n} \to \mathbb{C}^{m\times m})$ that sends an operator $T$ to the mapping $A \mapsto TAT^\dagger$. The exact calculation of distances induced by $\|\|_\diamond$ tends to be quite complicated, but a useful property for calculating the distance between quantum channels in the image of $r$ is provided Watrous [2018]: Consider two operators $T, S : n \to m$. There exists a unit vector $v \in \mathbb{C}^n$ such that,

$$\|r(T)(vv^\dagger) - r(S)(vv^\dagger)\|_1 = \|r(T) - r(S)\|_\diamond \tag{3.9}$$

The no-cloning theorem states that it is impossible to duplicate a quantum bit Wootters and Zurek [1982]. This principle is upheld by the type system outlined in Figure 1, which does not allow the repeated use of a variable (seen as a quantum resource).

## 3.3 Quantum Lambda Calculus:Interpretation

In order to define the interpretation of judgments $\Gamma \triangleright v : \mathbb{A}$, it is necessary to establish some notation first. Considering $v \in V, w \in W$, and $u \in U$ where $V, W, U$ represent vector spaces, $\mathsf{sw}_{V,W} : V \otimes W \to W \otimes V$, denotes the swap operator, defined as $\mathsf{sw}_{V,W} = v \otimes w \mapsto w \otimes v$; $\rho_V : \mathbb{C} \otimes V \to V$ is the left unitor defined as $\rho_V = 1 \otimes v \mapsto v$; $\lambda_V : V \otimes \mathbb{C} \to V$ is the right unitor defined as $\lambda_V = v \otimes 1 \mapsto v$; $\alpha_{V,W,U} : V \otimes (W \otimes U) \to (V \otimes W) \otimes U$ is the left associator, defined as $\alpha_{V,W,U} = v \otimes (w \otimes u) \mapsto (v \otimes w) \otimes u$; and $!_V : V \to \mathbb{C}$ is the trace operation applied to a vector, defined as $!_V = v \to \mathrm{Tr} v$. Moreover, for all operators $f : V \otimes W \to U$, the operator $\overline{f} : V \to (W \multimap U)$ denotes the corresponding curried version, defined as $\overline{f}(v) = w \mapsto f(v, w)$. The subscripts in these operators will be omitted unless ambiguity arises.

For all ground types $X \in G$ the interpretation of $[\![X]\!]$ is postulated as a vector space $V$. Types are interpreted inductively using the unit $\mathbb{I}$, the tensor $\otimes$, and the linear map $\multimap$. Given a non-empty context $\Gamma = \Gamma', x : \mathbb{A}$, its interpretation is defined by $[\![\Gamma', x : \mathbb{A}]\!] = [\![\Gamma']\!] \otimes [\![\mathbb{A}]\!]$ if $\Gamma'$ is non-empty and $[\![\Gamma', x : \mathbb{A}]\!] = [\![\mathbb{A}]\!]$ otherwise. The empty context $-$ is interpreted as $[\![-]\!] = \mathbb{I}$. Given $X_1, ..., X_n \in V$, the $n$-tensor $(\ldots(X_1 \otimes X_2) \otimes \ldots) \otimes X_n$ is denoted as $X_1 \otimes \ldots \otimes X_n$, and similarly for operators.

"Housekeeping" operators are employed to handle interactions between context interpretation and the vectorial model. Given $\Gamma_1, \ldots, \Gamma_n$, the operator that splits $[\![\Gamma_1, \ldots, \Gamma_n]\!]$ into $[\![\Gamma_1]\!] \otimes \ldots \otimes [\![\Gamma_n]\!]$ is denoted by $\mathsf{sp}_{\Gamma_1;\ldots;\Gamma_n} : [\![\Gamma_1, \ldots, \Gamma_n]\!] \to [\![\Gamma_1]\!] \otimes \ldots \otimes [\![\Gamma_n]\!]$. On the other hand, $\mathsf{jn}_{\Gamma_1;\ldots;\Gamma_n}$ denotes the inverse of $\mathsf{sp}_{\Gamma_1;\ldots;\Gamma_n}$. Next, given $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta$, the operator permuting $x$ and $y$ is denoted by $\mathsf{exch}_{\Gamma,x:\mathbb{A},y:\mathbb{B},\Delta} : [\![\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta]\!] \to [\![\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta]\!]$. The shuffling operator $\mathsf{sh}_E : [\![E]\!] \to [\![\Gamma_1, \ldots, \Gamma_n]\!]$ is defined as a suitable composition of exchange operators.

For every operation symbol $f : \mathbb{A}_1, \ldots, \mathbb{A}_n \to \mathbb{A}$ we assume the existence of an operator $[\![f]\!] : [\![\mathbb{A}_1]\!] \otimes \ldots \otimes [\![\mathbb{A}_n]\!] \to [\![\mathbb{A}]\!]$. The interpretation of judgments is defined by induction over derivations according to the rules in Figure 5 Dahlqvist and Neves [2023a].

$$\frac{[\![\Gamma_i \rhd v_i : \mathbb{A}_i]\!] = m_i \quad f : \mathbb{A}_1, \ldots, \mathbb{A}_n \in \Sigma \quad E \in \mathsf{Sf}(\Gamma_1; \ldots; \Gamma_n)}{[\![E \rhd f(v_1, \ldots, v_n) : \mathbb{A}]\!] = [\![f]\!] \cdot (m_1 \otimes \ldots \otimes m_n) \cdot \mathsf{sp}_{\Gamma_1; \ldots; \Gamma_n} \cdot \mathsf{sh}_E} \qquad \frac{}{[\![x : \mathbb{A} \rhd x : \mathbb{A}]\!] = \mathsf{id}_{[\![\mathbb{A}]\!]}}$$

$$\frac{}{[\![- \rhd * : \mathbb{I}]\!] = \mathsf{id}_{[\![\mathbb{I}]\!]}} \qquad \frac{[\![\Gamma \rhd v : \mathbb{A} \otimes \mathbb{B}]\!] = m \quad [\![\Delta, x : \mathbb{A}, y : \mathbb{B} \rhd w : \mathbb{C}]\!] = n \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{[\![E \rhd \mathsf{pm}\ v\ \mathsf{to}\ x \otimes y.w : \mathbb{C}]\!] = n \cdot \mathsf{jn}_{\Delta; \mathbb{A}; \mathbb{B}} \cdot \alpha \cdot \mathsf{sw} \cdot (m \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma; \Delta} \cdot \mathsf{sh}_E}$$

$$\frac{[\![\Gamma \rhd v : \mathbb{A}]\!] = m \quad [\![\Delta \rhd w : \mathbb{B}]\!] = n \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{[\![E \rhd v \otimes w : \mathbb{A} \otimes \mathbb{B}]\!] = (m \otimes n) \cdot \mathsf{sp}_{\Gamma; \Delta} \cdot \mathsf{sh}_E}$$

$$\frac{[\![\Gamma \rhd v : \mathbb{I}]\!] = m \quad [\![\Delta \rhd w : \mathbb{A}]\!] = n \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{[\![E \rhd v\ \mathsf{to}\ *.w : \mathbb{A}]\!] = n \cdot \lambda \cdot (m \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma; \Delta} \cdot \mathsf{sh}_E} \qquad \frac{[\![\Gamma, x : \mathbb{A} \rhd v : \mathbb{B}]\!] = m}{[\![\Gamma \rhd \lambda x : \mathbb{A}. v : \mathbb{A} \multimap \mathbb{B}]\!] = \overline{m \cdot \mathsf{jn}_{\Gamma; \mathbb{A}}}}$$

$$\frac{[\![\Gamma \rhd v : \mathbb{A} \multimap \mathbb{B}]\!] = m \quad [\![\Delta \rhd w : \mathbb{A}]\!] = n \quad E \in \mathit{Sf}(\Gamma; \Delta)}{[\![E \rhd vw : \mathbb{A}]\!] = \mathsf{app} \cdot (m \otimes n) \cdot \mathsf{sp}_{\Gamma; \Delta} \cdot \mathsf{sh}_E} \qquad \frac{[\![\Gamma \rhd v : \mathbb{A}]\!] = f}{[\![\Gamma \rhd \mathsf{dis}(v) : \mathbb{I}]\!] = !_{[\![\mathbb{A}]\!]} \cdot f}$$

Figure 5: Judgment interpretation

Adicionar os operadores CPTP que vou usar

In the case of quantum lambda calculus, which combines classical and quantum features, it is natural to consider two distinct basic data types: a type *bit* of classical bits and a type *qbit* of quantum bits. The interpretation of these types is defined as $[\![\mathit{bit}]\!] = \mathbb{C} \oplus \mathbb{C}$ and $[\![\mathit{qbit}]\!] = \mathbb{C}^{2 \cdot 2}$. The type $\mathbb{I}$ is interpreted as $[\![\mathbb{I}]\!] = \mathbb{C}$.

The following operations are considered: $\mathit{new}\,0 : \mathbb{I} \multimap \mathit{bit}$, $\mathit{new}\,1 : \mathbb{I} \multimap \mathit{bit}$, $q : \mathit{bit} \multimap \mathit{qbit}$, $\mathit{meas} : \mathit{qbit} \to \mathit{bit}$, and $U : \mathit{qbit}, \ldots, \mathit{qbit} \to \mathit{qbit}^{\otimes n}$. Their correspondent judgment interpretation is shown in Figure 6.

$$\llbracket \textit{new}\, 0 \rrbracket : \mathbb{C} \multimap \llbracket \textit{bit} \rrbracket \qquad \llbracket \textit{new}\, 1 \rrbracket : \mathbb{C} \multimap \llbracket \textit{bit} \rrbracket \qquad \llbracket q \rrbracket : \llbracket \textit{bit} \rrbracket \multimap \llbracket \textit{qbit} \rrbracket$$

$$1 \mapsto (1,0) \qquad 1 \mapsto (0,1) \qquad (a,b) \mapsto \left(\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right)$$

$$\llbracket \textit{meas} \rrbracket : \llbracket \textit{qbit} \rrbracket \to \llbracket \textit{bit} \rrbracket \qquad \llbracket U \rrbracket : \llbracket \textit{qbit} \rrbracket^{\otimes n} \to \llbracket \textit{qbit} \rrbracket^{\otimes n}$$

$$\rho \mapsto (\mathsf{Tr}(M_0 \rho M_0^\dagger), \mathsf{Tr}(M_1 \rho M_1^\dagger)) \qquad \rho \mapsto U \rho U^\dagger$$

Figure 6: Judgment interpretation of the operations in quantum lambda calculus.

# Chapter 4

# Conditionals

## 4.1 Measurements

In order to established that the theory introduced is valid in quantum programming, it is necessary to build a model. The model can be seen as a category where the morphisms are the CPTP super-operators (quantum channels). The algebric structure of this model is given by the vector spaces. Any completely-positive and trace-preserving map has a diamond norm equal to one Watrous [2018]. Since the measurement operation is completely positive and trace-preserving, its diamond norm is equal to one. This is a desirable property, as it ensures that the measurement operation does not increase the distance between states, and as a consequence, composition of programs remains valid.

### 4.1.1 Example: Deutsch's Algorithm

In 1985, David Deutsch presented an algorithm that determines whether a function $f$ is constant for a single-bit input (*i.e.*, either equal to 1 for all $x$ or equal to 0 for all $x$) or balanced (*i.e.*, equal to 1 for half of the values of x and equal to 0 for the other half) Deutsch [1985]. Classically, to determine which case holds requires running $f$ twice. Quantumly, it suffices to run f once. The Deutsch-Jozsa Algorithm is a simple example of a quantum algorithm that outperforms its classical counterpart. The algorithm is based on the concept of a quantum oracle, which is a black box that implements a unitary transformation $U_f$ such that $U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle$, where $\oplus$ denotes addition modulo 2. The quantum circuit implementing Deutsch's algorithm is presented in Figure 7.

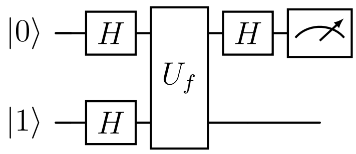

Figure 7: Quantum circuit implementing Deutsch's algorithm

Using lambda calculus, the Deutsch-Jozsa Algorithm can be expressed as:

$$\text{Deutsch} : \quad (\textit{qbit} \otimes \textit{qbit} \multimap \textit{qbit} \otimes \textit{qbit}) \multimap \textit{bit} \otimes \textit{qbit}$$

$$\begin{aligned}\text{Deutsch} = U_f &: \textit{qbit} \otimes \textit{qbit} \multimap \textit{qbit} \otimes \textit{qbit} \triangleright \\ &\text{pm } U_f(H(q\,(\textit{new}\,0\,(*))), (H(q\,(\textit{new}\,1\,(*)))) \textit{ to } q_1 \otimes q_2 \,.\, \textit{meas}(H(q_1)) \otimes q_2\end{aligned}$$

Regarding the interpretation of the Deutsch Algorithm, one has that:

$$\begin{aligned} & |0\rangle \otimes |1\rangle \\ \xmapsto{H\otimes H} \quad & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |-\rangle \end{aligned} \tag{4.1}$$

With respecto to quantum oracle $U_f$, it is possible to show that:

$$\begin{aligned} & |x\rangle \otimes |-\rangle = |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(|x\rangle \otimes |0\rangle - |x\rangle \otimes |1\rangle)) \\ \xmapsto{U_f} \; & \frac{1}{\sqrt{2}}(|x\rangle \otimes |0 \oplus f(x)\rangle - |x\rangle \otimes |1 \oplus f(x)\rangle) && \{\text{Defn. of } U_f\} \\ = \; & \frac{1}{\sqrt{2}}(|x\rangle\,|f(x)\rangle - |x\rangle\,|\neg f(x)\rangle) && \{0 \oplus x = x, 1 \oplus x = \neg x\} \\ = \; & \frac{1}{\sqrt{2}}(|x\rangle \otimes (|f(x)\rangle - |\neg f(x)\rangle)) \end{aligned} \tag{4.2}$$

Proceding by case distinction:

$$\frac{1}{\sqrt{2}}(|x\rangle \otimes (|f(x)\rangle - |\neg f(x)\rangle) = \begin{cases} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \text{if } f(x) = 0 \\ |x\rangle \otimes \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) & \text{if } f(x) = 1 \end{cases} \tag{4.3}$$

And conclude that

$$|x\rangle \otimes \frac{1}{\sqrt{2}}(|f(x)\rangle - |\neg f(x)\rangle) = (-1)^{f(x)}\,|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = (-1)^{f(x)}\,|x\rangle \otimes |-\rangle \tag{4.4}$$

Returning to the interpretation of the Deutsch Algorithm, one has that:

$$\begin{aligned}
& \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |-\rangle \\
\xmapsto{U_f} \quad & \frac{1}{\sqrt{2}}(U_f |0\rangle \otimes |-\rangle + U_f |1\rangle \otimes |-\rangle)) \\
= \quad & \frac{1}{\sqrt{2}}((-1)^{f(0)} |0\rangle \otimes |-\rangle + (-1)^{f(1)} |1\rangle \otimes |-\rangle) \\
= \quad & \begin{cases} (\pm 1) |+\rangle \otimes |-\rangle & \text{if } f(0) = f(1) \\ (\pm 1) |-\rangle \otimes |-\rangle & \text{if } f(0) \neq f(1) \end{cases} \\
\xmapsto{H \otimes I} \quad & \begin{cases} (\pm 1) |0\rangle \otimes |-\rangle & \text{if } f(0) = f(1) \\ (\pm 1) |1\rangle \otimes |-\rangle & \text{if } f(0) \neq f(1) \end{cases}
\end{aligned} \tag{4.5}$$

Attending to the interpretation of quantum states, concerning the measurement of the first qubit, one has that:

$$\begin{aligned}
& \begin{cases} |0\rangle \langle 0| \otimes |-\rangle \langle -| & \text{if } f(0) = f(1) \\ |1\rangle \langle 1| \otimes |-\rangle \langle -| & \text{if } f(0) \neq f(1) \end{cases} \\
\xmapsto{\textit{meas} \otimes I} \quad & \begin{cases} (|-\rangle \langle -|, 0) & \text{if } f(0) = f(1) \\ (0, |-\rangle \langle -|) & \text{if } f(0) \neq f(1) \end{cases}
\end{aligned} \tag{4.6}$$

A measurement error is characterized by reading a "1" as a "0" or vice versa. Furthermore, it's important to note that measurement errors do not impact all states uniformly Tannu and Qureshi [2019]. Consequently, there is a discrepancy in how frequently the state "1" is incorrectly read as "0" compared to how often the state "0" is measured as "1" or vice versa.

For example, considering there is a 10% chance of measuring a "0" as a "1" and a 30% chance of measuring a "1" as a "0", the resulting state after measurement is:

$$\begin{cases} (0.9 |-\rangle \langle -|, 0.1 |-\rangle \langle -|) & \text{if } f(0) = f(1) \\ (0.3 |-\rangle \langle -|, 0.7 |-\rangle \langle -|) & \text{if } f(0) \neq f(1) \end{cases} \tag{4.7}$$

The norm of a tuple is defined as the sum of the norms of its components, *i.e.*, for any operators $v$ and $w$:

$$\|(v, w)\| = \|v\| + \|w\| \tag{4.8}$$

As a result, the discrepancy between the ideal and actual measurement results corresponds to:

$$
\begin{aligned}
&\begin{cases}
\|(\left|-\right\rangle\left\langle-\right|, 0) - (0.9\left|-\right\rangle\left\langle-\right|, 0.1\left|-\right\rangle\left\langle-\right|)\|_\diamond & \text{if } f(0) = f(1) \\
\|(0, \left|-\right\rangle\left\langle-\right|) - (0.3\left|-\right\rangle\left\langle-\right|, 0.7\left|-\right\rangle\left\langle-\right|)\|_\diamond & \text{if } f(0) \neq f(1)
\end{cases} \\
= &\begin{cases}
\|(0.1\left|-\right\rangle\left\langle-\right|, -0.1\left|-\right\rangle\left\langle-\right|)\|_\diamond & \text{if } f(0) = f(1) \\
\|(-0.3\left|-\right\rangle\left\langle-\right|, 0.3\left|-\right\rangle\left\langle-\right|)\|_\diamond & \text{if } f(0) \neq f(1)
\end{cases} \\
= &\begin{cases}
\|0.1\left|-\right\rangle\left\langle-\right|\|_\diamond + \|-0.1\left|-\right\rangle\left\langle-\right|\|_\diamond & \text{if } f(0) = f(1) \\
\|-0.3\left|-\right\rangle\left\langle-\right|\|_\diamond + \|-0.3\left|-\right\rangle\left\langle-\right|\|_\diamond & \text{if } f(0) \neq f(1)
\end{cases}
\end{aligned}
\tag{4.9}
$$

Employing Equation 3.6, it is easily concluded that the Bloch vector of the state $\left|-\right\rangle\left\langle-\right|$ is $(-1, 0, 0)$. Consequently, the discrepancy between the ideal and actual measurement results is:

$$
\begin{aligned}
&\begin{cases}
\|(-0.1, 0, 0)\|_2 + \|(0.1, 0, 0)\left|-\right\rangle\left\langle-\right|\|_2 & \text{if } f(0) = f(1) \\
\|(0.3, 0, 0)\left|-\right\rangle\left\langle-\right|\|_2 + \|(-0.3, 0, 0)\left|-\right\rangle\left\langle-\right|\|_2 & \text{if } f(0) \neq f(1)
\end{cases} \\
= &\begin{cases}
\sqrt{(-0.1)^2 + 0^2 + 0^2} + \sqrt{(0.1)^2 + 0^2 + 0^2} & \text{if } f(0) = f(1) \\
\sqrt{(0.3)^2 + 0^2 + 0^2} + \sqrt{(-0.3)^2 + 0^2 + 0^2} & \text{if } f(0) \neq f(1)
\end{cases} \\
= &\begin{cases}
2\sqrt{0.01} & \text{if } f(0) = f(1) \\
2\sqrt{0.09} & \text{if } f(0) \neq f(1)
\end{cases}
\end{aligned}
\tag{4.10}
$$

Via the metric deductive system in Figure 3, it is easily verified that for an arbitrary error $\epsilon$:

$$
\begin{aligned}
&U_f : \textit{qbit} \otimes \textit{qbit} \multimap \textit{qbit} \otimes \textit{qbit} \triangleright \\
&\text{pm } U_f(H(q\,(\textit{new}\,0\,(*))), (H(q\,(\textit{new}\,1\,(*)))) \textit{ to } q_1 \otimes q_2 \,.\, \textit{meas}(H(q_1)) \otimes q_2 \\
=_\epsilon \\
&U_f : \textit{qbit} \otimes \textit{qbit} \multimap \textit{qbit} \otimes \textit{qbit} \triangleright \\
&\text{pm } U_f(H(q\,(\textit{new}\,0\,(*))), (H(q\,(\textit{new}\,1\,(*)))) \textit{ to } q_1 \otimes q_2 \,.\, \textit{meas}^\epsilon(H(q_1)) \otimes q_2
\end{aligned}
$$

Therefore, Deutsch $=_\epsilon$ Deutsch$^\epsilon$, and consequently, for scenario under consideration, if $f$ is a constant function, Deutsch $= 2\sqrt{0.01}$Deutsch$^{0.1,0.3}$; otherwise, Deutsch $=_{2\sqrt{0.09}}$ Deutsch$^{0.1,0.3}$.

## 4.2 Conditionals

The notion of approximate equivalence for quantum programming explored in Dahlqvist and Neves [2023a] does not encompass classical control flow. As a result, preliminary work based on Crole [1993]; Selinger [2013] has been undertaken to address the integration of conditionals.

### 4.2.1 Syntax

The term formation rules for conditionals are depicted in Figure 8.

$$\frac{\Gamma \triangleright v : \mathbb{A}}{\Gamma \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A} \oplus \mathbb{B}}\,(\mathsf{inl}) \qquad \frac{\Gamma \triangleright v : \mathbb{B}}{\Gamma \triangleright \mathsf{inr}_{\mathbb{A}}(v) : \mathbb{A} \oplus \mathbb{B}}\,(\mathsf{inr})$$

$$\frac{\Gamma \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Delta, x : \mathbb{A} \triangleright w : \mathbb{D} \quad \Delta, y : \mathbb{B} \triangleright u : \mathbb{D} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright \mathsf{case}\ v\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D}}\,(\mathsf{case})$$

Figure 8: Term formation rules for conditionals

#### Properties

The rules presented in Figure 8 are subject the properties in Theorem 4.2.1.

**Theorem 4.2.1.** *Lambda calculus with conditionals has the following properties:*

1. *for all judgements $\Gamma \triangleright v$ and $\Gamma' \triangleright v$, te($\Gamma$) $\simeq_\pi$ te($\Gamma'$);*
2. *additionally if $\Gamma \triangleright v : \mathbb{A}, \Gamma' \triangleright v : \mathbb{A}'$, and $\Gamma \simeq_\pi \Gamma'$, then $\mathbb{A}$ must be equal to $\mathbb{A}'$;*
3. *all judgements $\Gamma \triangleright v : \mathbb{A}$ have a unique derivation.*

**Proof** Regarding the first property, for the injections, taking into account the inl and inr derivations in Figure 8 and

$$\frac{\Gamma' \triangleright v : \mathbb{A}}{\Gamma' \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A} \oplus \mathbb{B}} \qquad \frac{\Gamma' \triangleright v : \mathbb{B}}{\Gamma' \triangleright \mathsf{inr}_{\mathbb{A}}(v) : \mathbb{A} \oplus \mathbb{B}}$$

, it is necessary to prove that te$(\Gamma) \simeq_\pi$ te$(\Gamma')$. By induction hypothesis, te$(\Gamma) \simeq_\pi$ te$(\Gamma')$ and te$(\Gamma) \simeq_\pi$ te$(\Gamma')$. Therefore, te$(\Gamma) \simeq_\pi$ te$(\Gamma')$.

Concerning the case statement, considering Figure 8 and

$$\frac{\Gamma' \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Delta', x : \mathbb{A} \triangleright w : \mathbb{D} \quad \Delta', y : \mathbb{B} \triangleright u : \mathbb{D} \quad E' \in \mathsf{Sf}(\Gamma'; \Delta')}{E' \triangleright \mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D}}$$

we want to prove that te$(E) \simeq_\pi$ te$(E')$. By induction hypothesis, te$(\Gamma) \simeq_\pi$ te$(\Gamma)$, te$(\Delta, x) \simeq_\pi$ te$(\Delta', x)$ and te$(\Delta, y) \simeq_\pi$ te$(\Delta', y)$. This implies that te$(\Delta) \simeq_\pi$ te$(\Delta')$. Since, $E \in \mathsf{Sf}(\Gamma; \Delta)$ and $E' \in \mathsf{Sf}(\Gamma'; \Delta')$, one has that te$(E) \simeq_\pi$ te$(\Gamma, \Delta)$ and te$(E') \simeq_\pi$ te$(\Gamma', \Delta')$. Consequently, te$(E)$ $\simeq_\pi$ te$(E')$.

With respect to the second property, for the injections, taking into account the inl and inr derivations in Figure 8 and

$$\frac{\Gamma' \triangleright v : \mathbb{A}'}{\Gamma' \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A}' \oplus \mathbb{B}} \qquad \frac{\Gamma' \triangleright v : \mathbb{B}'}{\Gamma' \triangleright \mathsf{inr}_{\mathbb{A}}(v) : \mathbb{A} \oplus \mathbb{B}'}$$

concerning the left injection it is neccessary to prove that if $\Gamma \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A} \oplus \mathbb{B}$, $\Gamma' \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A}' \oplus \mathbb{B}$, and $\Gamma \simeq_\pi \Gamma'$, then $\mathbb{A} \oplus \mathbb{B}$ must be equal to $\mathbb{A}' \oplus \mathbb{B}$. By induction hypothesis over the premises it follows that $\mathbb{A}$ must be equal to $\mathbb{A}'$. Consequentlty, $\mathbb{A} \oplus \mathbb{B}$ must be equal $\mathbb{A}' \oplus \mathbb{B}$
The same reasoning can be applied to the right injection.

Regarding the case statement in considering Figure 8 and

$$\frac{\Gamma \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Delta', x : \mathbb{A} \triangleright w : \mathbb{C}' \quad \Delta', y : \mathbb{B} \triangleright u : \mathbb{C}' \quad E' \in \mathsf{Sf}(\Gamma'; \Delta')}{E' \triangleright \mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{C}'}$$

we want to prove that if $E \triangleright \mathsf{case}\, v\, \{\mathsf{inl}(x) \Rightarrow w; \mathsf{inr}(y) \Rightarrow u\} : \mathbb{D}; \Gamma' \triangleright \mathsf{case}\, v\, \{\mathsf{inl}(x) \Rightarrow w; \mathsf{inr}(y) \Rightarrow u\} : \mathbb{D}'$, and $E \simeq_\pi E'$, then $\mathbb{D}$ must be equal to $\mathbb{D}'$. Assuming, that $E \simeq_\pi E'$ and knowing that $E \in \mathsf{Sf}(\Gamma; \Delta)$ and $E' \in \mathsf{Sf}(\Gamma'; \Delta')$, one has that

$$
\begin{aligned}
& x : \mathbb{A} \in \Delta \\
\Longrightarrow\ & x : \mathbb{A} \in E && \{E \in \mathsf{Sf}(\Gamma; \Delta)\} \\
\Longrightarrow\ & x : \mathbb{A} \in E' && \{E \simeq_\pi E'\} \\
\Longrightarrow\ & x : \mathbb{A} \in \Delta' && \{\text{All terms are well typed and contexts do not share variables}\}
\end{aligned}
$$

This proves that $\Delta \simeq_\pi \Delta'$. Therefore, by induction hypothesis on the premises of the conditional statement, one has that $\mathbb{D}$ must be equal to $\mathbb{D}'$.

Finally, concerning the third property, firstly it is necessary to demonstrate that the injections have unique derivations. This means proving that the premises of the inl and inr rules in Figure 8 and in

$$
\frac{\Gamma' \triangleright v : \mathbb{A}'}{\Gamma \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A}' \oplus \mathbb{B}} \qquad \frac{\Gamma' \triangleright v : \mathbb{B}'}{\Gamma \triangleright \mathsf{inr}_{\mathbb{A}}(v) : \mathbb{A} \oplus \mathbb{B}'}
$$

are equal, which means proving that $\Gamma = \Gamma'$. In both cases, the derivation in Figure 8 enforces that $\Gamma = \Gamma'$.

Now, it is necessary to demonstrate that the case statement in Figure 8 has a unique derivation. This means proving that the premises in Figure 8 and in

$$
\frac{\Gamma' \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Delta', x : \mathbb{A} \triangleright w : \mathbb{C}' \quad \Delta', y : \mathbb{B} \triangleright u : \mathbb{D} \quad E' \in \mathsf{Sf}(\Gamma'; \Delta')}{E \triangleright \mathsf{case}\ v\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D}}
$$

are equal, more concretely that $\Gamma = \Gamma'$ and $\Delta = \Delta'$.

$$
\begin{aligned}
& x : \mathbb{A} \in \Gamma \\
\Longrightarrow\ & x : \mathbb{A} \in E \wedge te(x : \mathbb{A}) \in \Gamma' && \{E \in \mathsf{Sf}(\Gamma; \Delta), te(\Gamma) \simeq_\pi te(\Gamma')\} \\
\Longrightarrow\ & x : \mathbb{A} \in E \wedge x : \mathbb{A} \in \Gamma' && \{E \in \mathsf{Sf}(\Gamma; \Delta), E \in \mathsf{Sf}(\Gamma'; \Delta')\}
\end{aligned}
$$

This last implication is related with the fact that in $E$, there can only exist one variable designated by $x$. Given that a shuffle preserves the relative order of the variables in each context, it follows that $\Gamma = \Gamma'$. The same reasoning can be applied to $\Delta$ and $\Delta'$, which concludes the proof.

$\square$

**Lemma 4.2.1.** *(Exchange and Substitution) For every judgement* $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright v : \mathbb{D}$ *it is possible to derive* $\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright v : \mathbb{D}$*. For all judgements* $\Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}$ *and* $\Delta \triangleright w : \mathbb{A}$ *it is possible to derive* $\Gamma, \Delta \triangleright v[w/x] : \mathbb{B}$*.*

**Proof** Regarding the exchange property, for the left injection, it is necessary to demonstrate that for every judgement $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright \mathsf{inl}_{\mathbb{D}}(v) : \mathbb{D} \oplus \mathbb{D}$, it is possible to derive $\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright \mathsf{inl}_{\mathbb{D}}(v) : \mathbb{D} \oplus \mathbb{D}$. By induction hypothesis on the premises of $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright \mathsf{inl}_{\mathbb{D}}(v) : \mathbb{D} \oplus \mathbb{D}$ and applying the inl rule, one has that:

$$\dfrac{\dfrac{\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright v : \mathbb{D}}{\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright v : \mathbb{D}}}{\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright \mathsf{inl}_{\mathbb{D}}(v) : \mathbb{D} \oplus \mathbb{D}}$$

For the right injection the proof is analogous.
With respect to the case statement it is necessary to prove that for every judgment $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright$ case $v$ $\{\mathsf{inl}_{\mathbb{E}}\ (a) \Rightarrow w; \mathsf{inr}_{\mathbb{D}}\ (b) \Rightarrow u\} : \mathbb{D}$, it is possible to derive $\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright$ case $v$ $\{\mathsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \mathsf{inr}_{\mathbb{D}}(b) \Rightarrow u\} : \mathbb{D}$. It is necessary to consider three scenarios:

1. $x : \mathbb{A}, y : \mathbb{B}$ are variables in the context of $v$;

2. $x : \mathbb{A}, y : \mathbb{B}$ are variables in the context of $w$ and $u$;

3. $x : \mathbb{A}$ is a variable in the context of $v$ and $y : \mathbb{B}$ is a variable in the context of $w$ and $u$.

With respect to the first case, by induction hypothesis and applying the case rule, one has that:

$$\frac{\dfrac{\Gamma_1, x : \mathbb{A}, y : \mathbb{B}, \Gamma_2 \triangleright v : \mathbb{D} \oplus \mathbb{E} \quad \begin{array}{c} E, a : \mathbb{D} \triangleright w : \mathbb{D} \\ E, b : \mathbb{E} \triangleright u : \mathbb{D} \end{array} \quad \Gamma, y : \mathbb{B}, x : \mathbb{A}; \Delta \in \mathsf{Sf}(\Gamma_1, x : \mathbb{A}, y : \mathbb{B}, \Gamma_2; E)}{\Gamma_1, y : \mathbb{B}, x : \mathbb{A}, \Gamma_2 \triangleright v : \mathbb{D} \oplus \mathbb{E} \quad \begin{array}{c} E, a : \mathbb{D} \triangleright w : \mathbb{D} \\ E, b : \mathbb{E} \triangleright u : \mathbb{D} \end{array} \quad \Gamma, y : \mathbb{B}, x : \mathbb{A}; \Delta \in \mathsf{Sf}(\Gamma_1, y : \mathbb{B}, x : \mathbb{A}, \Gamma_2; E)}}{\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright \ \mathsf{case}\ v \, \{\mathsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \mathsf{inr}_{\mathbb{D}}(b) \Rightarrow u\} : \mathbb{D}}$$

Next, for the second case, by induction hypothesis and applying the case rule, one has that:

$$\frac{\dfrac{E \triangleright v : \mathbb{D} \oplus \mathbb{E} \quad \begin{array}{c} \Delta_1, x : \mathbb{A}, y : \mathbb{B}, \Delta_2, a : \mathbb{D} \triangleright w : \mathbb{D} \\ \Delta_1, x : \mathbb{A}, y : \mathbb{B}, \Delta_2, b : \mathbb{E} \triangleright u : \mathbb{D} \end{array} \quad \Gamma, x : \mathbb{A}, y : \mathbb{B}; \Delta \in \mathsf{Sf}(E; \Delta_1, x : \mathbb{A}, y : \mathbb{B}, \Delta_2)}{E \triangleright v : \mathbb{D} \oplus \mathbb{E} \quad \begin{array}{c} \Delta_1, y : \mathbb{B}, x : \mathbb{A}, \Delta_2, a : \mathbb{D} \triangleright w : \mathbb{D} \\ \Delta_1, y : \mathbb{B}, x : \mathbb{A}, \Delta_2, b : \mathbb{E} \triangleright u : \mathbb{D} \end{array} \quad \Gamma, y : \mathbb{B}, x : \mathbb{A}; \Delta \in \mathsf{Sf}(E; \Delta_1, y : \mathbb{B}, x : \mathbb{A}, \Delta_2)}}{\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright \ \mathsf{case}\ v \, \{\mathsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \mathsf{inr}_{\mathbb{D}}(b) \Rightarrow u\} : \mathbb{D}}$$

Finally, for the third case, considering the premises

$$\Gamma_1, x : \mathbb{A}, \Gamma_2 \triangleright v : \mathbb{D} \oplus \mathbb{E} \quad \begin{array}{c} \Delta_1, y : \mathbb{B}, \Delta_2, a : \mathbb{D} \triangleright w : \mathbb{D} \\ \Delta_1, y : \mathbb{B}, \Delta_2, b : \mathbb{E} \triangleright u : \mathbb{D} \end{array} \quad \begin{array}{c} \Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \in \\ \mathsf{Sf}(\Gamma_1, x : \mathbb{A}, \Gamma_2; \Delta_2, y : \mathbb{B}, \Delta_1) \end{array}$$

and attending to the definition of shuffle, a possibility for $\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \in \mathsf{Sf}(\Gamma_1, x : \mathbb{A}, \Gamma_2; \Delta_2, y : \mathbb{B}, \Delta_1)$, given that in exchanging these variables the relative order of the variables in $\Gamma_1, x : \mathbb{A}, \Gamma_2$ and $\Delta_2, y : \mathbb{B}, \Delta_1$ is preserved. As a result, it is possible to derive $\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright \mathsf{case}\ v \, \{\mathsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \mathsf{inr}_{\mathbb{D}}(b) \Rightarrow u\} : \mathbb{D}$.

With respect to the substitution property, for the left injection is necessary to demonstrate that for all judgements $\Gamma, x : \mathbb{D} \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A} \oplus \mathbb{B}$ and $\Delta \triangleright w : \mathbb{D}$ it is possible to derive $\Gamma, \Delta \triangleright \mathsf{inl}_{\mathbb{B}}(v[w/x]) : \mathbb{A} \oplus \mathbb{B}$. By induction hypothesis, and applying the inl rule, one has that:

$$\begin{array}{c} \Gamma, x : \mathbb{D} \triangleright v : \mathbb{A} \quad \Delta \triangleright w : \mathbb{D} \\ \hline \Gamma, \Delta \triangleright v[w/x] : \mathbb{A} \\ \hline \Gamma, \Delta \triangleright \mathsf{inl}_{\mathbb{B}}(v[w/x]) : \mathbb{A} \oplus \mathbb{B} \\ \hline \Gamma, \Delta \triangleright \mathsf{inl}_{\mathbb{B}}(v)[w/x] : \mathbb{A} \oplus \mathbb{B} \end{array}$$

For the right injection the proof is analogous.

Regarding the case statement it is necessary to prove that for all judgements $E, z : \mathbb{D} \triangleright$ $\mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{A}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{B}}(y) \Rightarrow u\} : \mathbb{D}$ and $\Sigma \triangleright a : \mathbb{D}$ it is possible to derive $E, \Sigma \triangleright$ $\mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{A}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{B}}(y) \Rightarrow u\}[a/z] : \mathbb{D}$. In this case, it is necessary to consider two scenarios:

1. $z : \mathbb{D}$ is a variable in the context of $v$;
2. $z : \mathbb{D}$ is a variable in the context of $w$ and $u$.

Regarding the first case, by induction and applying the case rule, one has that:

$$\begin{array}{c} \Gamma_1, z : \mathbb{D} \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Gamma_2, x : \mathbb{A} \triangleright w : \mathbb{D} \quad \Gamma_2, y : \mathbb{B} \triangleright u : \mathbb{D} \quad E, z : \mathbb{D} \in \mathsf{Sf}(\Gamma_1, z : \mathbb{D}; \Gamma_2) \quad \Delta \triangleright a : \mathbb{D} \\ \hline \Gamma_1, \Delta \triangleright v[a/z] : \mathbb{A} \oplus \mathbb{B} \quad \Gamma_2, x : \mathbb{A} \triangleright w : \mathbb{D} \quad \Gamma_2, y : \mathbb{B} \triangleright u : \mathbb{D} \quad E, \Delta \in \mathsf{Sf}(\Gamma_1, \Delta; \Gamma_2) \\ \hline E, \Delta \triangleright \mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w[a/z]; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u[a/z]\} : \mathbb{D} \\ \hline E, \Delta \triangleright \mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\}[a/z] : \mathbb{D} \end{array}$$

The second case is similar to the first one, applying the exchange property, then the induction,followed by the exchange property once more, and finally the case rule, one has that

$$\begin{array}{c} \Gamma_2, z : \mathbb{D}, y : \mathbb{B} \triangleright u : \mathbb{D} \\ \Gamma_1 \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Gamma_2, z : \mathbb{D}, x : \mathbb{A} \triangleright w : \mathbb{D} \qquad E, z : \mathbb{D} \in \mathsf{Sf}(\Gamma_1; \Gamma_2, z : \mathbb{D}) \quad \Delta \triangleright a : \mathbb{D} \\ \hline \Gamma_2, y : \mathbb{B}, z : \mathbb{D} \triangleright u : \mathbb{D} \\ \Gamma_1 \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Gamma_2, x : \mathbb{A}, z : \mathbb{D} \triangleright w : \mathbb{D} \qquad E, z : \mathbb{D} \in \mathsf{Sf}(\Gamma_1; \Gamma_2, z : \mathbb{D}) \quad \Delta \triangleright a : \mathbb{D} \\ \hline \Gamma_1 \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Gamma_2, x : \mathbb{A}, \Delta \triangleright w[a/z] : \mathbb{D} \quad \Gamma_2, y : \mathbb{B}, \Delta \triangleright u[a/z] : \mathbb{D} \quad E, \Delta \in \mathsf{Sf}(\Gamma_1; \Gamma_2, \Delta) \\ \hline \Gamma_1 \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Gamma_2, \Delta, x : \mathbb{A} \triangleright w[a/z] : \mathbb{D} \quad \Gamma_2, \Delta, y : \mathbb{B} \triangleright u[a/z] : \mathbb{D} \quad E, \Delta \in \mathsf{Sf}(\Gamma_1; \Gamma_2, \Delta) \\ \hline E, \Delta \triangleright \mathsf{case}\, v[a/z]\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D} \\ \hline E, \Delta \triangleright \mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\}[a/z] : \mathbb{D} \end{array}$$

$\square$

### 4.2.2 Interpretation

Considering $v \in V$, $w \in W$, and $u \in U$ where $V, W, U$ represent vector spaces, $\mathsf{IL}_V : V \to V \oplus W$, denotes the left injection operator, defined as $\mathsf{IL}_V = v \mapsto (v, 0)$; $\mathsf{IR}_V : V \to W \oplus V$, denotes the right injection operator, defined as $\mathsf{IR}_V = v \mapsto (0, v)$; and $\mathsf{dist}_{V,W,U} : V \otimes (W \oplus U) \to (V \otimes W) \oplus (V \otimes U)$, denotes the distributive property of the tensor product over the direct sum, defined as $\mathsf{dist}_{V,W,U} = v \otimes (w, u) \mapsto (v \otimes w, v \otimes u)$. The subscripts in these operators will be omitted unless ambiguity arises. Moreover, the operation either corresponds to:

$$\frac{\begin{array}{c} V \to U \\ W \to U \end{array}}{[T, S] : V \oplus W \to U} \tag{4.11}$$

$$[T, S] = (v, w) \mapsto T(v) + S(w)$$

The interpretation of conditionals is illustrated in Figure 9.

$$\frac{\dfrac{[\![\Gamma \triangleright v : \mathbb{A}]\!] = m}{[\![\Gamma \triangleright \mathsf{inl}(v) : \mathbb{A} \oplus \mathbb{B}]\!] = \mathsf{IL} \cdot m} \qquad \dfrac{[\![\Gamma \triangleright v : \mathbb{B}]\!] = m}{[\![\Gamma \triangleright \mathsf{inr}(v) : \mathbb{A} \oplus \mathbb{B}]\!] = \mathsf{IR} \cdot m} \\ [\![\Gamma \triangleright v : \mathbb{A} \oplus \mathbb{B}]\!] = b \quad [\![\Delta, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] = p \quad [\![\Delta, y : \mathbb{B} \triangleright u : \mathbb{D}]\!] = q \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{[\![E \triangleright \mathsf{case}\, v\, \{\mathsf{inl}(x) \Rightarrow w; \mathsf{inr}(y) \Rightarrow u\} : \mathbb{D}]\!] = [p \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, q \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (b \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E} \tag{4.12}$$

Figure 9: Judgment interpretation for conditionals

**Proof** In order to validate the judgment interpretation for conditionals, it is necessary to demonstrate its correctness.

For the booleans:

$$\begin{array}{c} [\![\Gamma]\!] \xrightarrow{m} [\![\mathbb{A}]\!] \xrightarrow{\mathsf{IL}} [\![\mathbb{A} \oplus \mathbb{B}]\!] \\ [\![\Gamma]\!] \xrightarrow{m} [\![\mathbb{B}]\!] \xrightarrow{\mathsf{IR}} [\![\mathbb{A} \oplus \mathbb{B}]\!] \end{array} \tag{4.13}$$

Now, for the case statement:

$$\begin{aligned}
\llbracket E \rrbracket &\xrightarrow{\mathsf{sh}_E} \llbracket \Gamma, \Delta \rrbracket \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} \llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket \xrightarrow{b \otimes \mathsf{id}} (\llbracket \mathbb{A} \rrbracket \oplus \llbracket \mathbb{B} \rrbracket) \otimes \llbracket \Delta \rrbracket \xrightarrow{\mathsf{sw}} \llbracket \Delta \rrbracket \otimes (\llbracket \mathbb{A} \rrbracket \oplus \llbracket \mathbb{B} \rrbracket) \\
&\xrightarrow{\mathsf{dist}} (\llbracket \Delta \rrbracket \otimes \llbracket \mathbb{A} \rrbracket) \oplus (\llbracket \Delta \rrbracket \otimes \llbracket \mathbb{B} \rrbracket) \xrightarrow{[p \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, q \cdot \mathsf{jn}_{\Delta;\mathbb{B}}]} \llbracket \mathbb{D} \rrbracket
\end{aligned} \tag{4.14}$$

Next, it is necessary to demonstrate that the interpretation of exchange and substitution holds for injections and the case statement.

**Lemma 4.2.2.** *(Exchange) For all judgements $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright v : \mathbb{D}$, the following equation holds: $\llbracket \Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright v : \mathbb{D} \rrbracket = \llbracket \Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright v : \mathbb{D} \rrbracket \cdot$ exch$_{\Gamma, \underline{\mathbb{A}, \mathbb{B}}, \Delta}$*

**Proof** Firstly, for the left injection,

$$\begin{aligned}
&\llbracket \Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright \mathsf{inl}_{\mathbb{D}}(v) : \mathbb{D} \oplus \mathbb{D} \rrbracket \\
=& \mathsf{I}\mathsf{L} \cdot \llbracket \Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \triangleright v : \mathbb{D} \rrbracket \\
=& \mathsf{I}\mathsf{L} \cdot \llbracket \Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright v : \mathbb{D} \rrbracket \cdot \mathsf{exch}_{\Gamma, \underline{\mathbb{A}, \mathbb{B}}, \Delta} && \{\text{by induction hypothesis}\} \\
=& \llbracket \Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \triangleright \mathsf{inl}_{\mathbb{D}}(v) : \mathbb{D} \oplus \mathbb{D} \rrbracket
\end{aligned}$$

The proof for the right injection is analogous.
Regarding the case statement, it is necessary to consider three scenarios:

1. $x : \mathbb{A}, y : \mathbb{B}$ are variables in the context of $v$;

2. $x : \mathbb{A}, y : \mathbb{B}$ are variables in the context of $w$ and $u$;

3. $x : \mathbb{A}$ is a variable in the context of $v$ and $y : \mathbb{B}$ is a variable in the context of $w$ and $u$.

For the first case,

$$
\begin{aligned}
&[\![\Gamma, x:\mathbb{A}, y:\mathbb{B}, \Delta \triangleright \textsf{case}\ \{\textsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \textsf{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{D}]\!]\\
=&\ \big[[\![E, a:\mathbb{D} \triangleright w]\!] \cdot \textsf{jn}_{E;\mathbb{D}}, [\![E, b:\mathbb{E} \triangleright u]\!] \cdot \textsf{jn}_{E;\mathbb{E}}\big] \cdot \textsf{dist} \cdot \textsf{sw}\\
&\cdot ([\![\Gamma_1, x:\mathbb{A}, y:\mathbb{B}, \Gamma_2 \triangleright v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \textsf{id}) \cdot \textsf{sp}_{\Gamma_1,\mathbb{A},\mathbb{B},\Gamma_2;E} \cdot \textsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta}\\
=&\ \big[[\![E, a:\mathbb{D} \triangleright w]\!] \cdot \textsf{jn}_{E;\mathbb{D}}, [\![E, b:\mathbb{E} \triangleright u]\!] \cdot \textsf{jn}_{E;\mathbb{E}}\big] \cdot \textsf{dist} \cdot \textsf{sw}\\
&\cdot ([\![\Gamma_1, y:\mathbb{B}, x:\mathbb{A}, \Gamma_2 \triangleright v : \mathbb{D}]\!] \cdot \textsf{exch}_{\Gamma_1,\underline{\mathbb{A},\mathbb{B}},\Gamma_2} \otimes \textsf{id}) \cdot \textsf{sp}_{\Gamma_1,\mathbb{A},\mathbb{B},\Gamma_2;E} \cdot \textsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta}\\
=&\ \big[[\![E, a:\mathbb{D} \triangleright w]\!] \cdot \textsf{jn}_{E;\mathbb{D}}, [\![E, b:\mathbb{E} \triangleright u]\!] \cdot \textsf{jn}_{E;\mathbb{E}}\big] \cdot \textsf{dist} \cdot \textsf{sw}\\
&\cdot ([\![\Gamma_1, y:\mathbb{B}, x:\mathbb{A}, \Gamma_2 \triangleright v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \textsf{id}) \cdot \textsf{sp}_{\Gamma_1,\mathbb{B},\mathbb{A},\Gamma_2;E} \cdot \textsf{exch}_{\Gamma_1,\underline{\mathbb{A},\mathbb{B}},\Gamma_2,E} \cdot \textsf{jn}_{\Gamma_1,\mathbb{A},\mathbb{B},\Gamma_2;E}\\
&\cdot \textsf{sp}_{\Gamma_1,\mathbb{A},\mathbb{B},\Gamma_2;E} \cdot \textsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta}\\
=&\ \big[[\![E, a:\mathbb{D} \triangleright w]\!] \cdot \textsf{jn}_{E;\mathbb{D}}, [\![E, b:\mathbb{E} \triangleright u]\!] \cdot \textsf{jn}_{E;\mathbb{E}}\big] \cdot \textsf{dist} \cdot \textsf{sw}\\
&\cdot ([\![\Gamma_1, y:\mathbb{B}, x:\mathbb{A}, \Gamma_2 \triangleright v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \textsf{id}) \cdot \textsf{sp}_{\Gamma_1,\mathbb{B},\mathbb{A},\Gamma_2;E} \cdot \textsf{exch}_{\Gamma,\underline{\mathbb{A},\mathbb{B}},\Delta} \cdot \textsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta}\\
=&\ \big[[\![E, a:\mathbb{D} \triangleright w]\!] \cdot \textsf{jn}_{E;\mathbb{D}}, [\![E, b:\mathbb{E} \triangleright u]\!] \cdot \textsf{jn}_{E;\mathbb{E}}\big] \cdot \textsf{dist} \cdot \textsf{sw}\\
&\cdot ([\![\Gamma_1, y:\mathbb{B}, x:\mathbb{A}, \Gamma_2 \triangleright v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \textsf{id}) \cdot \textsf{sp}_{\Gamma_1,\mathbb{B},\mathbb{A},\Gamma_2;E} \cdot \textsf{sh}_{\Gamma,\mathbb{B},\mathbb{A},\Delta} \cdot \textsf{exch}_{\Gamma,\underline{\mathbb{A},\mathbb{B}},\Delta}\\
=&\ [\![\Gamma, y:\mathbb{B}, x:\mathbb{A}, \Delta \triangleright \textsf{case}\ \{\textsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \textsf{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{D}]\!] \cdot \textsf{exch}_{\Gamma,\underline{\mathbb{A},\mathbb{B}},\Delta}
\end{aligned}
$$

Now, for the second case,

$$
\begin{aligned}
&[\![\Gamma, x:\mathbb{A}, y:\mathbb{B}, \Delta \rhd \text{ case } \{\mathsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \mathsf{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{D}]\!]\\
&= \big[[\![\Delta_1, x:\mathbb{A}, y:\mathbb{B}, \Delta_2, a:\mathbb{D} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_1,\mathbb{A},\mathbb{B},\Delta_2,\mathbb{D}}, [\![\Delta_1, x:\mathbb{A}, y:\mathbb{B}, \Delta_2, b:\mathbb{E} \rhd u]\!] \cdot \mathsf{jn}_{\Delta_1,\mathbb{A},\mathbb{B},\Delta_2,\mathbb{D}}\big]\\
&\quad \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot ([\![E \rhd v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \mathsf{id}) \cdot \mathsf{sp}_{E;\Delta_1,\mathbb{A},\mathbb{B},\Delta_2} \cdot \mathsf{sh}_{\Gamma;\mathbb{A},\mathbb{B},\Delta}\\
&= [[\![\Delta_1, y:\mathbb{B}, x:\mathbb{A}, \Delta_2, a:\mathbb{D} \rhd w]\!] \cdot \mathsf{exch}_{\Delta_1,\underline{\mathbb{A},\mathbb{B}},\Delta_2,\mathbb{D}} \cdot \mathsf{jn}_{\Delta_1,\mathbb{A},\mathbb{B},\Delta_2,\mathbb{D}},\\
&\quad [\![\Delta_1, y:\mathbb{B}, x:\mathbb{A}, \Delta_2, b:\mathbb{E} \rhd w]\!] \cdot \mathsf{exch}_{\Delta_1,\underline{\mathbb{A},\mathbb{B}},\Delta_2,\mathbb{E}} \cdot \mathsf{jn}_{\Delta_1,\mathbb{A},\mathbb{B},\Delta_2,\mathbb{E}}] \cdot \mathsf{dist} \cdot \mathsf{sw}\\
&\quad \cdot ([\![E \rhd v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \mathsf{id}) \cdot \mathsf{sp}_{E;\Delta_1,\mathbb{A},\mathbb{B},\Delta_2} \cdot \mathsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta}\\
&= [[\![\Delta_1, y:\mathbb{B}, x:\mathbb{A}, \Delta_2, a:\mathbb{D} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_1,\mathbb{B},\mathbb{A},\Delta_2,\mathbb{D}}, [\![\Delta_1, y:\mathbb{B}, x:\mathbb{A}, \Delta_2, b:\mathbb{E} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_1,\mathbb{B},\mathbb{A},\Delta_2,\mathbb{E}}]\\
&\quad \cdot \mathsf{dist} \cdot \mathsf{sp}_{\Delta_1,\mathbb{B},\mathbb{A},\Delta_2;\mathbb{D}\oplus\mathbb{E}} \cdot \mathsf{exch}_{\Delta_1,\underline{\mathbb{A},\mathbb{B}},\Delta_2,\mathbb{D}\oplus\mathbb{E}} \cdot \mathsf{jn}_{\Delta_1,\mathbb{A},\mathbb{B},\Delta_2;\mathbb{D}\oplus\mathbb{E}} \cdot \mathsf{sw} \cdot ([\![E \rhd v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \mathsf{id})\\
&\quad \cdot \mathsf{sp}_{E;\Delta_1,\mathbb{A},\mathbb{B},\Delta_2} \cdot \mathsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta}\\
&= [[\![\Delta_1, y:\mathbb{B}, x:\mathbb{A}, \Delta_2, a:\mathbb{D} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_1,\mathbb{B},\mathbb{A},\Delta_2,\mathbb{D}}, [\![\Delta_1, y:\mathbb{B}, x:\mathbb{A}, \Delta_2, b:\mathbb{E} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_1,\mathbb{B},\mathbb{A},\Delta_2,\mathbb{E}}]\\
&\quad \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot ([\![E \rhd v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta_1,\mathbb{B},\mathbb{A},\Delta_2} \cdot \mathsf{exch}_{\Gamma,\Delta_1,\underline{\mathbb{A},\mathbb{B}},\Delta_2} \cdot \mathsf{jn}_{\Gamma;\Delta_1,\mathbb{A},\mathbb{B},\Delta_2}\\
&\quad \cdot \mathsf{sp}_{E;\Delta_1,\mathbb{A},\mathbb{B},\Delta_2} \cdot \mathsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta}\\
&= [[\![\Delta_1, y:\mathbb{B}, x:\mathbb{A}, \Delta_2, a:\mathbb{D} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_1,\mathbb{B},\mathbb{A},\Delta_2,\mathbb{D}}, [\![\Delta_1, y:\mathbb{B}, x:\mathbb{A}, \Delta_2, b:\mathbb{E} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_1,\mathbb{B},\mathbb{A},\Delta_2,\mathbb{E}}]\\
&\quad \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot ([\![E \rhd v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \mathsf{id}) \cdot \mathsf{sp}_{E;\Delta_1,\mathbb{B},\mathbb{A},\Delta_2} \cdot \mathsf{sh}_{\Gamma,\mathbb{B},\mathbb{A},\Delta} \cdot \mathsf{exch}_{\Gamma,\underline{\mathbb{A},\mathbb{B}},\Delta}\\
&= [\![\Gamma, y:\mathbb{B}, x:\mathbb{A}, \Delta \rhd \text{ case } \{\mathsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \mathsf{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{D}]\!] \cdot \mathsf{exch}_{\Gamma,\underline{\mathbb{A},\mathbb{B}},\Delta}
\end{aligned}
$$

Finally, for the third case, note that, the shuffle operator is permutation of typed variables that preserves the relative order of the variables in both contexts, and, as a result, $\Gamma, \mathbb{B}, \mathbb{A}, \Delta \in \mathsf{Sf}(\Gamma_1, x:\mathbb{A}, \Gamma_2; \Delta_2, y:\mathbb{B}, \Delta_1)$. Thus, the proof is as follows:

$$
\begin{aligned}
&[\![\Gamma, x:\mathbb{A}, y:\mathbb{B}, \Delta \rhd \text{ case } \{\mathsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \mathsf{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{D}]\!]\\
&= \big[[\![\Delta_2, y:\mathbb{B}, \Delta_1, a:\mathbb{D} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_2,\mathbb{B},\Delta_1;\mathbb{D}}, [\![\Delta_2, y:\mathbb{B}, \Delta_1, b:\mathbb{E} \rhd u]\!] \cdot \mathsf{jn}_{\Delta_2,\mathbb{B},\Delta_1;\mathbb{E}}\big] \cdot \mathsf{dist} \cdot \mathsf{sw}\\
&\quad \cdot ([\![\Delta_2, y:\mathbb{B}, \Delta_1 \rhd v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma_1,x:\mathbb{A},\Gamma_2;\Delta_2,y:\mathbb{B},\Delta_1} \cdot \mathsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta}\\
&= \big[[\![\Delta_2, y:\mathbb{B}, \Delta_1, a:\mathbb{D} \rhd w]\!] \cdot \mathsf{jn}_{\Delta_2,\mathbb{B},\Delta_1;\mathbb{D}}, [\![\Delta_2, y:\mathbb{B}, \Delta_1, b:\mathbb{E} \rhd u]\!] \cdot \mathsf{jn}_{\Delta_2,\mathbb{B},\Delta_1;\mathbb{E}}\big] \cdot \mathsf{dist} \cdot \mathsf{sw}\\
&\quad \cdot ([\![\Delta_2, y:\mathbb{B}, \Delta_1 \rhd v : \mathbb{D} \oplus \mathbb{E}]\!] \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma_1,x:\mathbb{A},\Gamma_2;\Delta_2,y:\mathbb{B},\Delta_1} \cdot \mathsf{sh}_{\Gamma,\mathbb{A},\mathbb{B},\Delta} \cdot \mathsf{exch}_{\Gamma,\underline{\mathbb{A},\mathbb{B}},\Delta}\\
&= [\![\Gamma, y:\mathbb{B}, x:\mathbb{A}, \Delta \rhd \text{ case } \{\mathsf{inl}_{\mathbb{E}}(a) \Rightarrow w; \mathsf{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{D}]\!] \cdot \mathsf{exch}_{\Gamma,\underline{\mathbb{A},\mathbb{B}},\Delta}
\end{aligned}
$$

$\square$

**Lemma 4.2.3.** *(Substitution) For all judgements $\Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}$ and $\Delta \triangleright w : \mathbb{A}$ the following equation holds:* $[\![\Gamma, \Delta \triangleright v[w/x] : \mathbb{B}]\!] = [\![\Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}]\!] \cdot jn_{\Gamma;\mathbb{A}} \cdot (id \otimes [\![\Delta \triangleright w : \mathbb{A}]\!]) \cdot sp_{\Gamma;\Delta}$

Regarding the left injection,

$$
\begin{aligned}
&[\![\Gamma, \Delta \triangleright \mathsf{inl}_{\mathbb{B}}(v)[w/x] : \mathbb{A} \oplus \mathbb{B}]\!] \\
=&\mathsf{IL} \cdot [\![\Gamma, \Delta \triangleright v[w/x] : \mathbb{A}]\!] \\
=&\mathsf{IL} \cdot [\![\Gamma, x : \mathbb{D} \triangleright v : \mathbb{A}]\!] \cdot \mathsf{jn}_{\Gamma;\mathbb{D}} \cdot (\mathsf{id} \otimes [\![\Delta \triangleright w : \mathbb{D}]\!]) \cdot \mathsf{sp}_{\Gamma;\Delta} \\
=&[\![\Gamma, x : \mathbb{D} \triangleright \mathsf{inl}_{\mathbb{B}}(v)[w/x] : \mathbb{A} \oplus \mathbb{B}]\!] \cdot \mathsf{jn}_{\Gamma;\mathbb{D}} \cdot (\mathsf{id} \otimes [\![\Delta \triangleright w : \mathbb{D}]\!]) \cdot \mathsf{sp}_{\Gamma;\Delta}
\end{aligned}
$$

The proof for the right injection is analogous.

With respect to the case statement, in this case, it is necessary to consider two scenarios:

1. $z : \mathbb{D}$ is a variable in the context of $v$;
2. $z : \mathbb{D}$ is a variable in the context of $w$ and $u$.

For the first case,

ver Prof Renato onde tem reticencias

$$
\begin{aligned}
&[\![E, \Delta \triangleright \mathsf{case}\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\}[a/z] : \mathbb{D}]\!] \\
=& \left[[\![\Gamma_2, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{A}}, [\![\Gamma_2, y : \mathbb{B} \triangleright u]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{B}}\right] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot ([\![\Gamma_1, \Delta \triangleright v[a/z] : \mathbb{A} \oplus \mathbb{B}]\!] \\
&\otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma_1,\Delta;\Gamma_2} \cdot \mathsf{sh}_{E,\Delta} \\
=& \left[[\![\Gamma_2, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{A}}, [\![\Gamma_2, y : \mathbb{B} \triangleright u]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{B}}\right] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (([\![\Gamma_1, z : \mathbb{D} \triangleright v : \mathbb{A} \oplus \mathbb{B}]\!] \\
&\cdot \mathsf{jn}_{\Gamma_1;\mathbb{D}} \cdot (\mathsf{id} \otimes [\![\Delta \triangleright a : \mathbb{D}]\!]) \cdot \mathsf{sp}_{\Gamma_1;\Delta}) \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma_1,\Delta;\Gamma_2} \cdot \mathsf{sh}_{E,\Delta} \\
=& \left[[\![\Gamma_2, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{A}}, [\![\Gamma_2, y : \mathbb{B} \triangleright u]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{B}}\right] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot ([\![\Gamma_1, z : \mathbb{D} \triangleright v : \mathbb{A} \oplus \mathbb{B}]\!] \\
&\otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma_1,\mathbb{D};\Gamma_2} \cdot \mathsf{jn}_{\Gamma_1;\mathbb{D};\Gamma_2} \cdot (\mathsf{id} \otimes [\![\Delta \triangleright a : \mathbb{D}]\!] \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma_1;\Delta;\Gamma_2} \cdot \mathsf{sh}_{E,\Delta} \\
=& \left[[\![\Gamma_2, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{A}}, [\![\Gamma_2, y : \mathbb{B} \triangleright u]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{B}}\right] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot ([\![\Gamma_1, z : \mathbb{D} \triangleright v : \mathbb{A} \oplus \mathbb{B}]\!] \\
&\otimes \mathsf{id})\ldots \\
=& \left[[\![\Gamma_2, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{A}}, [\![\Gamma_2, y : \mathbb{B} \triangleright u]\!] \cdot \mathsf{jn}_{\Gamma_2;\mathbb{B}}\right] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot ([\![\Gamma_1, z : \mathbb{D} \triangleright v : \mathbb{A} \oplus \mathbb{B}]\!] \\
&\otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma_1,\mathbb{D};\Gamma_2} \cdot \mathsf{sh}_{E,\mathbb{D}} \cdot \mathsf{jn}_{E;\mathbb{D}} \cdot (\mathsf{id} \otimes [\![\Delta \triangleright a : \mathbb{D}]\!]) \cdot \mathsf{sp}_{E;\Delta} \\
=& [\![E, z : \mathbb{D} \triangleright \mathsf{case}\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D}]\!] \cdot \mathsf{jn}_{E;\mathbb{D}} \cdot (\mathsf{id} \otimes [\![\Delta \triangleright a : \mathbb{D}]\!]) \cdot \mathsf{sp}_{E;\Delta}
\end{aligned}
$$

For the second case,

$$
\begin{aligned}
&[\![E,\Delta \triangleright \text{ case } \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\}[a/z] : \mathbb{D}]\!]\\
&= \big[[\![\Gamma_2,\Delta,x:\mathbb{A}\triangleright w[a/z]:\mathbb{D}]\!]\cdot \mathsf{jn}_{\Gamma_2,\Delta,\mathbb{A}}, [\![\Gamma_2,\Delta,y:\mathbb{B}\triangleright u[a/z]:\mathbb{D}]\!]\cdot \mathsf{jn}_{\Gamma_2,\Delta,\mathbb{B}}\big]\cdot \mathsf{dist}\cdot \mathsf{sw}\\
&\quad \cdot ([\![\Gamma_1\triangleright v:\mathbb{A}\oplus\mathbb{B}]\!]\otimes \mathsf{id})\cdot \mathsf{sp}_{\Gamma_1;\Delta,\Gamma_2}\cdot \mathsf{sh}_{E,\Delta}\\
&= [[\![\Gamma_2,x:\mathbb{A},\Delta\triangleright w[a/z]:\mathbb{D}]\!]\cdot \mathsf{exch}_{\Gamma_2,\Delta,\mathbb{A}}\cdot \mathsf{jn}_{\Gamma_2,\Delta,\mathbb{A}}, [\![\Gamma_2,y:\mathbb{B},\Delta\triangleright u[a/z]:\mathbb{D}]\!]\cdot \mathsf{exch}_{\Gamma_2,\Delta,\mathbb{B}}\\
&\quad \cdot \mathsf{jn}_{\Gamma_2,\Delta,\mathbb{B}}]\cdot \mathsf{dist}\cdot \mathsf{sw}\cdot ([\![\Gamma_1\triangleright v:\mathbb{A}\oplus\mathbb{B}]\!]\otimes \mathsf{id})\cdot \mathsf{sp}_{\Gamma_1;\Delta,\Gamma_2}\cdot \mathsf{sh}_{E,\Delta}\\
&= [[\![\Gamma_2,x:\mathbb{A},z:\mathbb{D}\triangleright w:\mathbb{D}]\!]\cdot \mathsf{jn}_{\Gamma_2,\mathbb{A};\mathbb{D}}\cdot (\mathsf{id}\otimes [\![\Delta\triangleright a:\mathbb{D}]\!])\cdot \mathsf{sp}_{\Gamma_2,\mathbb{A};\Delta}\cdot \mathsf{exch}_{\Gamma_2,\Delta,\mathbb{A}}\cdot \mathsf{jn}_{\Gamma_2,\Delta,\mathbb{A}},\\
&\quad [\![\Gamma_2,y:\mathbb{B},z:\mathbb{D}\triangleright u:\mathbb{D}]\!]\cdot \mathsf{jn}_{\Gamma_2,\mathbb{A};\mathbb{D}}\cdot (\mathsf{id}\otimes [\![\Delta\triangleright a:\mathbb{D}]\!])\cdot \mathsf{sp}_{\Gamma_2,\mathbb{B};\Delta}\cdot \mathsf{exch}_{\Gamma_2,\Delta,\mathbb{B}}]\cdot \mathsf{jn}_{\Gamma_2,\Delta,\mathbb{B}}\\
&\quad \cdot \mathsf{dist}\cdot \mathsf{sw}\cdot ([\![\Gamma_1\triangleright v:\mathbb{A}\oplus\mathbb{B}]\!]\otimes \mathsf{id})\cdot \mathsf{sp}_{\Gamma_1;\Delta,\Gamma_2}\cdot \mathsf{sh}_{E,\Delta}\\
&= [[\![\Gamma_2,x:\mathbb{A},z:\mathbb{D}\triangleright w:\mathbb{D}]\!]\cdot \mathsf{exch}_{\Gamma_2,\mathbb{D},\mathbb{A}}\cdot \mathsf{jn}_{\Gamma_2,\mathbb{D},\mathbb{A}}, [\![\Gamma_2,y:\mathbb{B},z:\mathbb{D}\triangleright u:\mathbb{D}]\!]\cdot \mathsf{exch}_{\Gamma_2,\mathbb{D},\mathbb{B}}\\
&\quad \cdot \mathsf{jn}_{\Gamma_2,\mathbb{D},\mathbb{B}}]\cdot \mathsf{dist}\cdot \mathsf{sp}_{\Gamma_2,\mathbb{D};\mathbb{A}\oplus\mathbb{B}}\cdot \mathsf{jn}_{\Gamma_2;\mathbb{D};\mathbb{A}\oplus\mathbb{B}}\cdot (\mathsf{id}\otimes [\![\Delta\triangleright a:\mathbb{D}]\!]\otimes \mathsf{id})\cdot \mathsf{sp}_{\Gamma_2;\Delta,\mathbb{A}\oplus\mathbb{B}}\cdot \mathsf{sw}\\
&\quad \cdot ([\![\Gamma_1\triangleright v:\mathbb{A}\oplus\mathbb{B}]\!]\otimes \mathsf{id})\cdot \mathsf{sp}_{\Gamma_1;\Delta,\Gamma_2}\cdot \mathsf{sh}_{E,\Delta}\\
&= [[\![\Gamma_2,x:\mathbb{A},z:\mathbb{D}\triangleright w:\mathbb{D}]\!]\cdot \mathsf{exch}_{\Gamma_2,\mathbb{D},\mathbb{A}}\cdot \mathsf{jn}_{\Gamma_2,\mathbb{D},\mathbb{A}}, [\![\Gamma_2,y:\mathbb{B},z:\mathbb{D}\triangleright u:\mathbb{D}]\!]\cdot \mathsf{exch}_{\Gamma_2,\mathbb{D},\mathbb{B}}\\
&\quad \cdot \mathsf{jn}_{\Gamma_2,\mathbb{D},\mathbb{B}}]\cdot \mathsf{dist}\cdot \mathsf{sw}\cdot ([\![\Gamma_1\triangleright v:\mathbb{A}\oplus\mathbb{B}]\!]\otimes \mathsf{id})\cdot \mathsf{sp}_{\Gamma_1;\Gamma_2,\mathbb{D}}\cdot \mathsf{jn}_{\Gamma_1;\Gamma_2;\mathbb{D}}\cdot (\mathsf{id}\otimes [\![\Delta\triangleright a:\mathbb{D}]\!]\otimes \mathsf{id})\\
&\quad \cdot \mathsf{sp}_{\Gamma_1;\Delta;\Gamma_2}\cdot \mathsf{sh}_{E,\Delta}\\
&= [[\![\Gamma_2,x:\mathbb{A},z:\mathbb{D}\triangleright w:\mathbb{D}]\!]\cdot \mathsf{exch}_{\Gamma_2,\mathbb{D},\mathbb{A}}\cdot \mathsf{jn}_{\Gamma_2,\mathbb{D},\mathbb{A}}, [\![\Gamma_2,y:\mathbb{B},z:\mathbb{D}\triangleright u:\mathbb{D}]\!]\cdot \mathsf{exch}_{\Gamma_2,\mathbb{D},\mathbb{B}}\\
&\quad \cdot \mathsf{jn}_{\Gamma_2,\mathbb{D},\mathbb{B}}]\cdot \mathsf{dist}\cdot \mathsf{sw}\cdot ([\![\Gamma_1\triangleright v:\mathbb{A}\oplus\mathbb{B}]\!]\otimes \mathsf{id})\cdot \mathsf{sp}_{\Gamma_1;\Gamma_2,\mathbb{D}}\cdot \mathsf{sh}_{E,\mathbb{D}}\cdot \mathsf{jn}_{E;\mathbb{D}}\cdot (\mathsf{id}\otimes [\![\Delta\triangleright a:\mathbb{D}]\!])\\
&\quad \cdot \mathsf{sp}_{E;\Delta}\\
&= \big[[\![\Gamma_2,z:\mathbb{D},x:\mathbb{A}\triangleright w:\mathbb{D}]\!]\cdot \mathsf{jn}_{\Gamma_2,\mathbb{D},\mathbb{A}}, [\![\Gamma_2,z:\mathbb{D},y:\mathbb{B}\triangleright u:\mathbb{D}]\!]\cdot \mathsf{jn}_{\Gamma_2,\mathbb{D},\mathbb{B}}\big]\cdot \mathsf{dist}\cdot \mathsf{sw}\\
&\quad \cdot ([\![\Gamma_1\triangleright v:\mathbb{A}\oplus\mathbb{B}]\!]\otimes \mathsf{id})\cdot \mathsf{sp}_{\Gamma_1;\Gamma_2,\mathbb{D}}\cdot \mathsf{sh}_{E,\mathbb{D}}\cdot \mathsf{jn}_{E;\mathbb{D}}\cdot (\mathsf{id}\otimes [\![\Delta\triangleright a:\mathbb{D}]\!])\cdot \mathsf{sp}_{E;\Delta}\\
&= [\![E,z:\mathbb{D}\triangleright \text{ case } \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\}[a/z] : \mathbb{D}]\!]\cdot \mathsf{jn}_{E;\mathbb{D}}\cdot (\mathsf{id}\otimes [\![\Delta\triangleright a:\mathbb{D}]\!])\cdot \mathsf{sp}_{E;\Delta}
\end{aligned}
$$

□

### 4.2.3 $\beta$ and $\eta$ Equations

In this subsection it will be shown that the following equations hold for the model considered.

$$(\beta^{inl}_{case}):\ \Delta,\Gamma \triangleright \mathsf{case}\ \mathsf{inl}_{\mathbb{B}}(v)\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} = w[v/x] : \mathbb{D}$$

$$(\beta^{inr}_{case}):\ \Delta,\Gamma \triangleright \mathsf{case}\ \mathsf{inr}_{\mathbb{A}}(v)\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} = w[v/y] : \mathbb{D}$$

$$(\eta_{case}):\ \Delta,\Gamma \triangleright \mathsf{case}\ (v)\ \{\mathsf{inl}_{\mathbb{B}}(y) \Rightarrow w[\mathsf{inl}_{\mathbb{B}}(y)/x];\ \mathsf{inr}_{\mathbb{A}}(z) \Rightarrow w[\mathsf{inr}_{\mathbb{A}}(z)/x]\} = w[v/x] : \mathbb{D}$$

**Proof** It is necessary to demonstrate that

$$[\![\Delta,\Gamma \triangleright \mathsf{case}\ \mathsf{inl}_{\mathbb{B}}(v)\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D}]\!] = [\![\Delta,\Gamma \triangleright w[v/x] : \mathbb{D}]\!]$$

$$[\![\Delta,\Gamma \triangleright \mathsf{case}\ \mathsf{inr}_{\mathbb{A}}(v)\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D}]\!] = [\![\Delta,\Gamma \triangleright w[v/x] : \mathbb{D}]\!]$$

$$[\![\Delta,\Gamma \triangleright \mathsf{case}\ (v)\ \{\mathsf{inl}_{\mathbb{B}}(y) \Rightarrow w[\mathsf{inl}_{\mathbb{B}}(y)/x];\ \mathsf{inr}_{\mathbb{A}}(z) \Rightarrow w[\mathsf{inr}_{\mathbb{A}}(z)/x]\} : \mathbb{D}]\!] = [\![\Delta,\Gamma \triangleright w[v/x] : \mathbb{D}]\!]$$

Regarding the first equation,

$$\begin{aligned}
&[\![\Delta,\Gamma \triangleright \mathsf{case}\ \mathsf{inl}_{\mathbb{B}}(v)\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D}]\!] \\
=&[[\![\Delta, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, [\![\Delta, y : \mathbb{B} \triangleright u : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot ([\![\Gamma \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A} \oplus \mathbb{B}]\!] \\
&\otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&[[\![\Delta, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, [\![\Delta, y : \mathbb{B} \triangleright u : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (\mathsf{IL} \cdot [\![\Gamma \triangleright v : \mathbb{A}]\!] \otimes \mathsf{id}) \\
&\cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&[[\![\Delta, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, [\![\Delta, y : \mathbb{B} \triangleright u : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (\mathsf{IL} \otimes \mathsf{id}) \cdot ([\![\Gamma \triangleright v : \mathbb{A}]\!] \\
&\otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&[[\![\Delta, x : \mathbb{A} \triangleright w : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, [\![\Delta, y : \mathbb{B} \triangleright u : \mathbb{D}]\!] \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{dist} \cdot (\mathsf{id} \otimes \mathsf{IL}) \cdot \mathsf{sw} \cdot ([\![\Gamma \triangleright v : \mathbb{A}]\!] \\
&\otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma}
\end{aligned}$$

Given that $[\mathsf{id} \otimes \mathsf{IL}, \mathsf{id} \otimes \mathsf{IR}] \cdot \mathsf{IL} = \mathsf{id} \otimes \mathsf{IL}$, it follows that the following diagram commutes.

$$\begin{tikzcd}
X \otimes Y \arrow[r, "\mathsf{id} \otimes \mathsf{IL}"] \arrow[d, "\mathsf{IL}"'] & X \otimes (Y \oplus Y) \arrow[dl, bend right, "\mathsf{dist}"'] \\
X \otimes Y \oplus X \otimes Z \arrow[ur, bend right, "{[\mathsf{id} \otimes \mathsf{IL}, \mathsf{id} \otimes \mathsf{IR}]}"'] &
\end{tikzcd}$$

And as a result, $\mathsf{dist} \cdot (\mathsf{id} \otimes \mathsf{Il}) = \mathsf{Il}$. Therefore,

$$
\begin{aligned}
&[\llbracket \Delta, x : \mathbb{A} \rhd w : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, \llbracket \Delta, y : \mathbb{B} \rhd u : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{dist} \cdot (\mathsf{id} \otimes \mathsf{Il}) \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \rrbracket \\
&\quad \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&[\llbracket \Delta, x : \mathbb{A} \rhd w : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, \llbracket \Delta, y : \mathbb{B} \rhd u : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{Il} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \\
&\quad \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&\llbracket \Delta, x : \mathbb{A} \rhd w : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&\llbracket \Delta, x : \mathbb{A} \rhd w : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}} \cdot (\mathsf{id} \otimes \llbracket \Gamma \rhd v : \mathbb{A} \rrbracket) \cdot \mathsf{sw} \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&\llbracket \Delta, x : \mathbb{A} \rhd w : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}} \cdot (\mathsf{id} \otimes \llbracket \Gamma \rhd v : \mathbb{A} \rrbracket) \cdot \mathsf{sp}_{\Delta;\Gamma} \\
=&\llbracket w[v/x] : \mathbb{D} \rrbracket
\end{aligned}
$$

The proof for the second equation is analogous to the first one.
Taking into account that $[\mathsf{id} \otimes \mathsf{Il}, \mathsf{id} \otimes \mathsf{Ir}] \cdot \mathsf{Ir} = \mathsf{id} \otimes \mathsf{Ir}$, it follows that the following diagram commutes.

$$
\begin{tikzcd}
X \otimes Y \arrow[r, "\mathsf{id} \otimes \mathsf{Ir}"] \arrow[d, "\mathsf{Ir}"'] & X \otimes (Z \oplus Y) \arrow[dl, bend right, "\mathsf{dist}"'] \\
X \otimes Z \oplus X \otimes Y \arrow[ur, bend right, "{[\mathsf{id} \otimes \mathsf{Il}, \mathsf{id} \otimes \mathsf{Ir}]}"'] &
\end{tikzcd}
$$

Consequentlty, $\mathsf{dist} \cdot (\mathsf{id} \otimes \mathsf{Ir}) = \mathsf{Ir}$. Thus,

$$
\begin{aligned}
&\llbracket \Delta, \Gamma \rhd \ \mathsf{case}\ \mathsf{inr}_{\mathbb{A}}(v)\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w; \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D} \rrbracket \\
=&[\llbracket \Delta, x : \mathbb{A} \rhd w : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, \llbracket \Delta, y : \mathbb{B} \rhd u : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{dist} \cdot (\mathsf{id} \otimes \mathsf{Ir}) \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{B} \rrbracket \\
&\quad \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&[\llbracket \Delta, x : \mathbb{A} \rhd w : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, \llbracket \Delta, y : \mathbb{B} \rhd u : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{Ir} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{B} \rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \\
&\quad \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&\llbracket \Delta, y : \mathbb{B} \rhd w : \mathbb{D} \rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{B}} \cdot (\mathsf{id} \otimes \llbracket \Gamma \rhd v : \mathbb{B} \rrbracket) \cdot \mathsf{sp}_{\Delta;\Gamma} \\
=&\llbracket w[v/y] : \mathbb{D} \rrbracket
\end{aligned}
$$

With respect to the third equation,

$$
\begin{aligned}
&\llbracket \Delta, \Gamma \rhd \text{ case } (v)\,\{\mathsf{inl}_{\mathbb{B}}(y) \Rightarrow w[\mathsf{inl}_{\mathbb{B}}(y)/x]; \mathsf{inr}_{\mathbb{A}}(z) \Rightarrow w[\mathsf{inr}_{\mathbb{A}}(z)/x]\} : \mathbb{D}\rrbracket \\
=&[\llbracket \Delta, y : \mathbb{B} \rhd w[\mathsf{inl}_{\mathbb{B}}(y)/x] : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, \llbracket \Delta, z : \mathbb{A} \rhd w[\mathsf{inr}_{\mathbb{A}}(z)/x] : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \cdot \mathsf{dist} \cdot \mathsf{sw} \\
&\cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&[\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\mathsf{id} \otimes \llbracket y : \mathbb{A} \rhd \mathsf{inl}_{\mathbb{B}}(y) : \mathbb{A} \oplus \mathbb{B}\rrbracket) \cdot \mathsf{sp}_{\Delta;\mathbb{A}} \cdot \mathsf{jn}_{\Delta;\mathbb{A}}, \\
&\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\mathsf{id} \otimes \llbracket z : \mathbb{B} \rhd \mathsf{inr}_{\mathbb{A}}(z) : \mathbb{A} \oplus \mathbb{B}\rrbracket) \cdot \mathsf{sp}_{\Delta;\mathbb{B}} \cdot \mathsf{jn}_{\Delta;\mathbb{B}}] \\
&\cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&[\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\mathsf{id} \otimes \mathsf{IL} \cdot \llbracket y : \mathbb{A} \rhd y : \mathbb{A}\rrbracket), \llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \\
&\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\mathsf{id} \otimes \mathsf{IR} \cdot \llbracket z : \mathbb{B} \rhd z : \mathbb{B}\rrbracket)] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket) \\
&\otimes \mathsf{id} \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&[\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\mathsf{id} \otimes \mathsf{IL} \cdot \mathsf{id}_{\llbracket\mathbb{A}\rrbracket}), \llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \\
&\cdot (\mathsf{id} \otimes \mathsf{IR} \cdot \mathsf{id}_{\llbracket\mathbb{B}\rrbracket})] \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\mathsf{id} \otimes [\mathsf{IL} \cdot \mathsf{id}_{\llbracket\mathbb{A}\rrbracket}, \mathsf{IR} \cdot \mathsf{id}_{\llbracket\mathbb{B}\rrbracket}]) \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket \\
&\otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma}
\end{aligned}
$$

Considering that $[\mathsf{IL} \cdot \mathsf{id}, \mathsf{IR} \cdot \mathsf{id}] = \mathsf{id} + \mathsf{id} = \mathsf{id}$, it follows that the following diagram commutes.

$$
\begin{tikzcd}
X \otimes (Y \oplus Y) \arrow[r, bend left=10, "{[\mathsf{id} \otimes \mathsf{IL}, \mathsf{id} \otimes \mathsf{IR}]}"] \arrow[d, "\mathsf{id} \otimes \mathsf{id}"'] & X \otimes Y \oplus X \otimes Z \arrow[l, bend left=10, "\mathsf{dist}"] \arrow[dl, "{\mathsf{id} \otimes [\mathsf{IL} \cdot \mathsf{id}, \mathsf{IR} \cdot \mathsf{id}]}"] \\
X \otimes (Y \oplus Y) &
\end{tikzcd}
$$

And as a result, one has that $\mathsf{id} \otimes [\mathsf{IL} \cdot \mathsf{id}_{\llbracket\mathbb{A}\rrbracket}, \mathsf{IR} \cdot \mathsf{id}_{\llbracket\mathbb{B}\rrbracket}] \cdot \mathsf{dist} = \mathsf{id} \otimes \mathsf{id}$. Therefore,

$$
\begin{aligned}
&\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\mathsf{id} \otimes [\mathsf{IL} \cdot \mathsf{id}_{\llbracket\mathbb{A}\rrbracket}, \mathsf{IR} \cdot \mathsf{id}_{\llbracket\mathbb{B}\rrbracket}]) \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket \\
&\otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\mathsf{id} \otimes \mathsf{id}) \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot \mathsf{sw} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_{\Delta;\Gamma} \\
=&\llbracket \Delta, x : \mathbb{A} \oplus \mathbb{B} \rhd w : \mathbb{D}\rrbracket \cdot \mathsf{jn}_{\Delta;\mathbb{A}\oplus\mathbb{B}} \cdot (\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B}\rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Delta;\Gamma} \\
=&\llbracket w[v/x] : \mathbb{D}\rrbracket
\end{aligned}
$$

$\square$

### 4.2.4 Metric equations

The metric equations for conditionals are presented in Figure 10. Note that the first two equations are redundant.

$$\frac{v =_q w}{\mathsf{inl}(v) =_q \mathsf{inl}(w)} \qquad \frac{v =_q w}{\mathsf{inr}(v) =_q \mathsf{inr}(w)}$$

$$\frac{v =_q v' \quad w =_r w' \quad u =_s u'}{\mathsf{case}\ v\ \{\mathsf{inl}(x) \Rightarrow w; \mathsf{inr}(y) \Rightarrow u\} =_{q+\mathsf{max}(r,s)} \mathsf{case}\ v'\ \{\mathsf{inl}(x) \Rightarrow w'; \mathsf{inr}(y) \Rightarrow u'\}}$$

Figure 10: Metric equational system for condicionals

**Proof** In order to validate the metric equational system for conditionals, it is necessary to demonstrate its correctness.

The diamond norm is a particular instance of the operator norm.

For the **injections**:

Firstly, it is necessary to prove that the identity operator $I$ has a norm equal to 1.

**Lemma 4.2.4.** $\|I\|_\sigma = 1$

*Proof.* Using the definition of operator norm in Definition 3.1.23, it follows that:

$$\|I\|_\sigma = \mathsf{sup}\{\|I(v)\| \mid \|v\| = 1\} = \mathsf{sup}\{\|v\| \mid \|v\| = 1\} = 1 \tag{4.15}$$

Thereafter, it is imperative to show that the injection operators IL and IR are have a norm equal to 1.

**Lemma 4.2.5.** $\|\mathsf{IL}\|_\sigma = 1$

**Lemma 4.2.6.** $\|\mathsf{IR}\|_\sigma = 1$

*Proof.* Employing the definition of operator norm as defined in Definition 3.1.23, it ensues that:

$$\begin{aligned}
\|\mathsf{IL}\|_\sigma &= \mathsf{sup}\{\|\mathsf{IL}(v)\| \mid \|v\| = 1\} = \mathsf{sup}\{\|(v,0)\| \mid \|v\| = 1\} = \mathsf{sup}\{\|v\| + \|0\| \mid \|v\| = 1\} \\
&= \mathsf{sup}\{\|v\| \ + 0 \mid \|v\| = 1\} && \{\text{Positive definiteness}\} \\
&= \mathsf{sup}\{\|v\| \mid \|v\| = 1\} = 1
\end{aligned}$$

(4.16)

The proof for Lemma 4.2.6 is analogous to the proof for Lemma 4.2.5.

$$\begin{aligned} \|\mathsf{IR}\|_\sigma &= \sup\{\|\mathsf{IR}(v)\| \mid \|v\| = 1\} = \sup\{\|(0,v)\| \mid \|v\| = 1\} = \sup\{\|0\| + \|v\| \mid \|v\| = 1\} \\ &= \sup\{0 + \|v\| \; | \|v\| = 1\} && \{\text{Positive definiteness}\} \\ &= \sup\{\|v\| \mid \|v\| = 1\} = 1 \end{aligned} \tag{4.17}$$

Futhermore, given the submultiplicative property of the operator norm, for any super-operators $P$ and $Q$,where $\|P\|_\sigma = 1$ the following holds:

**Lemma 4.2.7.** $\|PQ\|_\sigma \leq \|Q\|_\sigma, \quad \|P\|_\sigma = 1$

Using these properties it is possible to prove the validity of the metric equations for the injections. Demonstrating the correctness of the metric equations for the injections is equivalent to proving that for any non-negative rational $q$ and super-operators $v$ and $w$ such that $d(v,w) \leq q$, where $d(v,w)$ represents the distance between $v$ and $w$ the following holds:

**Theorem 4.2.2.** $d(\mathsf{IL}(v), \mathsf{IL}(w)) \leq q$

**Theorem 4.2.3.** $d(\mathsf{IR}(v), \mathsf{IR}(w)) \leq q$

*Proof.* In the quantum paradigm, the distance between two super-operators $E$ and $E'$ corresponds to the diamond norm between $E$ and $E'$. Therefore,

$$d(v,w) \leq q \Leftrightarrow \|v \otimes I - w \otimes I\|_\sigma \leq q \tag{4.18}$$

As a result, to prove that $d(\mathsf{IL}(v), \mathsf{IL}(w)) \leq q$, it suffices to show that:

$$\|\mathsf{IL} \otimes I(v \otimes I) - \mathsf{IL} \otimes I(w \otimes I)\|_\sigma \leq \|v \otimes I - w \otimes I\|_\sigma \tag{4.19}$$

$$\|\mathsf{IR} \otimes I(v \otimes I) - \mathsf{IR} \otimes I(w \otimes I)\|_\sigma \leq \|v \otimes I - w \otimes I\|_\sigma \tag{4.20}$$

Given that IL and IR possess a norm equal to 1, as established by Lemmas 4.2.5 and 4.2.6 respectively, and considering the multiplicative property of the operator norm with respect to tensor products alongside the fact that the identity operator also exhibits a norm equal to 1, as demonstrated in Lemma 4.2.4, it follows that both $\|\mathsf{IL} \otimes I\|_\sigma$ and $\|\mathsf{IR} \otimes I\|_\sigma$ are equal to one 1. Hence, by Lemma 4.2.7,

$$\|\text{IL} \otimes I(v \otimes I) - \text{IL} \otimes I(w \otimes I)\|_\sigma = \|\text{IL} \otimes I(v \otimes I - w \otimes I)\|_\sigma \leq \|v \otimes I - w \otimes I\|_\sigma \tag{4.21}$$

$$\|\text{IR} \otimes I(v \otimes I) - \text{IR} \otimes I(w \otimes I)\|_\sigma = \|\text{IR} \otimes I(v \otimes I - w \otimes I)\|_\sigma \leq \|v \otimes I - w \otimes I\|_\sigma \tag{4.22}$$

Now, regarding the metric equation for the **conditional statement**, before validating its correctness, it is necessary to prove a few intermediate results.
The first step is to demonstrate that for any super-operators $P$ and $Q$ the following holds:

**Lemma 4.2.8.** $\|[P, Q]\|_\sigma \leq \max\{\|P\|_\sigma, \|Q\|_\sigma\}$

*Proof.* Employing the definition of the operator norm in Definition 3.1.23, it follows that:

$$\begin{aligned}
&\text{sup}\{\|[P, Q](v)\| \mid \|v\| = 1\} \leq \text{max}\{\text{sup}\{\|P(w)\| \mid \|w\| = 1\}, \text{sup}\{\|Q(u)\| \mid \|u\| = 1\}\} \\
&= \text{sup}\{\|[P, Q](w, u)\| \mid \|w\| + \|u\| = 1\} \leq \text{max}\{\text{sup}\{\|P(w)\| \mid \|w\| = 1, \|Q(u)\| \mid \|u\| = 1\}\} \\
&= \text{sup}\{\|P(w) + Q(u)\| \mid \|w\| + \|u\| = 1\| = 1\} \leq \text{max}\{\text{sup}\{\|P(w)\| \mid \|w\| = 1, \|Q(u)\| \mid \|u\| = 1\}\} \\
&= \text{sup}\{\|P(w) + Q(u)\| \mid \|w\| + \|u\| = 1\} \leq \text{sup}\{\text{max}\{\|P(w)\| \mid \|w\| = 1, \|Q(u)\| \mid \|u\| = 1\}\}
\end{aligned} \tag{4.23}$$

Therefore, by the triangle inequality, proving the inequality in Equation 4.24 suffices to establish Lemma 4.2.8.

$$\text{sup}\{\|P(w)\| + \|Q(u)\| \mid \|w\| + \|u\| = 1\} \leq \text{sup}\{\text{max}\{\|P(w)\| \mid \|w\| = 1, \|Q(u)\| \mid \|u\| = 1\}\} \tag{4.24}$$

This can be rewritten as:

$$\|w\| + \|u\| = 1 \land \text{sup}\{\|P(w)\| + \|Q(u)\|\} \leq \text{max}\left\{\frac{1}{\|w\|}\|P(w)\|, \frac{1}{\|u\|}\|Q(u)\|\right\} \tag{4.25}$$

As a result,

$$\|w\| + \|u\| = 1 \land \text{sup}\{\|P(w)\| + \|Q(u)\|\} \leq \text{max}\left\{\left\|P\left(\frac{1}{\|w\|}w\right)\right\|, \left\|Q\left(\frac{1}{\|u\|}u\right)\right\|\right\} \tag{4.26}$$

This is equivalent to demonstrating that for $a + b = 1$,

$$x + y \leq \max\left\{\frac{1}{a}x, \frac{1}{b}y\right\} \tag{4.27}$$

This is done by arguing by *reductio ad absurdum*, *i.e.*, supposing otherwise leads to a contradiction:

$$
\begin{aligned}
& x + y > \max\left\{\frac{1}{a}x, \frac{1}{b}y\right\} \\
& \Rightarrow x + y > \frac{1}{a}x \wedge x + y > \frac{1}{b}y \\
& \Rightarrow a(x+y) > x \wedge b(x+y) > y \\
& \Rightarrow ax + ay > x \wedge bx + by > y \\
& \Rightarrow ax + ay > x \wedge (1-a)x + (1-a)y > y \\
& \Rightarrow ax + ay > x \wedge x - ax + y - ay > y \\
& \Rightarrow x < ax + ay \wedge x > ax + ay
\end{aligned}
\tag{4.28}
$$

Subsequently, it is imperative to prove that:

**Lemma 4.2.9.** $i = [\mathsf{IL} \otimes I, \mathsf{IR} \otimes I]$ *is an isomorphism.*

*Proof.* The proof is as follows:
For any vector spaces $V$, $W$, and $U$, $i : (V \otimes U) \oplus (W \otimes U) \to (V \oplus W) \otimes U$. If $V$ has dimension $m$, $W$ has dimension $n$, and $U$ has dimension $o$, then the space $(V \otimes U) \oplus (W \otimes U)$ has dimension $mo + no = (m+n) \cdot o$. Similarly, the space $(V \oplus W) \otimes U$ has dimension $(m+n) \cdot o$. Hence, the spaces have the same dimension. Given that spaces with the same dimension are isomorphic Hefferon [2006], it follows that $i$ is an isomorphism.

Next, it is necessary to demonstrate that for any operators $P$ and $Q$, the identity operator $I$, and an isomorphism $i = [\mathsf{IL} \otimes I, \mathsf{IR} \otimes I]$ the following holds:

**Lemma 4.2.10.** $([P, Q] \otimes I) \cdot i = [P \otimes I, Q \otimes I]$

Which is equivalent to showing that for any vector spaces $V$, $W$, $U$, and $Z$ and super-operators $P : V \to Z$, $Q : W \to Z$, and $I : U \to U$, the following diagram holds:

$$
\begin{tikzcd}
V \otimes U \oplus W \otimes U \arrow[r, "i"] \arrow[d, "{[P \otimes I, Q \otimes I]}"'] & (V \oplus W) \otimes U \arrow[dl, "{[P, Q] \otimes I}"] \\
Z \otimes U &
\end{tikzcd}
$$

*Proof.* The proof is straightforward:

$$
\begin{aligned}
&([P,Q]\otimes I)\cdot[\mathsf{IL}\otimes I,\mathsf{IR}\otimes I] \\
&=[([P,Q]\otimes I)\cdot(\mathsf{IL}\otimes I),([P,Q]\otimes I)\cdot(\mathsf{IR}\otimes I)] \\
&=[P\otimes I,Q\otimes I]
\end{aligned}
\tag{4.29}
$$

Furhtermore, it is imperative to show that the following relation holds:

**Lemma 4.2.11.** $[P\otimes I,Q\otimes I]\cdot i^{-1}=[P,Q]\otimes I$

Demonstrating this is equivalent to establishing that for any vector spaces $V$, $W$, $U$, and $Z$, and super-operators $P:V\to Z$, $Q:W\to Z$, and $I:U\to U$, the following diagram commutes:

$$
\begin{tikzcd}
V\otimes U\oplus W\otimes U \arrow[d, "{[P\otimes I,Q\otimes I]}"'] & (V\oplus W)\otimes U \arrow[l, "i^{-1}"'] \arrow[dl, "{[P,Q]\otimes I}"] \\
Z\otimes U &
\end{tikzcd}
$$

*Proof.* The proof is as follows:

$$
\begin{aligned}
&([P,Q]\otimes I)\cdot i=[P\otimes I,Q\otimes I] && \{\text{Lemma 4.2.10}\} \\
\Leftrightarrow\ &([P,Q]\otimes I)\cdot i\cdot i^{-1}=[P\otimes I,Q\otimes I]\cdot i^{-1} \\
\Leftrightarrow\ &([P,Q]\otimes I)=[P\otimes I,Q\otimes I]\cdot i^{-1} && \{\text{Lemma 4.2.9}\}
\end{aligned}
\tag{4.30}
$$

With Lemma 4.2.10 and Lemma 4.2.11, it has been proved that the diagram below is valid:

$$
\begin{tikzcd}
V\otimes U\oplus W\otimes U \arrow[r, bend left, "i"] \arrow[d, "{[P\otimes I,Q\otimes I]}"'] & (V\oplus W)\otimes U \arrow[l, bend left, "i^{-1}"] \arrow[dl, "{[P,Q]\otimes I}"] \\
Z\otimes U &
\end{tikzcd}
$$

Now, it is possivel to prove that $i$ has a norm equal to 1.

**Lemma 4.2.12.** $\|i\|_\sigma\geq 1$

*Proof.* Considering the vector $(v\otimes u,0)$ with $\|(v\otimes u,0)\|=1$, and attending the multiplicative property of the operator norm with respect to tensor products, along with the definition of the norm of a tuple as in Equation 4.8, it holds that $\|v\| = 1$ and $\|u\| = 1$. Therefore, using this same property and definition, it is possible to demonstrate that the following holds:

$$\|[\mathsf{IL} \otimes I, \mathsf{IR} \otimes I](v \otimes u, 0)\| = \|(v, 0) \otimes u\| = (\|v\| + \|0\|)\|u\| = \|v\|\|u\| = 1 \tag{4.31}$$

Given the definition of the operator norm as presented in Definition 3.1.23, it follows that:

$$\|[\mathsf{IL} \otimes I, \mathsf{IR} \otimes I]\|_\sigma = \mathsf{sup}\{\|[\mathsf{IL} \otimes I, \mathsf{IR} \otimes I](a)\| \mid \|a\| = 1\} \tag{4.32}$$

From this, it can be deduced that $\|i\|_\sigma \geq 1$.
Subsequently, it is possible to demontrate that $i^{-1}$ has a norm greater than or equal to 1,

**Lemma 4.2.13.** $\|i^{-1}\|_\sigma \leq 1$

*Proof.* Given that $i$ is an isomophism, it follows that

$$\begin{aligned}
&\|i \cdot i^{-1}\|_\sigma = 1 \\
\leq &\|i\|_\sigma \cdot \|i^{-1}\|_\sigma = 1 && \{\text{Norm submultiplicative with respect to compositions}\} \\
\leq &1 \cdot \|i^{-1}\|_\sigma = 1 && \{\text{Lemma 4.2.13}\} \\
\Leftrightarrow &\|i^{-1}\|_\sigma = 1
\end{aligned} \tag{4.33}$$

Next, one has to prove that for any super-operators $P$ and $Q$ and their respective erroneous versions $P'$ and $Q'$, the following holds:

**Lemma 4.2.14.** $\|P \cdot Q \otimes I - P' \cdot Q' \otimes I\|_\sigma \leq \|(P - P') \otimes I\|_\sigma + \|(Q - Q') \otimes I\|_\sigma$

*Proof.* Applying the triangle inequality, he submultiplicative property of the operator norm with respect to compositions, and given that a positive and trace-preserving operator map, $E$, has norm $\|E \otimes I\|_\sigma = 1$ (Watrous [2018]), it follows that:

$$\begin{aligned}
&\|P \cdot Q \otimes I - P' \cdot Q' \otimes I\|_\sigma \\
&= \|P \cdot Q \otimes I - P \cdot Q' \otimes I + P \cdot Q' \otimes I - P' \cdot Q' \otimes I\|_\sigma \\
&\leq \|P \cdot Q \otimes I - P \cdot Q' \otimes I\|_\sigma + \|P \cdot Q' \otimes I - P' \cdot Q' \otimes I\|_\sigma \\
&\leq \|P\|_\sigma \|Q \otimes I - Q' \otimes I\|_\sigma + \|P \otimes I - P' \otimes I\|_\sigma \|Q'\|_\sigma \\
&= \|P\|_\sigma \|(Q - Q') \otimes I\|_\sigma + \|(P - P') \otimes I\|_\sigma \|Q'\|_\sigma \\
&= \|(P - P') \otimes I\|_\sigma + \|(Q - Q') \otimes I\|_\sigma
\end{aligned} \tag{4.34}$$

Finally, considering the the semantics the conditional statement in Figure 9, demonstrating the conditional statement rule in Figure 10 includes proving that for any super-operators $P$, $Q$, $P'$ and $Q'$, denoting the distance between super-operators $A$ and $B$ as $d(A, B)$, the following holds:

**Lemma 4.2.15.** *$d([P, Q], [P', Q']) \leq max\{d(P, P'), d(Q, Q')\}$*

*Proof.* In the quantum paradigm, the distance between two super-operators corresponds to the diamond norm between the two super-operators. Hence, denoting $[\mathsf{IL} \otimes I, \mathsf{IR} \otimes I]$ by $i$ it follows that:

$$
\begin{aligned}
&\mathsf{d}([P, Q], [P', Q']) \\
&= \|[P, Q] \otimes I - [P', Q'] \otimes I\|_\sigma \\
&= \|[P \otimes I, Q \otimes I] \cdot i^{-1} - [P' \otimes I, Q' \otimes I] \cdot i^{-1}\|_\sigma && \{\text{Lemma 4.2.11}\} \\
&= \|[P - P' \otimes I, Q - Q' \otimes I] \cdot i^{-1}\|_\sigma \\
&\leq \|[P - P' \otimes I, Q - Q' \otimes I]\| \|\|i^{-1}\|\|_\sigma && \{\text{Norm submultiplicative with respect to compositions}\} \\
&\leq \|[(P - P') \otimes I, (Q - Q') \otimes I]\|_\sigma && \{\text{Lemma 4.2.13}\}
\end{aligned}
\tag{4.35}
$$

and

$$
\begin{aligned}
&\max\{\mathsf{d}(P, P'), \mathsf{d}(Q, Q')\} \\
=&\max\{\|P \otimes I - P' \otimes I\|_\sigma, \|Q \otimes I - Q' \otimes I\|_\sigma\} \\
=&\max\{\|(P - P') \otimes I\|_\sigma, \|(Q - Q') \otimes I\|_\sigma\}
\end{aligned}
\tag{4.36}
$$

Finally, by Lemma 4.2.8, it can be deduced that $\mathsf{d}([P, Q], [P', Q']) \leq \max\{\mathsf{d}(P, P'), \mathsf{d}(Q, Q')\}$, which concludes the proof of theorem Lemma 4.2.15.

An alternative method to establish Theorem 4.2.4 is now presented.

*Proof.* The proof is as follows:

$$
\begin{aligned}
&\mathsf{d}([P,Q],[P',Q'])\\
&= \|[P,Q]\otimes I - [P',Q']\otimes I\|_\sigma\\
&= \|([P,Q]-[P',Q'])\otimes I\|_\sigma\\
&= \|[P-P',Q-Q']\otimes I\|_\sigma\\
&= \|[P-P',Q-Q']\|_\sigma\|I\|_\sigma && \{\text{Norm multiplicative with respect to tensor products}\}\\
&= \|[P-P',Q-Q']\|_\sigma && \{\text{Lemma 4.2.4}\}
\end{aligned}
\tag{4.37}
$$

Moreover,

$$
\begin{aligned}
&\max\{\mathsf{d}(P,P'),\mathsf{d}(Q,Q')\}\\
=&\max\{\|P\otimes I - P'\otimes I\|_\sigma, \|Q\otimes I - Q'\otimes I\|_\sigma\}\\
=&\max\{\|(P-P')\otimes I\|_\sigma, \|(Q-Q')\otimes I\|_\sigma\}\\
=&\max\{\|(P-P')\|_\sigma\|I\|_\sigma, \|(Q-Q')\|_\sigma\|I\|_\sigma\} && \{\text{Norm multiplicative with}\\
& && \text{respect to tensor products}\}\\
=&\max\{\|(P-P')\|_\sigma, \|(Q-Q')\|_\sigma\} && \{\text{Lemma 4.2.4}\}
\end{aligned}
\tag{4.38}
$$

Therefore, by Lemma 4.2.8, it can be deduced that $\mathsf{d}([P,Q],[P',Q']) \leq \max\{\mathsf{d}(P,P'),\mathsf{d}(Q,Q')\}$, which concludes the proof of theorem Lemma 4.2.15.

Now, it is finally possible to adress the proof of the metric equation for the conditional statement as a whole. Considering the the semantics of the conditional statement in Figure 9, the rule for the conditional statement in Figure 10 is valid is equivalent to demonstrating that the distance between the evalution of a boolen $B$ followed by the execution of a program $P$ or a program $Q$ and the evalution of a boolean $B'$ followed by the execution of a program $P'$ or a program $Q'$ is less or equal to the distance between the evaluation of the boolean $B$ and the evaluation of the boolean $B'$ plus the maximum distance between the execution of the programs $P$ and $P'$ and the execution of the programs $Q$ and $Q'$, *ergo*, that for any booleand $B$ and $B'$ super-operators $P$, $Q$, $P'$ and $Q'$, the following holds:

**Theorem 4.2.4.** *$d(B\cdot[P,Q], B'\cdot[P',Q']) \leq d(B,B') + max\{d(P,P'), d(Q,Q')\}$*

*Proof.* Considering that in the quantum paradigm, the distance between two super-operators corresponds to the diamond norm between the two super-operators, it follows that:

$$
\begin{aligned}
&\mathsf{d}(B \cdot [P, Q], B' \cdot [P', Q']) \\
&= \|B \cdot [P, Q] \otimes I - B' \cdot [P', Q'] \otimes I\|_\sigma \\
&\leq \|(B - B') \otimes I\|_\sigma + \|([P, Q] - [P', Q']) \otimes I\|_\sigma && \{\text{Lemma 4.2.14}\} \\
&= d(B, B') + \|[P, Q] \otimes I - [P', Q'] \otimes I\|_\sigma \\
&= \mathsf{d}(B, B') + \mathsf{d}([P, Q], [P', Q']) \\
&= d(B, B') + \max\{\mathsf{d}(P, P'), \mathsf{d}(Q, Q')\} && \{\text{Lemma 4.2.15}\}
\end{aligned}
\tag{4.39}
$$

### 4.2.5 Quantum Teleportation

Bennett et al. [1993] introduced the concept of quantum teleportation, which is a protocol that allows the transfer of unknown quantum states between distant parties. The quantum teleportation protocol is a fundamental building block for quantum communication, quantum computation, and quantum networks, its applications ranging from secure quantum communication to distributed quantum computing Briegel et al. [1998]; Gottesman and Chuang [1999]; Kimble [2008].

The circuit corresponding to the implementation of the quantum teleportation protocol is depicted in Figure 11.

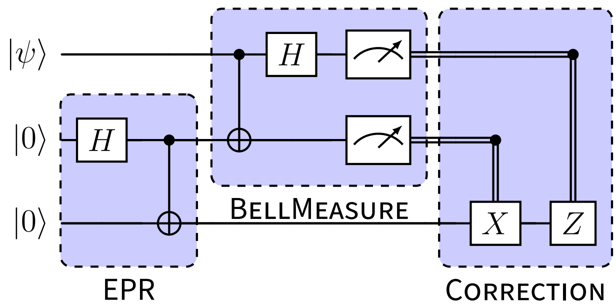


Figure 11: Quantum Teleportation Protocol

When formalizing the quantum teleportation protocol within the lambda calculus framework, each part of the protocol is instantiated as a distinct function. This entails the definition of three specific functions:

$$\textbf{EPR} : \ \mathbb{I} \multimap (\textit{qbit} \otimes \textit{qbit})$$
$$\textbf{BellMeasure} : \ \textit{qbit} \otimes \textit{qbit} \multimap \textit{bit} \otimes \textit{bit}$$
$$\textbf{Correction} : \ \textit{qbit} \otimes \textit{bit} \otimes \textit{bit} \multimap \textit{qbit}$$

The only part that is not self- explanatory is EPR, an acronym derived from a famous article written in 1935 by Albert Einstein, Boris Podolsky, and Nathan Rosen , where these authors questioned the completeness of Quantum Mechanics Einstein et al. [1935].
Considering the unitary operations $H : \textit{qbit} \to \textit{qbit}$, $X : \textit{qbit} \to \textit{qbit}$, $Z : \textit{qbit} \to \textit{qbit}$, $I : \textit{qbit} \to \textit{qbit}$, and $\textit{CNOT} : \textit{qbit}, \textit{qbit} \to \textit{qbit} \otimes \textit{qbit}$ , these functions are defined as follows:

$$\begin{aligned}
\textbf{EPR} &= - \triangleright \textit{CNOT}\,(H\,(q\,(\textit{new}\,0\,(*))), (q\,(\textit{new}\,0\,(*))))\\
\textbf{BellMeasure} &= q_1 : \textit{qbit}, q_2 : \textit{qbit} \triangleright (\text{pm } \textit{CNOT}(q_1, q_2) \text{ to } x \otimes y.\, \textit{meas}(H(x)) \otimes \textit{meas}(y))\\
\textbf{Correction} &= q : \textit{qbit}, x : \text{bit}, y : \text{bit} \triangleright \text{ case } x \,\{\text{inl}(x_0) \Rightarrow (\text{cond}\, y\, \{\text{inl}(y_0) \Rightarrow I(q);\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{inr}(y_1) \Rightarrow X(q))\};\\
&\qquad\qquad\qquad\qquad \text{inr}(x_1) \Rightarrow (\text{cond}\, y\, \{\text{inl}(y_0) \Rightarrow Z(q);\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{inr}(y_1) \Rightarrow Z(X(q))\})\}
\end{aligned}$$

Designating the qubit to be teleported as $q_0$, one can conceptualize the teleportation procedure as follows:

$$\begin{aligned}
q_0 : \textit{qbit} \triangleright &\text{ pm } \textbf{EPR}\,(*) \text{ to } q_1 \otimes q_2.\\
&\text{ pm } \textbf{BellMeasure}\,(q_0, q_1) \text{ to } c_0 \otimes c_1.\\
&\text{ pm } \textbf{Correction}\,(q_2, c_0, c_1) \text{ to } q.\, q
\end{aligned}$$

Regarding the interpretation of the quantum teleportation protocol, considering $\rho = |\phi\rangle\langle\phi|$ as the state of the system before measurement, $|\phi\rangle$ is calculated as follows, where $|\psi\rangle$ is the state of the qubit to be teleported:

$$
\begin{aligned}
& |\psi\rangle \otimes |0\rangle \otimes |0\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle \otimes |0\rangle \\
\xmapsto{I\otimes H\otimes I} \quad & (\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \\
\xmapsto{I\otimes CNOT} \quad & (\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle) \\
\xmapsto{CNOT\otimes I} \quad & \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle) \\
\xmapsto{H\otimes I\otimes I} \quad & \frac{1}{2}(\alpha|000\rangle + \alpha|001\rangle + \alpha|011\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|110\rangle + \beta|101\rangle - \beta|001\rangle) \\
= \quad & \frac{1}{2}(|00\rangle \otimes (\alpha|0\rangle + \beta|1\rangle) + |01\rangle \otimes (\alpha|1\rangle + \beta|0\rangle) + |10\rangle \otimes (\alpha|0\rangle - \beta|1\rangle) \\
& + |11\rangle \otimes (\alpha|1\rangle - \beta|0\rangle)) \\
= \quad & |00\rangle \otimes |\psi\rangle + |01\rangle \otimes X|\psi\rangle + |10\rangle \otimes Z|\psi\rangle + |11\rangle \otimes XZ|\psi\rangle = |\phi\rangle
\end{aligned}
\tag{4.40}
$$

Regarding the remaining steps of the protocol,

$$
\begin{aligned}
|\phi\rangle\langle\phi| = \quad & \frac{1}{4}(|00\rangle\langle 00| \otimes |\psi\rangle\langle\psi| + |00\rangle\langle 01| \otimes |\psi\rangle\langle\psi|X + |00\rangle\langle 10| \otimes |\psi\rangle\langle\psi|Z \\
& + |00\rangle\langle 11| \otimes |\psi\rangle\langle\psi|ZX + X|01\rangle\langle 00| \otimes |\psi\rangle\langle\psi| + |01\rangle\langle 01| \otimes X|\psi\rangle\langle\psi|X \\
& + |01\rangle\langle 10| \otimes X|\psi\rangle\langle\psi|Z + |01\rangle\langle 11| \otimes X|\psi\rangle\langle\psi|ZX + |10\rangle\langle 00| \otimes Z|\psi\rangle\langle\psi| \\
& + |10\rangle\langle 01| \otimes Z|\psi\rangle\langle\psi|X + |10\rangle\langle 10| \otimes Z|\psi\rangle\langle\psi|Z + |10\rangle\langle 11| \otimes Z|\psi\rangle\langle\psi|ZX \\
& + |00\rangle\langle 11| \otimes |\psi\rangle\langle\psi|ZX + |01\rangle\langle 11| \otimes X|\psi\rangle\langle\psi|ZX + |10\rangle\langle 11| \otimes Z|\psi\rangle\langle\psi|ZX \\
& + |11\rangle\langle 11| \otimes ZX|\psi\rangle\langle\psi|ZX) \\
\xmapsto{\text{meas}\otimes\text{meas}\otimes I} \quad & \Big(\Big(\frac{1}{4}|\psi\rangle\langle\psi|, \frac{1}{4}X|\psi\rangle\langle\psi|X\Big), \Big(\frac{1}{4}Z|\psi\rangle\langle\psi|Z, \frac{1}{4}XZ|\psi\rangle\langle\psi|ZX\Big)\Big)
\end{aligned}
\tag{4.41}
$$

With respect to the final step of the protocol, attending to the interpretation of the conditional statement (Figure 9), the state of the system after the application of the correction function is given by:

$$
\begin{aligned}
& \frac{1}{4}|\psi\rangle\langle\psi| + \frac{1}{4}XX|\psi\rangle\langle\psi|XX + \frac{1}{4}ZZ|\psi\rangle\langle\psi|ZZ + \frac{1}{4}ZXXZ|\psi\rangle\langle\psi|ZXXZ \\
= & \frac{1}{4}\left(|\psi\rangle\langle\psi| + |\psi\rangle\langle\psi| + |\psi\rangle\langle\psi| + |\psi\rangle\langle\psi|\right) = |\psi\rangle\langle\psi|
\end{aligned}
\tag{4.42}
$$

### 4.2.6 Ilustration: Noisy Quantum Teleportation

**Noisy Quantum Teleportation: Decoherence**

Realistic quantum systems are never isolated, but are immersed in the surrounding environment and interact continuously with it Schlosshauer [2005]. Decoherence can be seen as the consequence of that 'openness' of quantum systems to their environments . To study decoherence in a quantum channel within the presented metric deductive system, one can consider the application of a dephasing channel in the quantum teleportation protocol with a certain probability $p$.

The Kraus operators of the dephasing channel with probability $p$ are expressed as:

$$D_0 = \frac{\sqrt{2-p}}{\sqrt{2}} I, D_1 = \frac{\sqrt{p}}{\sqrt{2}} Z \tag{4.43}$$

Considering a density operator $\rho = |\alpha|^2 |0\rangle\langle 0| + \alpha\beta^\dagger |0\rangle\langle 1| + \alpha^\dagger\beta |1\rangle\langle 0| + |\beta|^2 |1\rangle\langle 1|$, using these Kraus operators, it is possible to easily verify that after applying the dephasing channel with probability $p$, the resulting operator $\rho'$ is given by:

$$\rho' = A_0\rho A_0^\dagger + A_1\rho A_1^\dagger = |\alpha|^2|0\rangle\langle 0| + (1-p)\alpha\beta^\dagger|0\rangle\langle 1| + (1-p)\alpha^\dagger\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| \tag{4.44}$$

This shows that the dephasing channel with probability $p$ preserves the diagonal elements of the density matrix while attenuating the off-diagonal elements by a factor of $(1-p)$.

The circuit representing the introduction of decoherence after EPR is illustrated in Figure 12.

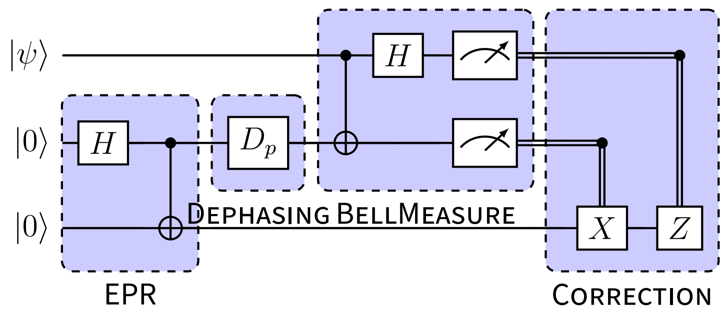


Figure 12: Quantum Teleportation Protocol: Dephasing with probability $p$ after EPR pair creation.

In this case, to facilate the analysis, the quantum teleportion protocol is devided in four parts: EPR, BellMeasure, Identity and Correction. This entails the definition of an aditional function and respective version subjected to decoherence with probability $p$:

$$\textbf{Identity} : \; \textit{qbit} \multimap \textit{qbit}$$
$$\textbf{Identity}^p : \; \textit{qbit} \multimap \textit{qbit}$$

Considering the unitary operation $I$ : *qbit* $\rightarrow$ *qbit*, and the operation $D_p$ : *qbit* $\rightarrow$ *qbit* the ideal version of this function, **Identity**, and its respective version subjected to decoherence with probability $p$, **Identity**$^p$, are defined as follows:

$$\textbf{Identity} = q : \textit{qbit} \triangleright I(q) : \textit{qbit} \tag{4.45}$$

$$\textbf{Identity}^p = q : \textit{qbit} \triangleright D_p(q) : \textit{qbit} \tag{4.46}$$

Designating the qubit to be teleported as $q_0$, one can conceptualize the teleportation procedure as follows:

$$\begin{aligned}
&\text{pm } \textbf{EPR}(*) \text{ to } q_1 \otimes q_2. \\
&\text{pm } \textbf{Identity}(q_1) \text{ to } id_q_1. \\
&\text{pm } \textbf{BellMeasure}(q_0, id_q_1) \text{ to } c_0 \otimes c_1. \\
&\text{pm } \textbf{Correction}(q_2, c_0, c_1) \text{ to } q.\, q
\end{aligned}$$

To evaluate the disparity between the ideal implementation of the quantum teleportation protocol and its realization subjected to decoherence, the initial step involves computing the distance between the density operators of the ideal and noisy implementations of the EPR state, denoted as $\rho$ and $\rho'$, respectively.

$$\begin{aligned}
& |0\rangle\langle 0| \otimes |0\rangle\langle 0| \\
\xmapsto{\text{EPR}} \; & \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|) = \rho \\
\xmapsto{D(p)\otimes I} \; & \frac{1}{2}(|00\rangle\langle 00| + (1-p)\,|00\rangle\langle 11| + (1-p)|11\rangle\langle 00| + |11\rangle\langle 11|) = \rho'
\end{aligned} \tag{4.47}$$

The distance between the $r$-image of the mapping $1 \mapsto \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)$ and the mapping $1 \mapsto \frac{1}{2}(|00\rangle\langle 00| + (1-p)\,|00\rangle\langle 11| + (1-p)|11\rangle\langle 00| + |11\rangle\langle 11|)$ is given by: $f(p) = \|\frac{p}{2}(|00\rangle\langle 11| + |11\rangle\langle 00|)\|_1$. Therefore, attending to Equation 3.9, $\|\rho -$

$\rho'(p)\|_\diamond = f(p)$.

$$
\begin{aligned}
f(p) &= \left\| \frac{p}{2} \left( |00\rangle \langle 11| + |11\rangle \langle 00| \right) \right\|_1 \\
&= \mathsf{Tr}\left( \sqrt{\frac{p^2}{4}(|00\rangle \langle 11| + |11\rangle \langle 00|)(|00\rangle \langle 11| + |11\rangle \langle 00|)^\dagger} \right) && \{\|\|_1 \text{ defn. for matrices}\} \\
&= \mathsf{Tr}\left( \sqrt{\frac{p^2}{4}(|00\rangle \langle 00| + |11\rangle \langle 11|)} \right) \\
&= \mathsf{Tr}\left( \frac{p}{2}(|00\rangle \langle 00| + |11\rangle \langle 11|) \right) && \{\text{Equation 3.5}\} \\
&= \frac{p}{2} + \frac{p}{2} = p
\end{aligned}
\tag{4.48}
$$

Therefore, the distance between the ideal and noisy implementations of the EPR state is given by $\|\rho - \rho'(p)\|_\diamond = p$.

Next, via the metric deductive system in Figure 3, it is easily verified that for an error $p$,

$$q : \textit{qbit} \triangleright I(q) =_p q : \textit{qbit} \triangleright D_p(q) : \textit{qbit} \tag{4.49}$$

Therefore **Identity** $=_p$ **Identity**$^p$ and finally, considering the entirety of the quantum teleportation protocol denoted as **QTP**, it follows that **QTP** $=_p$ **QTP**$^p$. This final metric equation indicates that by bounding the error associated with the application of decoherence with a specified probability $p$ to the initial qubit before measurement, it becomes feasible to limit the overall error of the entire quantum teleportation protocol. Moreover, it is interesting to observe that the error associated with the application of decoherence with a certain probability $p$ in one of the qubits corresponds exactly to that probability $p$.

#### Noisy Quantum Teleportation: Amplitude Damping

Next, the amplitude-damping channel is considered as a source of noise in the quantum teleportation protocol. Similarly to the dephasing channel, the amplitude damping channel serves as a model illustrating the dissipation of energy between a qubit and its environment. An example of this type of noise is found in the spontaneous emission of a photon by a two-level atom into an electromagnetic field environment with either a finite or infinite number of modes at zero temperature Salles et al. [2008]; Wang et al. [2011].

The amplitude damping channel with probability $\gamma$ is described by the Kraus operators:

$$A_0 = |0\rangle \langle 0| + \sqrt{1-\gamma}\, |1\rangle \langle 1|\,, A_1 = \sqrt{\gamma}\, |0\rangle \langle 1| \tag{4.50}$$

Applying these Kraus operators to the density operator $\rho = |\alpha|^2|0\rangle\langle 0|+\alpha\beta^\dagger|0\rangle\langle 1|+\alpha^\dagger\beta|1\rangle\langle 0|+|\beta|^2|1\rangle\langle 1|$, the the resulting operator $\rho'$ is given by:

$$\begin{aligned}\rho' &= A_0\rho A_0^\dagger + A_1\rho A_1^\dagger \\ &= (|\alpha|^2 + \gamma|\beta|^2)\,|0\rangle\langle 0| + \sqrt{1-\gamma}\,\alpha\beta^\dagger\,|0\rangle\langle 1| + \sqrt{1-\gamma}\,\alpha^\dagger\beta\,|1\rangle\langle 0| + (1-\gamma)|\beta|^2\,|1\rangle\langle 1|\end{aligned} \tag{4.51}$$

It is possible to observe that as $\gamma$ increases, while the $|1\rangle\langle 1|$ component, alongside the non-diagonal elements, are attenuated, the $|0\rangle\langle 0|$ element is amplified.

The circuit representing the introduction of amplitude damping after the correction step is presented in Figure 13.

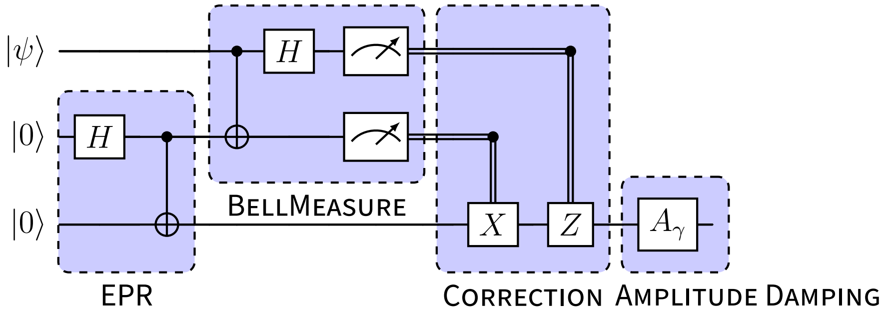


Figure 13: Quantum Teleportation Protocol: Amplitude Dampling with probability $\gamma$ after Correction.

Once again, a fourth part of the teleportation protocol, the **Identity**, is considered to facilitate the error analysis. In this case, it is neccessay to define the erroneous version of **Identity**, **Identity**$^{A(\gamma)}$:

$$\textbf{Identity}^{A(\gamma)} : \textit{qbit} \multimap \textit{qbit} \tag{4.52}$$

Considering the operation $A_\gamma$ : *qbit* $\rightarrow$ *qbit* the respective version of **Identity** subjected to amplitude damping with probability $\gamma$, **Identity**$^{A(\gamma)}$, is defined as follows:

$$\textbf{Identity}^{A(\gamma)} = q : \textit{qbit} \triangleright A_\gamma(q) : \textit{qbit} \tag{4.53}$$

Designating the qubit to be teleported as $q_0$, one can conceptualize the teleportation procedure as follows:

pm **EPR**$(*)$ to $q_1 \otimes q_2$.

pm **BellMeasure**$(q_0, q_1)$ to $c_0 \otimes c_1$.

pm **Correction**$(q_2, c_0, c_1)$ to $q$. **Identity**$(q)$

The first step to evaluate the distance between the ideal quantum teleportation protocol and the one subjected to amplitude damping with probability $\gamma$ is to compute the distance between the density operators of the ideal and noisy implementations of the teleported qubit, denoted as $\rho$ and $\rho'$, respectively.

As shown in Equation 4.42, the state of the teleported qubit is $\rho = |\psi\rangle\langle\psi|$. Given Equation 4.51, the state of the teleported qubit after amplitude damping with probability $\gamma$ is $(|\alpha|^2 + \gamma|\beta|^2)|0\rangle\langle 0| + \sqrt{1-\gamma}\alpha\beta^\dagger|0\rangle\langle 1| + \sqrt{1-\gamma}\alpha^\dagger\beta|1\rangle\langle 0| + (1-\gamma)|\beta|^2|1\rangle\langle 1|$, which is denoted as $\rho'$.

As a result,

$$
\begin{aligned}
\rho - \rho' =& |\alpha|^2|0\rangle\langle 0| + \alpha\beta^\dagger|0\rangle\langle 1| + \alpha^\dagger\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| - \big((|\alpha|^2 + \gamma|\beta|^2)|0\rangle\langle 0| \\
&+ \sqrt{1-\gamma}\alpha\beta^\dagger|0\rangle\langle 1| + \sqrt{1-\gamma}\alpha^\dagger\beta|1\rangle\langle 0| + (1-\gamma)|\beta|^2|1\rangle\langle 1|\big) \\
=& \gamma|\beta|^2|0\rangle\langle 0| + (1-\sqrt{1-\gamma})(\alpha\beta^\dagger|0\rangle\langle 1| + \alpha^\dagger\beta|1\rangle\langle 0|) - \gamma|\beta|^2|1\rangle\langle 1|
\end{aligned}
\tag{4.54}
$$

Employing Equation 3.6, the components of the Bloch vector of the state $\rho - \rho'$ are as follows:

$$
\begin{aligned}
r_x =& \mathrm{Tr}\left[\begin{pmatrix} \gamma|\beta|^2 & (1-\sqrt{1-\gamma})\alpha\beta^\dagger \\ (1-\sqrt{1-\gamma})\alpha^\dagger\beta & -\gamma|\beta|^2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right] \\
=& \mathrm{Tr}\left[\begin{pmatrix} (1-\sqrt{1-\gamma})\alpha\beta^\dagger & \gamma|\beta|^2 \\ -\gamma|\beta|^2 & (1-\sqrt{1-\gamma})\alpha^\dagger\beta \end{pmatrix}\right] = (1-\sqrt{1-\gamma})(\alpha\beta^\dagger + \alpha^\dagger\beta) \\
\\
r_y =& \mathrm{Tr}\left[\begin{pmatrix} \gamma|\beta|^2 & (1-\sqrt{1-\gamma})\alpha\beta^\dagger \\ (1-\sqrt{1-\gamma})\alpha^\dagger\beta & -\gamma|\beta|^2 \end{pmatrix} \cdot \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right] \\
=& \mathrm{Tr}\left[\begin{pmatrix} i(1-\sqrt{1-\gamma})\alpha\beta^\dagger & -i\gamma|\beta|^2 \\ i\gamma|\beta|^2 & -i(1-\sqrt{1-\gamma})\alpha^\dagger\beta \end{pmatrix}\right] = i(1-\sqrt{1-\gamma})(\alpha\beta^\dagger - \alpha^\dagger\beta) \\
\\
r_z =& \mathrm{Tr}\left[\begin{pmatrix} \gamma|\beta|^2 & (1-\sqrt{1-\gamma})\alpha\beta^\dagger \\ (1-\sqrt{1-\gamma})\alpha^\dagger\beta & -\gamma|\beta|^2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right] \\
=& \mathrm{Tr}\left[\begin{pmatrix} \gamma|\beta|^2 & -(1-\sqrt{1-\gamma})\alpha\beta^\dagger \\ (1-\sqrt{1-\gamma})\alpha^\dagger\beta & \gamma|\beta|^2 \end{pmatrix}\right] = \gamma|\beta|^2 + \gamma|\beta|^2 = 2\gamma|\beta|^2
\end{aligned}
\tag{4.55}
$$

Consequently, and knowing that the distance between two vectors corresponds to their Euclidean distance, it follows that the distance between the ideal and noisy implementations of the teleported qubit corresponds to:

$$
\begin{aligned}
&\|\rho - \rho'\|_{\diamond} \\
&= \left\| \left( (1-\sqrt{1-\gamma})(\alpha\beta^{\dagger} + \alpha^{\dagger}\beta), i(1-\sqrt{1-\gamma})(\alpha\beta^{\dagger} - \alpha^{\dagger}\beta), 2\gamma|\beta|^2 \right) \right\|_2 && \{\text{Equation 4.55}\} \\
&= \sqrt{\left( (1-\sqrt{1-\gamma})(\alpha\beta^{\dagger} + \alpha^{\dagger}\beta) \right)^2 + \left( i(1-\sqrt{1-\gamma})(\alpha\beta^{\dagger} - \alpha^{\dagger}\beta) \right)^2 + (2\gamma|\beta|^2)^2} && \{\text{Definition 3.1.21}\} \\
&= \sqrt{\left( (1-\sqrt{1-\gamma})(\alpha\beta^{\dagger} + \alpha^{\dagger}\beta) \right)^2 - \left( (1-\sqrt{1-\gamma})(\alpha\beta^{\dagger} - \alpha^{\dagger}\beta) \right)^2 + (2\gamma|\beta|^2)^2} \\
&= \sqrt{4 \cdot (1-\sqrt{1-\gamma})^2|\alpha|^2|\beta|^2 + 4\gamma^2|\beta|^4} \\
&= 2 \cdot \sqrt{(1-\sqrt{1-\gamma})^2|\alpha|^2|\beta|^2 + \gamma^2|\beta|^4}
\end{aligned}
\tag{4.56}
$$

Note that, as expected when $\gamma \to 0$ or $\beta \to 0$ ,$\|\rho - \rho'\|_{\diamond} \to 0$, and when $\gamma \to 1$, $\|\rho - \rho'\|_{\diamond} \to 2\left(\sqrt{|\alpha|^2|\beta|^2 + \gamma^2|\beta|^4}\right)$.

From this result, it follows that **Identity** $=_{2\cdot\sqrt{(1-\sqrt{1-\gamma})^2|\alpha|^2|\beta|^2+\gamma^2|\beta|^4}}$ **Identity**$^{A(\gamma)}$. Thus, **QTP** $=_{2\cdot\sqrt{(1-\sqrt{1-\gamma})^2|\alpha|^2|\beta|^2+\gamma^2|\beta|^4}}$ **QTP**$^{A(\gamma)}$.

#### Noisy Quantum Teleportation: An imperfect implementation of the Hadamard gate

Now, it will be considered an imperfect implementation of a Hadamard gate, denoted as $H^{\epsilon}$. Therefore, a new operation is added $H^{\epsilon}$ : *qbit* $\to$ *qbit* and it is postulated as an axiom that $q$ : *qbit* $\triangleright H =_{\epsilon} H^{\epsilon}$ : *qbit*. In this example, considering the Hadamard gate as the composition $R_y(\frac{\pi}{2}) \cdot P(\pi)$, $H^{\epsilon}$ is regarded as the composition $R_y(\frac{\pi}{2}) \cdot P(\pi + \delta)$. This imperfect implementation deviates from a precise rotation of $\pi$ radians along the $z$-axis, rotating by $\pi + \delta$ radians instead. This type of imperfection is inevitable during the implementation of quantum gates. The circuit representing the introduction of an erroneous Hadamard gate is presented in Figure 14.

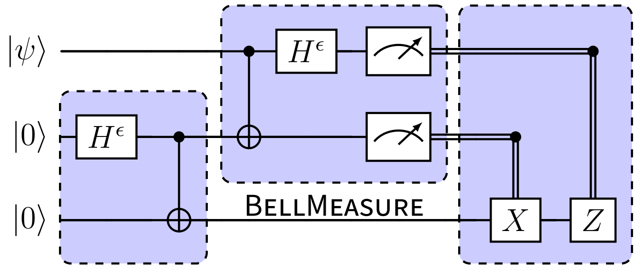


Figure 14: Quantum Teleportation Protocol: Erroneous implementation of the Hadamard gate. $H^\epsilon$ is regarded as the composition $R_y(\frac{\pi}{2}) \cdot P(\pi + \epsilon)$.

As usual, the initial step consists of evaluating the distance between the density operators of the ideal and noisy implementations of the Hadamard gate within each block. With respect to the EPR block, as presented in Equation 4.40 the ideal state of the EPR pair is $\frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)$. Regarding, the imperfect Hadamard gate one has that:

$$\begin{aligned} & |0\rangle \otimes |0\rangle \\ \xmapsto{H^\epsilon \otimes I} \quad & R_y\left(\frac{\pi}{2}\right) \cdot P(\pi+\epsilon)\,|0\rangle \otimes |0\rangle = R_y\left(\frac{\pi}{2}\right)|0\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \\ \xmapsto{CNOT} \quad & \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi\rangle \end{aligned} \tag{4.57}$$

Therefore, the state of the EPR pair with an imperfect Hadamard gate is $|\Phi\rangle\langle\Phi| = \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)$. Hence, the imperfect Hadamard gate does not affect the state of the EPR pair and, as a result, the distance between the ideal and noisy implementations of the EPR pair is zero, **EPR** $=_0$ **EPR**$^{H(\epsilon)}$.

Next, it is necessary to repeat this exercise regarding the BellMeasure block. As shown in Equation 4.41, the ideal state of the BellMeasure block is $\rho = \left(\left(\frac{1}{4}|\psi\rangle\langle\psi|, \frac{1}{4}X|\psi\rangle\langle\psi|X\right), \left(\frac{1}{4}Z|\psi\rangle\langle\psi|Z, \frac{1}{4}XZ|\psi\rangle\langle\psi|ZX\right)\right)$. Regarding the imperfect Hadamard gate, knowing that:

$$\begin{aligned} & \alpha|0\rangle + \beta|1\rangle \\ \xmapsto{H^\epsilon} \quad & R_y\left(\frac{\pi}{2}\right) \cdot P(\pi+\epsilon)(\alpha|0\rangle + \beta|1\rangle) = R_y\left(\frac{\pi}{2}\right) \cdot \left(\alpha|0\rangle + e^{i(\pi+\epsilon)}\beta|1\rangle\right) \\ = \quad & R_y\left(\frac{\pi}{2}\right) \cdot \left(\alpha|0\rangle - e^{i\epsilon}\beta|1\rangle\right) = \frac{1}{\sqrt{2}}\left((\alpha + e^{i\epsilon}\beta)|0\rangle + (\alpha - e^{\epsilon}\beta)|1\rangle\right) \end{aligned} \tag{4.58}$$

It follows, that:

$$
\begin{aligned}
&|\psi\rangle \otimes |0\rangle \otimes |0\rangle \\
\xmapsto{\text{EPR}}\ & \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle) \\
\xmapsto{CNOT\otimes I}\ & \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle) \\
\xmapsto{H^{\epsilon}\otimes I\otimes I}\ & \frac{1}{2}\left(\alpha(|000\rangle + |100\rangle + |011\rangle + |111\rangle) + \beta e^{i\epsilon}(|010\rangle - |110\rangle + |001\rangle - |101\rangle)\right) \\
&= \frac{1}{2}\left(\alpha(|000\rangle + |100\rangle + |011\rangle + |111\rangle) + \beta e^{i\epsilon}(|010\rangle - |110\rangle + |001\rangle - |101\rangle)\right) \\
&= \frac{1}{2}(|00\rangle \otimes (\alpha|0\rangle + \beta e^{i\epsilon}|1\rangle) + |01\rangle \otimes (\alpha|1\rangle + e^{i\epsilon}\beta|0\rangle) + |10\rangle \otimes (\alpha|0\rangle - e^{i\epsilon}\beta|1\rangle)) \\
&+ |11\rangle \otimes (\alpha|1\rangle - e^{i\epsilon}\beta|0\rangle)) \\
&= |00\rangle \otimes P(\epsilon)|\psi\rangle + |01\rangle \otimes XP(\epsilon)|\psi\rangle + |10\rangle \otimes ZP(\epsilon)|\psi\rangle + |11\rangle \otimes XZP(\epsilon)|\psi\rangle \\
&= |\phi'\rangle
\end{aligned}
\tag{4.59}
$$

Finally, measuring the first two qubits:

$$
\begin{aligned}
|\phi'\rangle\langle\phi'| \xmapsto{\text{meas}\otimes\text{meas}\otimes I}\ & \Big(\Big(\frac{1}{4}P(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon), \frac{1}{4}XP(\epsilon)|\psi\rangle\langle\psi|XP^{\dagger}(\epsilon)\Big), \\
& \Big(\frac{1}{4}ZP(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon)Z, \frac{1}{4}XZP(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon)ZX\Big)\Big) = \rho'
\end{aligned}
\tag{4.60}
$$

Given that,

$$
\begin{aligned}
|\psi\rangle\langle\psi| - P(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon) =& |\alpha|^2|0\rangle\langle 0| + \alpha\beta^{\dagger}|0\rangle\langle 1| + \alpha^{\dagger}\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| - \\
& (|\alpha|^2|0\rangle\langle 0| + e^{-i\epsilon}\alpha\beta^{\dagger}|0\rangle\langle 1| + e^{i\epsilon}\alpha^{\dagger}\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|) \\
=& (1 - e^{-i\epsilon})\alpha\beta^{\dagger}|0\rangle\langle 1| + (1 - e^{i\epsilon})\alpha^{\dagger}\beta|1\rangle\langle 0|
\end{aligned}
\tag{4.61}
$$

$$
\begin{aligned}
X|\psi\rangle\langle\psi|X - XP(\epsilon)|\psi\rangle\langle\psi|XP^{\dagger}(\epsilon) =& |\alpha|^2|1\rangle\langle 1| + \alpha\beta^{\dagger}|1\rangle\langle 0| + \alpha^{\dagger}\beta|0\rangle\langle 1| + |\beta|^2|0\rangle\langle 0| - \\
& (|\alpha|^2|1\rangle\langle 1| + e^{-i\epsilon}\alpha\beta^{\dagger}|1\rangle\langle 0| + e^{i\epsilon}\alpha^{\dagger}\beta|0\rangle\langle 1| + |\beta|^2|0\rangle\langle 0|) \\
=& (1 - e^{-i\epsilon})\alpha\beta^{\dagger}|1\rangle\langle 0| + (1 - e^{i\epsilon})\alpha^{\dagger}\beta|0\rangle\langle 1|
\end{aligned}
\tag{4.62}
$$

$$
\begin{aligned}
Z|\psi\rangle\langle\psi|Z - ZP(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon)Z =& |\alpha|^2|0\rangle\langle 0| - \alpha\beta^{\dagger}|0\rangle\langle 1| - \alpha^{\dagger}\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| - \\
& (|\alpha|^2|0\rangle\langle 0| - e^{-i\epsilon}\alpha\beta^{\dagger}|0\rangle\langle 1| - e^{i\epsilon}\alpha^{\dagger}\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|) \\
=& (e^{-i\epsilon} - 1)\alpha\beta^{\dagger}|0\rangle\langle 1| + (e^{i\epsilon} - 1)\alpha^{\dagger}\beta|1\rangle\langle 0|
\end{aligned}
$$

(4.63)

$$XZ\,|\psi\rangle\,\langle\psi|\,ZX - XZP(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon)ZX = |\alpha|^2|1\rangle\langle 1| - \alpha\beta^{\dagger}|1\rangle\langle 0| - \alpha^{\dagger}\beta|0\rangle\langle 1| + |\beta|^2|0\rangle\langle 0| - (|\alpha|^2|1\rangle\langle 1| - e^{-i\epsilon}\alpha\beta^{\dagger}|1\rangle\langle 0| - e^{i\epsilon}\alpha^{\dagger}\beta|0\rangle\langle 1| + |\beta|^2|0\rangle\langle 0|) = (e^{-i\epsilon}-1)\alpha\beta^{\dagger}|1\rangle\langle 0| + (e^{i\epsilon}-1)\alpha^{\dagger}\beta|0\rangle\langle 1| \tag{4.64}$$

Consequently,

$$\begin{aligned}\rho-\rho' =&\Big(\Big(\frac{1}{4}|\psi\rangle\langle\psi| - \frac{1}{4}P(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon), \frac{1}{4}XP(\epsilon)|\psi\rangle\langle\psi|XP^{\dagger}(\epsilon) - \frac{1}{4}XP(\epsilon)|\psi\rangle\langle\psi|XP^{\dagger}(\epsilon)\Big),\\ &\Big(\frac{1}{4}ZP(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon)Z - \frac{1}{4}ZP(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon)Z,\\ &\frac{1}{4}P(\epsilon)XZ|\psi\rangle\langle\psi|P^{\dagger}(\epsilon)ZX - \frac{1}{4}XZP(\epsilon)|\psi\rangle\langle\psi|P^{\dagger}(\epsilon)ZX\Big)\Big)\\ =&\Big(\Big(\frac{1}{4}(1-e^{-i\epsilon})\alpha\beta^{\dagger}|0\rangle\langle 1| + \frac{1}{4}(1-e^{i\epsilon})\alpha^{\dagger}\beta|1\rangle\langle 0|, \frac{1}{4}(1-e^{-i\epsilon})\alpha\beta^{\dagger}|1\rangle\langle 0| + \frac{1}{4}(1-e^{i\epsilon})\alpha^{\dagger}\beta|0\rangle\langle 1|\Big),\\ &\Big(\frac{1}{4}(e^{-i\epsilon}-1)\alpha\beta^{\dagger}|0\rangle\langle 1| + \frac{1}{4}(e^{i\epsilon}-1)\alpha^{\dagger}\beta|1\rangle\langle 0|, \frac{1}{4}(e^{-i\epsilon}-1)\alpha\beta^{\dagger}|1\rangle\langle 0| + \frac{1}{4}(e^{i\epsilon}-1)\alpha^{\dagger}\beta|0\rangle\langle 1||\Big)\Big)\\ =&\left(\left(\frac{1}{4}\sigma, \frac{1}{4}\sigma'\right), \left(\frac{1}{4}\sigma'', \frac{1}{4}\sigma'''\right)\right)\end{aligned} \tag{4.65}$$

Employing Equation 3.6, the components of the Bloch vector of each state $\sigma, \sigma', \sigma'', \sigma'''$ are as follows:

$$
\begin{aligned}
r_x =& \mathsf{Tr}\left[\begin{pmatrix} 0 & (1-e^{-i\epsilon})\alpha\beta^\dagger \\ (1-e^{i\epsilon})\alpha^\dagger\beta & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right] \\
=& \mathsf{Tr}\left[\begin{pmatrix} (1-e^{-i\epsilon})\alpha\beta^\dagger & 0 \\ 0 & (1-e^{i\epsilon})\alpha^\dagger\beta \end{pmatrix}\right] = (1-e^{-i\epsilon})\alpha\beta^\dagger + (1-e^{i\epsilon})\alpha^\dagger\beta \\
r_y =& \mathsf{Tr}\left[\begin{pmatrix} 0 & (1-e^{-i\epsilon})\alpha\beta^\dagger \\ (1-e^{i\epsilon})\alpha^\dagger\beta & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right] \\
=& \mathsf{Tr}\left[\begin{pmatrix} i(1-e^{-i\epsilon})\alpha\beta^\dagger & 0 \\ 0 & -i(1-e^{i\epsilon})\alpha^\dagger\beta \end{pmatrix}\right] = i(1-e^{-i\epsilon})\alpha\beta^\dagger - i(1-e^{i\epsilon})\alpha^\dagger\beta \\
r_z =& \mathsf{Tr}\left[\begin{pmatrix} 0 & (1-e^{-i\epsilon})\alpha\beta^\dagger \\ (1-e^{i\epsilon})\alpha^\dagger\beta & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right] \\
=& \mathsf{Tr}\left[\begin{pmatrix} 0 & (1-e^{-i\epsilon})\alpha\beta^\dagger \\ -(1-e^{i\epsilon})\alpha^\dagger\beta & 0 \end{pmatrix}\right] = 0
\end{aligned} \tag{4.66}
$$

$$
\begin{aligned}
r_x =& \mathsf{Tr}\left[\begin{pmatrix} 0 & (1-e^{i\epsilon})\alpha^\dagger\beta \\ (1-e^{-i\epsilon})\alpha\beta^\dagger & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right] \\
=& \mathsf{Tr}\left[\begin{pmatrix} (1-e^{i\epsilon})\alpha^\dagger\beta & 0 \\ 0 & (1-e^{-i\epsilon})\alpha\beta^\dagger \end{pmatrix}\right] = (1-e^{i\epsilon})\alpha^\dagger\beta + (1-e^{-i\epsilon})\alpha\beta^\dagger \\
r_y =& \mathsf{Tr}\left[\begin{pmatrix} 0 & (1-e^{i\epsilon})\alpha^\dagger\beta \\ (1-e^{-i\epsilon})\alpha\beta^\dagger & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right] \\
=& \mathsf{Tr}\left[\begin{pmatrix} i(1-e^{i\epsilon})\alpha^\dagger\beta & 0 \\ 0 & -i(1-e^{-i\epsilon})\alpha\beta^\dagger \end{pmatrix}\right] = i(1-e^{i\epsilon})\alpha^\dagger\beta - i(1-e^{-i\epsilon})\alpha\beta^\dagger \\
r_z =& \mathsf{Tr}\left[\begin{pmatrix} 0 & (1-e^{i\epsilon})\alpha^\dagger\beta \\ (1-e^{-i\epsilon})\alpha\beta^\dagger & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right] \\
=& \mathsf{Tr}\left[\begin{pmatrix} 0 & (1-e^{i\epsilon})\alpha^\dagger\beta \\ -(1-e^{-i\epsilon})\alpha\beta^\dagger & 0 \end{pmatrix}\right] = 0
\end{aligned} \tag{4.67}
$$

$$
\begin{aligned}
r_x =&\mathsf{Tr}\left[\begin{pmatrix} 0 & (e^{-i\epsilon}-1)\alpha\beta^\dagger \\ (e^{i\epsilon}-1)\alpha^\dagger\beta & 0 \end{pmatrix}\cdot\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right] \\
=&\mathsf{Tr}\left[\begin{pmatrix} (e^{-i\epsilon}-1)\alpha\beta^\dagger & 0 \\ 0 & (e^{i\epsilon}-1)\alpha^\dagger\beta \end{pmatrix}\right] = (e^{-i\epsilon}-1)\alpha\beta^\dagger + (e^{i\epsilon}-1)\alpha^\dagger\beta \\
r_y =&\mathsf{Tr}\left[\begin{pmatrix} 0 & (e^{-i\epsilon}-1)\alpha\beta^\dagger \\ (e^{i\epsilon}-1)\alpha^\dagger\beta & 0 \end{pmatrix}\cdot\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right] \\
=&\mathsf{Tr}\left[\begin{pmatrix} i(e^{-i\epsilon}-1)\alpha\beta^\dagger & 0 \\ 0 & -i(e^{i\epsilon}-1)\alpha^\dagger\beta \end{pmatrix}\right] = i(e^{-i\epsilon}-1)\alpha\beta^\dagger - i(e^{i\epsilon}-1)\alpha^\dagger\beta \\
r_z =&\mathsf{Tr}\left[\begin{pmatrix} 0 & (e^{-i\epsilon}-1)\alpha\beta^\dagger \\ (e^{i\epsilon}-1)\alpha^\dagger\beta & 0 \end{pmatrix}\cdot\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right] \\
=&\mathsf{Tr}\left[\begin{pmatrix} 0 & (e^{-i\epsilon}-1)\alpha\beta^\dagger \\ -(e^{i\epsilon}-1)\alpha^\dagger\beta & 0 \end{pmatrix}\right] = 0
\end{aligned}
\tag{4.68}
$$

$$
\begin{aligned}
r_x =&\mathsf{Tr}\left[\begin{pmatrix} 0 & (e^{i\epsilon}-1)\alpha^\dagger\beta \\ (e^{-i\epsilon}-1)\alpha\beta^\dagger & 0 \end{pmatrix}\cdot\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right] \\
=&\mathsf{Tr}\left[\begin{pmatrix} (e^{i\epsilon}-1)\alpha^\dagger\beta & 0 \\ 0 & (e^{-i\epsilon}-1)\alpha\beta^\dagger \end{pmatrix}\right] = (e^{i\epsilon}-1)\alpha^\dagger\beta + (e^{-i\epsilon}-1)\alpha\beta^\dagger \\
r_y =&\mathsf{Tr}\left[\begin{pmatrix} 0 & (e^{i\epsilon}-1)\alpha^\dagger\beta \\ (e^{-i\epsilon}-1)\alpha\beta^\dagger & 0 \end{pmatrix}\cdot\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right] \\
=&\mathsf{Tr}\left[\begin{pmatrix} i(e^{i\epsilon}-1)\alpha^\dagger\beta & 0 \\ 0 & -i(e^{-i\epsilon}-1)\alpha\beta^\dagger \end{pmatrix}\right] = i(e^{i\epsilon}-1)\alpha^\dagger\beta - i(e^{-i\epsilon}-1)\alpha\beta^\dagger \\
r_z =&\mathsf{Tr}\left[\begin{pmatrix} 0 & (e^{i\epsilon}-1)\alpha^\dagger\beta \\ (e^{-i\epsilon}-1)\alpha\beta^\dagger & 0 \end{pmatrix}\cdot\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right] \\
=&\mathsf{Tr}\left[\begin{pmatrix} 0 & (e^{i\epsilon}-1)\alpha^\dagger\beta \\ -(e^{-i\epsilon}-1)\alpha\beta^\dagger & 0 \end{pmatrix}\right] = 0
\end{aligned}
\tag{4.69}
$$

As a result, and given that the distance between two vectors corresponds to their Euclidean distance, it follows that the distance between the ideal and noisy implementations of the

Hadamard gate in the BellMeasure block corresponds to:

$$\begin{aligned}
\|\rho-\rho'\|_\diamond &= \left\|\left(\left(\frac{1}{4}\sigma, \frac{1}{4}\sigma'\right), \left(\frac{1}{4}\sigma'', \frac{1}{4}\sigma'''\right)\right)\right\|_\diamond \\
&= \left\|\frac{1}{4}\sigma\right\|_\diamond + \left\|\frac{1}{4}\sigma'\right\|_\diamond + \left\|\frac{1}{4}\sigma''\right\|_\diamond + \left\|\frac{1}{4}\sigma'''\right\|_\diamond \qquad \{\text{Equation 4.8}\} \\
&= \left\|\frac{1}{4}((1-e^{-i\epsilon})\alpha\beta^\dagger + (1-e^{i\epsilon})\alpha^\dagger\beta, i(1-e^{-i\epsilon})\alpha\beta^\dagger - i(1-e^{i\epsilon})\alpha^\dagger\beta)\right\|_2 + \\
&\quad \left\|\frac{1}{4}((1-e^{i\epsilon})\alpha^\dagger\beta + (1-e^{-i\epsilon})\alpha\beta^\dagger, i(1-e^{i\epsilon})\alpha^\dagger\beta - i(1-e^{-i\epsilon})\alpha\beta^\dagger)\right\|_2 + \\
&\quad \left\|\frac{1}{4}((e^{-i\epsilon}-1)\alpha\beta^\dagger + (e^{i\epsilon}-1)\alpha^\dagger\beta, i(e^{-i\epsilon}-1)\alpha\beta^\dagger - i(e^{i\epsilon}-1)\alpha^\dagger\beta)\right\|_2 + \\
&\quad \left\|\frac{1}{4}((e^{i\epsilon}-1)\alpha^\dagger\beta + (e^{-i\epsilon}-1)\alpha\beta^\dagger, i(e^{i\epsilon}-1)\alpha^\dagger\beta - i(e^{-i\epsilon}-1)\alpha\beta^\dagger)\right\|_2
\end{aligned} \tag{4.70}$$

Applying Definition 3.1.21 to each term, it follows that:

$$\begin{aligned}
\|\rho-\rho'\|_\diamond =& \frac{1}{4}\sqrt{((1-e^{-i\epsilon})\alpha\beta^\dagger + (1-e^{i\epsilon})\alpha^\dagger\beta)^2 + (i((1-e^{-i\epsilon})\alpha\beta^\dagger - (1-e^{i\epsilon})\alpha^\dagger\beta))^2} \\
& + \frac{1}{4}\sqrt{((1-e^{i\epsilon})\alpha^\dagger\beta + (1-e^{-i\epsilon})\alpha\beta^\dagger)^2 + (i((1-e^{i\epsilon})\alpha^\dagger\beta - (1-e^{-i\epsilon})\alpha\beta^\dagger))^2} \\
& + \frac{1}{4}\sqrt{((e^{-i\epsilon}-1)\alpha\beta^\dagger + (e^{i\epsilon}-1)\alpha^\dagger\beta)^2 + (i((e^{-i\epsilon}-1)\alpha\beta^\dagger - (e^{i\epsilon}-1)\alpha^\dagger\beta))^2} \\
& + \frac{1}{4}\sqrt{((e^{i\epsilon}-1)\alpha^\dagger\beta + (e^{-i\epsilon}-1)\alpha\beta^\dagger)^2 + (i((e^{i\epsilon}-1)\alpha^\dagger\beta - (e^{-i\epsilon}-1)\alpha\beta^\dagger))^2} \\
=& \frac{1}{4}\sqrt{((1-e^{-i\epsilon})\alpha\beta^\dagger + (1-e^{i\epsilon})\alpha^\dagger\beta)^2 + (i((1-e^{-i\epsilon})\alpha\beta^\dagger - (1-e^{i\epsilon})\alpha^\dagger\beta))^2} \\
& + \frac{1}{4}\sqrt{((1-e^{-i\epsilon})\alpha\beta^\dagger + (1-e^{i\epsilon})\alpha^\dagger\beta)^2 + (-(i((1-e^{-i\epsilon})\alpha\beta^\dagger - (1-e^{i\epsilon})\alpha^\dagger\beta)))^2} \\
& + \frac{1}{4}\sqrt{(-((1-e^{-i\epsilon})\alpha\beta^\dagger + (1-e^{i\epsilon})\alpha^\dagger\beta))^2 + (i((1-e^{-i\epsilon})\alpha\beta^\dagger - (1-e^{i\epsilon})\alpha^\dagger\beta))^2} \\
& + \frac{1}{4}\sqrt{(-((1-e^{-i\epsilon})\alpha\beta^\dagger + (1-e^{i\epsilon})\alpha^\dagger\beta))^2 + (-(i((1-e^{-i\epsilon})\alpha\beta^\dagger - (1-e^{i\epsilon})\alpha^\dagger\beta)))^2} \\
& = \frac{1}{4}\sqrt{((1-e^{-i\epsilon})\alpha\beta^\dagger + (1-e^{i\epsilon})\alpha^\dagger\beta)^2 + (i((1-e^{-i\epsilon})\alpha\beta^\dagger - (1-e^{i\epsilon})\alpha^\dagger\beta))^2} \\
& = \sqrt{4(1-e^{-i\epsilon})(1-e^{i\epsilon})|\alpha|^2|\beta|^2} = 2\sqrt{(1-e^{i\epsilon}-e^{-i\epsilon}+1)|\alpha|^2|\beta|^2} \\
& = 2\sqrt{2(1-\cos(\epsilon))|\alpha|^2|\beta|^2} = 2\sqrt{2}\sqrt{(1-\cos(\epsilon))|\alpha|^2|\beta|^2}
\end{aligned} \tag{4.71}$$

It is possible to observe that when $\epsilon = 0$, the distance between the ideal and noisy implementations of the Hadamard gate in the BellMeasure block is zero, which is consistent with the fact that the ideal and noisy implementations are the same. The same goes for $\epsilon = \pi$,

$\alpha = 0$ and $\beta = 0$ given that only the non-diagonal components of the density matrix are affected by an erroneous phase gate.
Given this result it is possible to conclude that **BellMeasure** $=_{2\sqrt{2}\sqrt{(1-\cos(\epsilon))|\alpha|^2|\beta|^2}}$ **BellMeasure**$^{H(\epsilon)}$.
Hence, **QTP** $=_{0+2\sqrt{2}\sqrt{(1-\cos(\epsilon))|\alpha|^2|\beta|^2}}$ **QTP**$^{H(\epsilon)}$, *i.e.*, **QTP** $=_{2\sqrt{2}\sqrt{(1-\cos(\epsilon))|\alpha|^2|\beta|^2}}$ **QTP**$^{H(\epsilon)}$ .

## 4.3 Discard Operation

The discard operation was defined as the trace, and therefore is also completely positive and trace-preserving.

### 4.3.1 Example: Proving an equivalence using the discard equation-in-context

This subsection aims to ilustrate how to prove that

$$- \triangleright \mathrm{disc}\,(q\,(\mathit{new}0\,(*)))\,\mathrm{to}\,*\,.\,q\,(\mathit{new}0\,(*))\,:\,\mathit{qbit} = - \triangleright q\,(\mathit{new}0\,(*))\,:\,\mathit{qbit} \tag{4.72}$$

using the discard equation-in-context.
The discard equation in the bottom line in Figure 2 states that all judgements $\Gamma \triangleright v : \mathbb{I}$ (with $\Gamma = x_1 : \mathbb{A}_1, ..., x_n : \mathbb{A}_n$) carry no different information than that of just discarding all variables available in context $\Gamma$. Therefore considering an empty context $\Gamma = -$, the discard equation states that:

$$- \triangleright v : \mathbb{I} = - \triangleright * : \mathbb{I} \tag{4.73}$$

Given that as established in Figure 1, $\mathrm{dis}(v) : \mathbb{I}$, it follows that:

$$- \triangleright \mathrm{disc}\,(q\,(\mathit{new}0\,(*)))\,\mathrm{to}\,*\,.\,q\,(\mathit{new}0\,(*))\,:\,\mathit{qbit} = - \triangleright *\,\mathrm{to}\,*\,.\,q\,(\mathit{new}0\,(*))\,:\,\mathit{qbit} \tag{4.74}$$

Subsequently applying the rule $*$ to $*\,.\,v = v$ in Figure 2, it holds that

$$\begin{aligned} - \triangleright \mathrm{disc}\,(q\,(\mathit{new}0\,(*)))\,\mathrm{to}\,*\,.\,q\,(\mathit{new}0\,(*))\,:\,\mathit{qbit} &= - \triangleright *\ \mathrm{to}\,*\,.\,q\,(\mathit{new}0\,(*))\,:\,\mathit{qbit} \\ &= - \triangleright q\,(\mathit{new}0\,(*))\,:\,\mathit{qbit} \end{aligned} \tag{4.75}$$

### 4.3.2 Ilustration: A malicious attack on the quantum teleportation protocol

Now, consider a malicious attack on the quantum teleportation protocol in the form of a bit-flip occurring with a 50% probability before measurement. More generally, one can define an operation $T$ that applies a unitary operation $U$ to the state given as input with 50% probability. Operation $T$ can be defined as follows:

$$T: \ \ \mathit{qbit}, ..., \mathit{qbit} \multimap \mathit{qbit}^{\otimes n}$$
$$T = q_1 : \mathit{qbit}, ..., q_n : \mathit{qbit} \triangleright \mathsf{pm}\ CU(R_X^{\frac{\pi}{2}}(q(\mathit{new}\,0(*))), q_1, ..., q_n)\ \mathsf{to}\ newq \otimes q.\, \mathit{disc}(newq)$$

This operation is depicted in Figure 15.

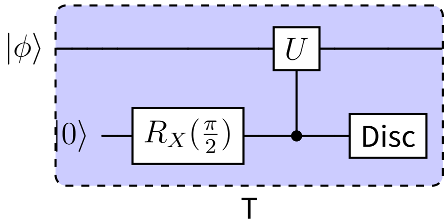


Figure 15: T operation

Regarding the calculations, applying operation $T$ to the state $|\psi\rangle$, one has that:

$$\begin{aligned}
& |\phi\rangle\langle\phi| \\
\xmapsto{I \otimes q(\mathit{new}\,0(*))} \quad & |\phi\rangle\langle\phi| \otimes |0\rangle\langle 0| \\
\xmapsto{I \otimes R_X(\frac{\pi}{2})} \quad & |\phi\rangle\langle\phi| \otimes \frac{1}{2}\left(|0\rangle\langle 0| - i\,|0\rangle\langle 1| + i\,|1\rangle\langle 0| + |1\rangle\langle 1|\right) \\
& = \frac{1}{2}\left(|\phi\rangle\langle\phi|\,|0\rangle\langle 0| - i\,|\phi\rangle\langle\phi|\,|0\rangle\langle 1| + i\,|\phi\rangle\langle\phi|\,|1\rangle\langle 0| + |\phi\rangle\langle\phi|\,|1\rangle\langle 1|\right) \\
\xmapsto{\mathsf{CU}} \quad & \frac{1}{2}\left(|\phi\rangle\langle\phi|\,|0\rangle\langle 0| - i\,|\phi\rangle\langle\phi|\,|0\rangle\langle 1|\,U^\dagger + iU\,|\phi\rangle\langle\phi|\,|1\rangle\langle 0| + U\,|\phi\rangle\langle\phi|\,|1\rangle\langle 1|\,U^\dagger\right) \\
\xmapsto{I \otimes \mathsf{Disc}} \quad & \frac{1}{2}\left(|\phi\rangle\langle\phi| + U\,|\phi\rangle\langle\phi|\,U^\dagger\right)
\end{aligned} \tag{4.76}$$

Revisiting the example at hand, the circuit that represents the quantum teleportation protocol with a 50% probability of occurring a bit flip prior to measurement is depicted in Figure 16.

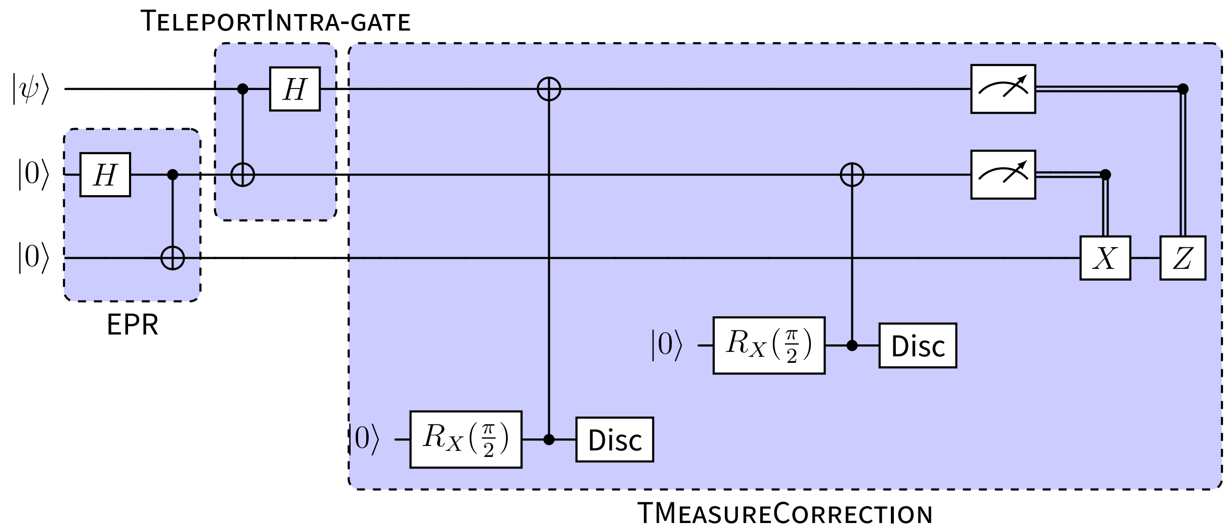


Figure 16: Quantum Teleportation Protocol: Bit flip with 50% probability before measurement.

In this case, the quantum teleportation protocol is divided into three parts: **EPR**, **TeleportIntra-gate** and **TMeasureCorrection**. As a result, it is necessary to define the new functions (note that the function **EPR** is the same as the one defined in subsection 4.2.5):

$$
\begin{aligned}
&\textbf{BellMeasure} : \ \mathit{qbit} \otimes \mathit{qbit} \multimap \mathit{qbit} \otimes \mathit{qbit} \\
&\textbf{TeleportIntra-gate} : \ \mathit{qbit} \otimes \mathit{qbit} \otimes \mathit{qbit} \multimap \mathit{qbit} \otimes \mathit{qbit} \otimes \mathit{qbit} \\
&\textbf{TMeasureCorrection} : \ \mathit{qbit} \otimes \mathit{qbit} \otimes \mathit{qbit} \multimap \mathit{qbit}
\end{aligned}
$$

Considering the operation $T_{X\otimes I\otimes I}$ as the operation $T$ with the unitary $U$ represented by $X \otimes I \otimes I$, and similarly, $T_{I\otimes X\otimes I}$ as $T$ with $U$ denoted by $I \otimes U \otimes I$, these funtions can be defined as follows:

$$
\begin{aligned}
\textbf{TeleportIntra-gate} &= q_1 : \mathit{qbit}, q_2 : \mathit{qbit} \rhd (\text{pm } \mathit{CNOT}(q_1, q_2) \text{ to } x \otimes y.\, H(x) \otimes y) \\
\textbf{TMeasureCorrection} &= q_1 : \mathit{qbit}, q_2 : \mathit{qbit}, q_3 : \mathit{qbit} \rhd \text{pm } T_{X\otimes I\otimes I}(q_1, q_2, q_3) \text{ to } a \otimes b \otimes c. \\
&\quad \text{pm } T_{I\otimes X\otimes I}(a, b, c) \text{ to } d \otimes e \otimes q. \\
&\quad \text{pm } \mathit{meas}(d) \otimes \mathit{meas}(e) \text{ to } x \otimes y. \\
&\quad \text{case } x \,\{\text{inl}(x_0) \Rightarrow (\text{cond } y \,\{\text{inl}(y_0) \Rightarrow I(q);\ \text{inr}(y_1) \Rightarrow X(q))\}; \\
&\qquad \text{inr}(x_1) \Rightarrow (\text{cond } y \,\{\text{inl}(y_0) \Rightarrow Z(q); \text{inr}(y_1) \Rightarrow Z(X(q))\})\}
\end{aligned}
$$

Designating the qubit to be teleported as $q_0$, one can conceptualize he quantum teleportation protocol with a 50% probability of occurring a bit flip prior to measurement as follows:

$$\begin{aligned}
&\text{pm } \textbf{EPR}(*) \text{ to } q_1 \otimes q_2. \\
&\text{pm } \textbf{TeleportIntra-gate}(q_0, q_1) \text{ to } tiq_0 \otimes tiq_1. \\
&\text{pm } \textbf{TMeasureCorrection}(tiq_0, tiq_1, q_2) \text{ to } q.\, q
\end{aligned}$$

Per Equation 4.42, the state of the system post-teleportation protocol corresponds to $|\psi\rangle\langle\psi|$ in the absence of a malicious attack, denoted as $\rho$.

Regarding the first two parts of the teleportation protocol, given Equation 4.40, one has that:

$$\begin{aligned}
&|\psi\rangle\langle\psi| \otimes |0\rangle\langle 0| \otimes |0\rangle\langle 0| \\
\xmapsto{\text{EPR}}\ & \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle) \\
\xmapsto{\text{TeleportIntra-gate}}\ & |00\rangle \otimes |\psi\rangle + |01\rangle \otimes X|\psi\rangle + |10\rangle \otimes Z|\psi\rangle + |11\rangle \otimes XZ|\psi\rangle = |\phi\rangle
\end{aligned} \tag{4.77}$$

Consequently, the state of the system post-teleportation protocol corresponds to $|\phi\rangle\langle\phi|$. With respect to **TMeasureCorrection**, considering that,

$$\begin{aligned}
&|\phi\rangle = |00\rangle \otimes |\psi\rangle + |01\rangle \otimes X|\psi\rangle + |10\rangle \otimes Z|\psi\rangle + |11\rangle \otimes XZ|\psi\rangle \\
\xmapsto{X\otimes I\otimes I}\ & |10\rangle \otimes |\psi\rangle + |11\rangle \otimes X|\psi\rangle + |00\rangle \otimes Z|\psi\rangle + |01\rangle \otimes XZ|\psi\rangle \\
&= |00\rangle \otimes Z|\psi\rangle + |01\rangle \otimes XZ|\psi\rangle + |10\rangle \otimes |\psi\rangle + |11\rangle \otimes X|\psi\rangle = |\phi'\rangle
\end{aligned} \tag{4.78}$$

And,

$$\begin{aligned}
&|\phi\rangle = |00\rangle \otimes |\psi\rangle + |01\rangle \otimes X|\psi\rangle + |10\rangle \otimes Z|\psi\rangle + |11\rangle \otimes XZ|\psi\rangle \\
\xmapsto{I\otimes X\otimes I}\ & |01\rangle \otimes |\psi\rangle + |00\rangle \otimes X|\psi\rangle + |11\rangle \otimes Z|\psi\rangle + |10\rangle \otimes XZ|\psi\rangle \\
&= |00\rangle \otimes X|\psi\rangle + |01\rangle \otimes |\psi\rangle + |10\rangle \otimes XZ|\psi\rangle + |11\rangle \otimes Z|\psi\rangle = |\phi''\rangle
\end{aligned} \tag{4.79}$$

And finally,

$$\begin{aligned}
&|\phi'\rangle = |00\rangle \otimes Z|\psi\rangle + |01\rangle \otimes XZ|\psi\rangle + |10\rangle \otimes |\psi\rangle + |11\rangle \otimes X|\psi\rangle \\
\xmapsto{I\otimes X\otimes I}\ & |01\rangle \otimes Z|\psi\rangle + |00\rangle \otimes XZ|\psi\rangle + |11\rangle \otimes |\psi\rangle + |10\rangle \otimes X|\psi\rangle \\
&= |00\rangle \otimes XZ|\psi\rangle + |01\rangle \otimes Z|\psi\rangle + |10\rangle \otimes X|\psi\rangle + |11\rangle \otimes |\psi\rangle = |\phi'''\rangle
\end{aligned} \tag{4.80}$$

It follows that,

$$
\begin{aligned}
& |\phi\rangle \langle\phi| \\
\xmapsto{T_{X\otimes I\otimes I}} \quad & \frac{1}{2}\left(|\phi\rangle \langle\phi| + |\phi'\rangle \langle\phi'|\right) && \{\text{Equation 4.76}\} \\
\xmapsto{T_{I\otimes X\otimes I}} \quad & \frac{1}{4}\left(|\phi\rangle \langle\phi| + |\phi'\rangle \langle\phi'| + |\phi''\rangle \langle\phi''| + |\phi'''\rangle \langle\phi'''|\right) && \{\text{Equation 4.76}\} \\
\xmapsto{\text{meas}\otimes\text{meas}\otimes I} \quad & \frac{1}{4}\Bigg(\bigg(\Big(\frac{1}{4}|\psi\rangle\langle\psi|, \frac{1}{4}X|\psi\rangle\langle\psi|X\Big), \Big(\frac{1}{4}Z|\psi\rangle\langle\psi|Z, \frac{1}{4}XZ|\psi\rangle\langle\psi|ZX\Big)\bigg) \\
& + \bigg(\Big(\frac{1}{4}Z|\psi\rangle\langle\psi|Z, \frac{1}{4}XZ|\psi\rangle\langle\psi|XZ\Big), \Big(\frac{1}{4}|\psi\rangle\langle\psi|, \frac{1}{4}X|\psi\rangle\langle\psi|X\Big)\bigg) \\
& + \bigg(\Big(\frac{1}{4}X|\psi\rangle\langle\psi|X, \frac{1}{4}|\psi\rangle\langle\psi|\Big), \Big(\frac{1}{4}XZ|\psi\rangle\langle\psi|ZX, \frac{1}{4}Z|\psi\rangle\langle\psi|Z\Big)\bigg) \\
& + \bigg(\Big(\frac{1}{4}XZ|\psi\rangle\langle\psi|ZX, \frac{1}{4}Z|\psi\rangle\langle\psi|Z\Big), \Big(\frac{1}{4}X|\psi\rangle\langle\psi|X, \frac{1}{4}|\psi\rangle\langle\psi|\Big)\bigg)\Bigg)
\end{aligned}
\tag{4.81}
$$

Next, regarding the condicional statements, applying correction to $\Big(\big(\frac{1}{4}|\psi\rangle\langle\psi|, \frac{1}{4}X|\psi\rangle\langle\psi|X\big), \big(\frac{1}{4}Z|\psi\rangle\langle\psi|Z,$ results in the state $|\psi\rangle$ (Equation 4.42). Moreover, with respect to $\Big(\big(\frac{1}{4}|\psi\rangle\langle\psi|, \frac{1}{4}X|\psi\rangle\langle\psi|X\big), \big(\frac{1}{4}Z|\psi\rangle\langle\psi|Z, \frac{1}{4}X$ one has that applying the condicional statements:

$$
\begin{aligned}
& \frac{1}{4}Z\,|\psi\rangle\,\langle\psi|\,Z + \frac{1}{4}XXZ\,|\psi\rangle\,\langle\psi|\,ZXX + \frac{1}{4}Z\,|\psi\rangle\,\langle\psi|\,Z + \frac{1}{4}ZXX\,|\psi\rangle\,\langle\psi|\,XXZ \\
=& \frac{1}{4}Z\,|\psi\rangle\,\langle\psi|\,Z + \frac{1}{4}Z\,|\psi\rangle\,\langle\psi|\,Z + \frac{1}{4}Z\,|\psi\rangle\,\langle\psi|\,Z + \frac{1}{4}Z\,|\psi\rangle\,\langle\psi|\,Z = Z\,|\psi\rangle\,\langle\psi|\,Z
\end{aligned}
\tag{4.82}
$$

Furthermore, applying correction to $\Big(\big(\frac{1}{4}X|\psi\rangle\langle\psi|X, \frac{1}{4}|\psi\rangle\langle\psi|\big), \big(\frac{1}{4}XZ|\psi\rangle\langle\psi|ZX, \frac{1}{4}Z|\psi\rangle\langle\psi|Z\big)\Big)$ results in

$$
\begin{aligned}
& \frac{1}{4}X\,|\psi\rangle\,\langle\psi|\,X + \frac{1}{4}X\,|\psi\rangle\,\langle\psi|\,X + \frac{1}{4}ZXZ|\psi\rangle\langle\psi|ZXZ + \frac{1}{4}ZXZ|\psi\rangle\langle\psi|ZXZ \\
=& \frac{1}{4}X\,|\psi\rangle\,\langle\psi|\,X + \frac{1}{4}X\,|\psi\rangle\,\langle\psi|\,X + \frac{1}{4}X\,|\psi\rangle\,\langle\psi|\,X + \frac{1}{4}X\,|\psi\rangle\,\langle\psi|\,X = X\,|\psi\rangle\,\langle\psi|\,X
\end{aligned}
\tag{4.83}
$$

And, at last, regarding $\Big(\big(\frac{1}{4}XZ|\psi\rangle\langle\psi|ZX, \frac{1}{4}Z|\psi\rangle\langle\psi|Z\big), \big(\frac{1}{4}X|\psi\rangle\langle\psi|X, \frac{1}{4}|\psi\rangle\langle\psi|\big)\Big)$,

$$
\begin{aligned}
& \frac{1}{4}XZ\,|\psi\rangle\,\langle\psi|\,ZX + \frac{1}{4}XZ\,|\psi\rangle\,\langle\psi|\,ZX + \frac{1}{4}ZX\,|\psi\rangle\,\langle\psi|\,XZ + \frac{1}{4}ZX\,|\psi\rangle\,\langle\psi|\,XZ \\
=& \frac{1}{4}XZ\,|\psi\rangle\,\langle\psi|\,ZX + \frac{1}{4}XZ\,|\psi\rangle\,\langle\psi|\,ZX + \frac{1}{4}XZ\,|\psi\rangle\,\langle\psi|\,ZX + \frac{1}{4}XZ\,|\psi\rangle\,\langle\psi|\,ZX \\
=& ZX\,|\psi\rangle\,\langle\psi|\,XZ
\end{aligned}
\tag{4.84}
$$

Consequentlty, applying the conditional statements to the state obtained in Equation 4.81, it follows that,

$$
\begin{aligned}
&\frac{1}{4}\left(|\psi\rangle\langle\psi| + Z|\psi\rangle\langle\psi| Z + X|\psi\rangle\langle\psi| X + ZX|\psi\rangle\langle\psi| XZ\right)\\
=&\frac{1}{4}(|\alpha|^2|0\rangle\langle 0| + \alpha\beta^\dagger|0\rangle\langle 1| + \alpha^\dagger\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|\\
&+|\alpha|^2|0\rangle\langle 0| - \alpha\beta^\dagger|0\rangle\langle 1| - \alpha^\dagger\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|\\
&+|\beta|^2|0\rangle\langle 0| + \alpha^\dagger\beta|0\rangle\langle 1| + \alpha\beta^\dagger|1\rangle\langle 0| + |\alpha|^2|1\rangle\langle 1|\\
&+|\beta|^2|0\rangle\langle 0| - \alpha^\dagger\beta|0\rangle\langle 1| - \alpha\beta^\dagger|1\rangle\langle 0| + |\alpha|^2|1\rangle\langle 1|)\\
=&\frac{|\alpha|^2+|\beta|^2}{2}|0\rangle\langle 0| + \frac{|\alpha|^2+|\beta|^2}{2}|1\rangle\langle 1| = \rho'
\end{aligned}
\tag{4.85}
$$

Therefore, $\rho - \rho'$ corresponds to:

$$
\begin{aligned}
\rho - \rho' =&|\alpha|^2|0\rangle\langle 0| + \alpha\beta^\dagger|0\rangle\langle 1| + \alpha^\dagger\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|\\
&-\left(\frac{|\alpha|^2+|\beta|^2}{2}|0\rangle\langle 0| + \frac{|\alpha|^2+|\beta|^2}{2}|1\rangle\langle 1|\right)\\
=&\frac{|\alpha|^2-|\beta|^2}{2}|0\rangle\langle 0| + \alpha\beta^\dagger|0\rangle\langle 1| + \alpha^\dagger\beta|1\rangle\langle 0| + \frac{|\beta|^2-|\alpha|^2}{2}|1\rangle\langle 1|
\end{aligned}
\tag{4.86}
$$

Employing Equation 3.6, the components of the Bloch vector of the state $\rho - \rho'$ are as follows:

$$
\begin{aligned}
r_x =&\text{Tr}\left[\begin{pmatrix}\frac{|\alpha|^2-|\beta|^2}{2} & \alpha\beta^\dagger\\ \alpha^\dagger\beta & \frac{|\beta|^2-|\alpha|^2}{2}\end{pmatrix}\cdot\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\right] = \text{Tr}\left[\begin{pmatrix}\alpha\beta^\dagger & \frac{|\alpha|^2-|\beta|^2}{2}\\ \frac{|\beta|^2-|\alpha|^2}{2} & \alpha^\dagger\beta\end{pmatrix}\right] = \alpha\beta^\dagger + \alpha^\dagger\beta\\
r_y =&\text{Tr}\left[\begin{pmatrix}\frac{|\alpha|^2-|\beta|^2}{2} & \alpha\beta^\dagger\\ \alpha^\dagger\beta & \frac{|\beta|^2-|\alpha|^2}{2}\end{pmatrix}\cdot\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}\right] = \text{Tr}\left[\begin{pmatrix}i\alpha\beta^\dagger & \frac{|\alpha|^2-|\beta|^2}{2}\\ \frac{|\beta|^2-|\alpha|^2}{2} & -i\alpha^\dagger\beta\end{pmatrix}\right] = i(\alpha\beta^\dagger - \alpha^\dagger\beta)\\
r_z =&\text{Tr}\left[\begin{pmatrix}\frac{|\alpha|^2-|\beta|^2}{2} & \alpha\beta^\dagger\\ \alpha^\dagger\beta & \frac{|\beta|^2-|\alpha|^2}{2}\end{pmatrix}\cdot\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\right] = \text{Tr}\left[\begin{pmatrix}\frac{|\alpha|^2-|\beta|^2}{2} & -\alpha\beta^\dagger\\ \alpha^\dagger\beta & -\frac{|\beta|^2-|\alpha|^2}{2}\end{pmatrix}\right] = |\alpha|^2 - |\beta|^2
\end{aligned}
\tag{4.87}
$$

Considering that the distance between two vectors corresponds to their Euclidean distance, it follows that the distance between the ideal state and its version subjected to the malicious attack is given by:

$$
\begin{aligned}
&\|\rho - \rho'\|_\diamond \\
&= \left\|\left(\alpha\beta^\dagger + \alpha^\dagger\beta, i(\alpha\beta^\dagger - \alpha^\dagger\beta), |\alpha|^2 - |\beta|^2\right)\right\|_2 && \{\text{Equation 4.87}\} \\
&= \sqrt{(\alpha\beta^\dagger + \alpha^\dagger\beta)^2 + (i(\alpha\beta^\dagger - \alpha^\dagger\beta))^2 + (|\alpha|^2 - |\beta|^2)^2} && \{\text{Definition 3.1.21}\} \\
&= \sqrt{(\alpha\beta^\dagger + \alpha^\dagger\beta)^2 + -(\alpha\beta^\dagger - \alpha^\dagger\beta)^2 + (|\alpha|^2 - |\beta|^2)^2} \\
&= \sqrt{4\alpha\beta^\dagger\alpha^\dagger\beta + |\alpha|^4 - 2|\alpha|^2|\beta|^2 + |\beta|^4} = \sqrt{4|\alpha|^2|\beta|^2 + |\alpha|^4 - 2|\alpha|^2|\beta|^2 + |\beta|^4} \\
&= \sqrt{|\alpha|^4 + 2|\alpha|^2|\beta|^2 + |\beta|^4} = \sqrt{(|\alpha|^2 + |\beta|^2)^2} = |\alpha|^2 + |\beta|^2 = 1
\end{aligned}
\tag{4.88}
$$

# Chapter 5

# Graded modalities

The linearity constraint is often deemed too restrictive, prompting research into relaxing it in various computational paradigms. In Dahlqvist and Neves [2023b], the controlled use of a resource multiple times is explored within approximate program equivalence paradigms. Moreover, the grammar introduced allows the specification of how many times a resource can be used—a notion particularly relevant in quantum computation, especially within the NISQ era where resources are scarce.

Intro

## 5.1 Syntax

Here, the following grammar of types is used.

$$\mathbb{A} ::= X \mid \mathbb{I} \mid \mathbb{A} \otimes \mathbb{A} \mid \mathbb{A} \oplus \mathbb{A} \mid \mathbb{A} \multimap \mathbb{A} \mid\, !_r\mathbb{A} \qquad\qquad X \in G, r \in \mathbb{N}$$

$$\frac{\Gamma_i \rhd v_i :!_{r \cdot s_i}\mathbb{A}_i \quad x_1 :!_{s_1}\mathbb{A}_1, \ldots, x_n :!_{s_n}\mathbb{A}_n \rhd u : \mathbb{A} \quad E \in \mathsf{Sf}(\Gamma_1; \ldots; \Gamma_n)}{E \rhd \mathsf{pr}_{(r,[s_1,\ldots,s_n])} v_1, \ldots, v_n \ \mathsf{fr}\ x_1, \ldots, x_n.u :!_r\mathbb{A}}(!_i) \qquad \frac{\Gamma \rhd v :!_1\mathbb{A}}{\Gamma \rhd \mathsf{dr}\ v : \mathbb{A}}(!_e)$$

$$\frac{\Gamma \rhd v :!_0\mathbb{A} \quad \Delta \rhd u : \mathbb{B} \quad E \in \mathsf{Sf}(\Gamma, \Delta)}{E \rhd v.u : \mathbb{B}}(!_0) \qquad \frac{\Gamma \rhd v :!_{n+m}\mathbb{A} \quad \Delta, x :!_n\mathbb{A}, y :!_m\mathbb{B} \rhd u : \mathbb{B} \quad E \in \mathsf{Sf}(\Gamma, \Delta)}{E \rhd \mathsf{cp}_{(n,m)} v \ \mathsf{to}\ x, y.u : \mathbb{B}}(!_{n+m})$$

Figure 17: Term formation rules of graded lambda calculus.

## 5.2 Interpretation

## 5.3 Quantum State Discrimination

Chapter 6

# Conclusions and future work

Conclusions and future work.

## 6.1 Conclusions

## 6.2 Prospect for future work

# Bibliography


John W Backus, Friedrich L Bauer, Julien Green, Charles Katz, John McCarthy, Alan J Perlis, Heinz Rutishauser, Klaus Samelson, Bernard Vauquois, Joseph Henry Wegstein, et al. Report on the algorithmic language algol 60. *Communications of the ACM*, 3(5):299–311, 1960.

Hendrik P Barendregt et al. *The lambda calculus*, volume 3. North-Holland Amsterdam, 1984.

Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical review letters*, 70(13):1895, 1993.

P Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *International Workshop on Computer Science Logic*, pages 121–135. Springer, 1994.

Paul Bernays. Alonzo church. an unsolvable problem of elementary number theory. american journal of mathematics, vol. 58 (1936), pp. 345–363. *The Journal of Symbolic Logic*, 1 (2):73–74, 1936.

Benjamin Bichsel, Maximilian Baader, Timon Gehr, and Martin Vechev. Silq: A high-level quantum language with safe uncomputation and intuitive semantics. In *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 286–300, 2020.

H-J Briegel, Wolfgang Dür, Juan I Cirac, and Peter Zoller. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters*, 81(26): 5932, 1998.

Lukas Burgholzer and Robert Wille. Advanced equivalence checking for quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(9):1810–1824, 2020.

A Robert Calderbank and Peter W Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098, 1996.

Frederic T Chong, Diana Franklin, and Margaret Martonosi. Programming languages and compiler design for realistic quantum hardware. *Nature*, 549(7671):180–187, 2017.

Roy L Crole. *Categories for types*. Cambridge University Press, 1993.

Fredrik Dahlqvist and Renato Neves. The syntactic side of autonomous categories enriched over generalised metric spaces. *Logical Methods in Computer Science*, 19, 2023a.

Fredrik Dahlqvist and Renato Neves. A complete v-equational system for graded lambda-calculus. *Electronic Notes in Theoretical Informatics and Computer Science*, 3, 2023b.

David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777, 1935.

Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7), 1982.

Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012.

Jean-Yves Girard, Yves Lafont, and Laurent Regnier. *Advances in linear logic*, volume 222. Cambridge University Press, 1995.

Daniel Gottesman. *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.

Daniel Gottesman and Isaac L Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390–393, 1999.

Alexander S Green, Peter LeFanu Lumsdaine, Neil J Ross, Peter Selinger, and Benoît Valiron. Quipper: a scalable quantum programming language. In *Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation*, pages 333–342, 2013.

A Hitchhiker's Guide. *Infinite dimensional analysis*. Springer, 2006.

Aram W Harrow, Benjamin Recht, and Isaac L Chuang. Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics*, 43(9):4445–4451, 2002.

Jim Hefferon. *Linear algebra*. -, 2006.

Shih-Han Hung, Kesha Hietala, Shaopeng Zhu, Mingsheng Ying, Michael Hicks, and Xiaodi Wu. Quantitative robustness analysis of quantum programs. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.

H Jeff Kimble. The quantum internet. *Nature*, 453(7198):1023–1030, 2008.

Radu Mardare, Prakash Panangaden, and Gordon Plotkin. Quantitative algebraic reasoning. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 700–709, 2016.

Radu Mardare, Prakash Panangaden, and Gordon Plotkin. On the axiomatizability of quantitative algebras. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

Prakash Murali, Jonathan M Baker, Ali Javadi-Abhari, Frederic T Chong, and Margaret Martonosi. Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. In *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*, pages 1015–1029, 2019.

Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

Simon Perdrix. Quantum entanglement analysis based on abstract interpretation. In *International Static Analysis Symposium*, pages 270–282. Springer, 2008.

John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.

Walter Rudin. *Functional Analysis*. McGraw-Hill, 1991.

Alejo Salles, Fernando de Melo, MP Almeida, Malena Hor-Meyll, SP Walborn, PH Souto Ribeiro, and Luiz Davidovich. Experimental investigation of the dynamics of entanglement: Sudden death, complementarity, and continuous monitoring of the environment. *Physical Review A*, 78(2):022322, 2008.

Maximilian Schlosshauer. Decoherence, the measurement problem, and interpretations of quantum mechanics. *Reviews of Modern physics*, 76(4):1267, 2005.

Peter Selinger. Lecture notes on the lambda calculus, 2013.

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

Peter Selinger, Benoıt Valiron, et al. Quantum lambda calculus. *Semantic techniques in quantum computation*, pages 135–172, 2009.

Manuel A Serrano, Jose A Cruz-Lemus, Ricardo Perez-Castillo, and Mario Piattini. Quantum software components and platforms: Overview and quality assessment. *ACM Computing Surveys*, 55(8):1–31, 2022.

Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

Michael Shulman. A practical type theory for symmetric monoidal categories. *Theory and Applications of Categories*, 37(5):863–907, 2021.

Andrew M Steane. Error correcting codes in quantum theory. *Physical Review Letters*, 77(5): 793, 1996.

Krysta Svore, Alan Geller, Matthias Troyer, John Azariah, Christopher Granade, Bettina Heim, Vadym Kliuchnikov, Mariia Mykhailova, Andres Paz, and Martin Roetteler. Q# enabling scalable quantum computing and development with a high-level dsl. In *Proceedings of the real world domain specific languages workshop 2018*, pages 1–10, 2018.

Swamit S Tannu and Moinuddin K Qureshi. Mitigating measurement errors in quantum computers by exploiting state-dependent bias. In *Proceedings of the 52nd annual IEEE/ACM international symposium on microarchitecture*, pages 279–290, 2019.

Runzhou Tao, Yunong Shi, Jianan Yao, John Hui, Frederic T Chong, and Ronghui Gu. Gleipnir: toward practical error analysis for quantum programs. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pages 48–64, 2021.

Joel J Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94(5):052325, 2016.

Jing Wang, Li Jiang, Han Zhang, Hanzhuang Zhang, and Liquan Zhang. Fidelity of structured amplitude-damping channels. *Physica Scripta*, 83(4):045008, mar 2011. doi: 10.1088/0031-8949/83/04/045008. URL https://dx.doi.org/10.1088/0031-8949/83/04/045008.

John Watrous. *The theory of quantum information*. Cambridge university press, 2018.

Glynn Winskel. *The formal semantics of programming languages: an introduction*. MIT press, 1993.

William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299 (5886):802–803, 1982.

Jianjun Zhao. Quantum software engineering: Landscapes and horizons. *arXiv preprint arXiv:2007.07047*, 2020.

Margherita Zorzi. On quantum lambda calculi: a foundational perspective. *Mathematical Structures in Computer Science*, 26(7):1107–1195, 2016.

Place here information about funding, FCT project, etc. in which the work is framed. Leave empty otherwise.