University of Minho
School of Engineering

Bruna Filipa Martins Salgado

Metric λ-calculus with conditionals: quantum, probabilities and beyond

August 2025

University of Minho
School of Engineering

Bruna Filipa Martins Salgado

Metric λ-calculus with conditionals: quantum, probabilities and beyond

Master's Dissertation
Master in Physics Engineering

Work carried out under the supervision of
Renato Jorge Araújo Neves

August 2025

Copyright and Terms of Use for Third Party Work

License granted to users of this work:

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 2025

Bruna Filipa Martins Salgado

Abstract

In recent decades, there has been an effort in computer science to move beyond rigid binary notions—such as equality and bisimulation—toward more flexible approaches that better reflect the subtleties of real-world computation. Traditional program equivalence, for example, is purely dichotomous: two programs are either equivalent or not. Yet in many computational paradigms, this binary perspective proves too restrictive. For instance, in contexts involving physical environments and noisy data, a more nuanced notion — such as approximate program equivalence—becomes imperative. It is within this evolving landscape that our work is situated.

We build on the work of [1] by introducing a metric equation for conditionals and proving its soundness and completeness. Syntactically, to illustrate the utility of the metric equation introduced, we present a illustrative example: a metric version of the copairing's extensionality. On the semantic side, we present five categories that satisfy the necessary requirements for interpreting this equation, thereby demonstrating the broad applicability of our approach across several domains. Finally, we illustrate the use of the metric equation in more detail within both the probabilistic and quantum computing paradigms. For quantum models, we focus on the first-order fragment of the λ-calculus, though extensions to higher-order are possible using advanced categorical tools, as in [1].

Keywords quantitative reasoning, λ-calculus, metric equations

Resumo

Escrever aqui o resumo (pt)

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

Contents

List of Figures

Acronyms

NISQ Noisy Intermediate-Scale Quantum 8

BNF Backus-Naur Form 13

CPTP Completely Positive Trace-Preserving 86

OSR Operator Sum Representation 87

miu Multiplicative involutive unital 98

Notation

$FV(v)$ Set of free variables of a term v. 14

$v[w/x]$ Substitution of a variable x for a term w in a term v. 15

Γ, Δ, E Typical symbols for typing contexts. 17

$\Gamma \triangleright v : \mathbb{A}$ Typing judgement. 17

$\Gamma \triangleright v = w : \mathbb{A}$ Equation-in-context. 22

$\Gamma \triangleright t =_\epsilon s$ Metric equation-in-context. 29

V, W Typical symbols for vector spaces. 56

$\mathcal{F}$ Field of scalars of a vector space. 56

$\| \cdot \|$ Norm of an arbitrary vector. 56

$d(v, w)$ distance between vectors v and w. 56

$\| \cdot \|_{\mathbf{op}}$ Operator norm. 56

$\mathcal{B}(V, W)$ Vector space of all bounded linear operators from V to W. 56

$\mathcal{B}(V)$ Vector space of all bounded linear operators from V to itself. 56

$\otimes_{\mathbf{meas}}$ Product measure. 63

$\mathcal{M}\mathbb{R}$ Banach space of finite Borel measures on $\mathbb{R}$. 63

$\langle \cdot, \cdot \rangle$ Inner product. 72

$\overline{(-)}$ Complex conjugate operation. 72

$\mathcal{H}, \mathcal{K}$ Typical symbols for Hilbert spaces. 72

$(-)^\dagger$ Adjoint operation. 73

U Typical designation for a unitary operator. 73

ρ Typical designation for a density matrix. 74

$(-)^{\otimes n}$ N-fold tensor product. 76

$\|\cdot\|_2$ Euclidean norm. 76

$\|\cdot\|_1$ Trace norm. 76

$\mathcal{H} \otimes_2 \mathcal{K}$ Hilbert space tensor product of $\mathcal{H}$ and $\mathcal{K}$. 76

$|\psi\rangle$ Typical symbols for a quantum state. Also known as ket. 77

$\langle\psi|$ Typical symbol for $|\psi\rangle^\dagger$. Also known as bra. 77

$\langle\psi|\phi\rangle$ Inner product between states $|\psi\rangle$ and $|\phi\rangle$. 77

$|\psi\rangle \otimes |\phi\rangle$ Tensor product of states $|\psi\rangle$ and $|\phi\rangle$. 79

$|\psi\rangle\,|\phi\rangle$ Tensor product of states $|\psi\rangle$ and $|\phi\rangle$. 79

$|\psi\phi\rangle$ Tensor product of states $|\psi\rangle$ and $|\phi\rangle$. 79

X Pauli operator σ_x. 81

Z Pauli operator σ_z. 81

H Hadamard gate 82

CNOT Controlled Not gate 82

$\|\cdot\|_\diamond$ Diamond norm. 88

V^* Topological dual space of V 94

$\mathscr{A}, \mathscr{B}, \mathscr{C}, \ldots$ Typical symbols for C*-algebras. 97

$\mathscr{M}, \mathscr{N}, \mathscr{T}$ Typical symbols for C*-algebras. 102

$\mathcal{M}_n$ Vector space of complex $n \times n$ matrices. 108

Part I

Foundations

Chapter 1

Introduction

1.1 Motivation and Context

Some History

Hilbert's Optimistic Vision of Mathematics In September 1928, David Hilbert presented his vision for the foundations of mathematics at the International Congress of Mathematicians in Bologna. He believed it possible to place mathematics on an absolutely secure foundation. This would mean that no matter how difficult a mathematical problem might be, one would only need to "take up the pen, sit at the abacus, and calculate" [2]. The process would be entirely deterministic, requiring no intuition or creativity, only strict adherence to formal rules, like performing multiplication in decimal notation. Every problem would, in principle, be solvable by such mechanical procedures. Mathematics would be both complete (able to answer every question) and consistent (free of contradictions).

Gödel's Incompleteness Theorems However, this vision was shattered by Kurt Gödel's Incompleteness Theorems (1931) [3], which showed that no set of mathematical rules powerful enough to handle basic arithmetic could ever be both complete (answering every question) and consistent (free of contradictions) at the same time.

Gödel's Completeness Theorems Interestingly, in his doctoral thesis, Gödel proved a foundational result in logic: first-order predicate logic—a formal system used to express statements involving quantifiers like "for all" and "there exists"—is *complete* [4]. It is important to note that the term "completeness" here differs from its use in Gödel's later Incompleteness Theorems. Before exploring this notion of completeness, it is helpful to introduce a few core concepts. *Syntax* refers to the formal symbols and inference rules used to construct well-formed statements, while *semantics* concerns the meaning assigned to these statements through interpretations. A *model* of a first-order system is a mathematical structure in which the axioms (or rules) hold true under a given interpretation. With these notions in place, we can now turn to Gödel's result, known as the *Completeness Theorem*. This theorem states that if a statement holds in every possible model of a theory, then it can also be *syntactically* proven using the system's formal rules. In other words, completeness is the property that all universally valid statements are provable within the system. The converse also holds: any statement provable syntactically must hold true in all models. This is known as *soundness* [5].

Entscheidungsproblem We have just introduced two of the main pillars of this thesis — soundness and completeness. We now introduce a third, deeply tied to Hilbert's ambitious vision for mathematics. Recall that Hilbert not only sought a complete and consistent foundation for mathematics, but also believed in the possibility of an entirely *mechanical* method to resolve any mathematical problem—a process requiring no intuition or creative insight. In 1928, he formulated the Entscheidungsproblem (German for "decision problem"), which sought an *effective method* (also called a mechanical procedure or algorithm) to determine the truth or falsity of any mathematical statement [6]. A method or procedure is effective if:

1. it can be described by a finite number of exact instructions;
2. it produces the desired result after a finite number of steps (provided the instructions are followed without error);
3. it can, in principle, be carried out by a human using only paper and pencil;
4. it does not require any creativity or insight from the human.

The algorithms that children learn to perform basic arithmetic operations are examples of effective procedures.

Alonzo Church and the λ-calculus enter the scene Remarkably, it was Alonzo Church—using λ-*calculus*—who first addressed Hilbert's *Entscheidungsproblem*. This brings us to the third central theme of this dissertation: λ-*calculus*. In 1936, Alonzo Church published a solution to the *Entscheidungsproblem*, proving that no universal algorithmic method could decide the truth of all mathematical statements [7]. Today, this result is often referred to as *Church's Theorem*. In the same work, he provided a mathematically precise notion of an effective method. He proposed that a function is effectively computable if and only if it can be written as a lambda term. This equivalence provided the first rigorous mathematical criterion for computability.

This calculus played an important role in functional programming, influencing the design of languages like LISP, Pascal, and GEDANKEN—many of which incorporate λ-calculus-inspired features, either explicitly or implicitly. Furthermore, λ-calculus may be employed to prove properties of programming language (such as: a well-formed program will not crash) and as a tool in the construction of compilers [8].

The idea that such important aspects of modern computer science emerged from foundational questions in mathematics is nothing short of extraordinary.

A note on Turing's work Around the same time, another researcher—unaware of Church's work—was independently addressing the same problem: Alan Turing, now widely regarded as the father of computer science. He introduced the concept of a universal machine, now known as the *Turing Machine*. While Turing's model is groundbreaking in its own right, it is largely orthogonal to this dissertation, as it is more closely associated with automata theory than with the syntax and semantics of programming languages.

λ-calculus

λ-calculus and functions The λ-calculus is a formal language that captures a key feature of higher-order functional languages: treating functions as "first-class citizens" that can be passed as arguments. Here, functions are expressed as *abstractions* of the form $\lambda x.\, f(x)$, with application denoted by juxtaposing the abstraction with its argument. For example, the expression $f(2)$, where $f(x) = x + 1$, is written as $(\lambda x.\, x + 1)(2)$.

Typed-lambda calculus In this work, we use the typed λ-calculus, a variant where each term is "labeled" by a syntactic object called a *type*. Types serve as a mechanism for ensuring that programs are meaningful. In contrast, the untyped version allows, for example, a function to be applied to itself, as in $(\lambda x.\, x\, x)(\lambda x.\, x\, x)$, leading to non-termination. It also allows nonsensical operations, such as applying a boolean to a number or a function to a string.

λ-calculus and logic We previously mentioned that one of our main results pertains to soundness and completeness—a notion we deliberately introduced in the setting of (first-order) logic. In fact, the typed lambda calculus itself is equipped with an equational logic, i.e., a system of equations. These equations arise because the λ-calculus includes n-ary function symbols, which may be accompanied by equality axioms specifying their intended properties. Moreover, the lambda calculus allows us to establish a correspondence between logical proofs and programs. This is known as the *Curry-Howard isomorphism* [9].

Semantics:λ-calculus and category theory In this work, we interpret programs as mathematical objects, particularly those arising in category theory. But why choose a categorical interpretation over other alternatives?

Consider an ancient Indian parable: six blind men encounter an elephant for the first time. Each man touches a different part of the animal—the side, tusk, trunk, leg, ear, or tail—and draws a conclusion based solely on that limited experience. One describes it as a spear (the tusk), another as a snake (the trunk), and another as a fan (the ear). Each is convinced of his own interpretation and dismisses the others as incorrect. None of them realise that they are each experiencing only a part of the same elephant, and that their individual descriptions are incomplete. In some versions of the story, the men stop arguing, begin listening to one another, and collaborate to form a more accurate understanding of the whole elephant.

Category theory plays a similar role in computer science. Each category embodies a distinct perspective —a ”part of the elephant” — capturing a specific computational paradigm. Adopting a categorical approach allows us to generalize our results across diverse computational paradigms.

However, there is a deeper reason for using category theory in this setting: it is intimately connected to the λ-calculus. First, it should be noted that λ-calculus is a type theory — and here lies the twist: categories themselves can be viewed as type theories. The objects may be regarded as types (of sorts), and the arrows as functions between those types. In this sense, a category may be thought of as a type theory stripped of its syntax. With this perspective in mind, in the 1970s, Joachim Lambek established a correspondence between cartesian closed categories and the λ-calculus [10]. That is, types correspond to objects, and programs correspond to arrows in such categories. This correspondence extends further to logic, under the so-called Curry–Howard–Lambek correspondence, where formulas correspond to types and proofs to arrows. Later, Lambek and Dana Scott independently observed that C-monoids (*i.e.*categories equipped with products, exponentials, and a single non-terminal object) correspond to the untyped λ-calculus [11].

Going quantitative

Beyond its foundational aspects, this calculus incorporates extensions for modeling side effects, including probabilistic or non-deterministic behaviors and shared memory. In this work, we are concerned with a version of λ-calculus that allows us to reason about approximate equivalence of programs, referred to as *metric λ-calculus*. The metric lambda calculus integrates notions of approximation into the equational system of linear lambda calculus, a variant of lambda calculus that restricts each variable to being exactly once.

Program equivalence and its underlying theories traditionally rely on a binary notion of equivalence: two programs are either equivalent or not [12]. While this dichotomy is often sufficient for classical programming, it proves too coarse-grained for other computational paradigms. For instance, in various programming paradigms, interaction with the physical environment calls for notions of approximate program equivalence.

To address this, [1, 13] incorporate a notion of approximate equivalence into the equational system of the affine λ-calculus by introducing, among other elements, *metric equations* [14, 15]. These are equations of the form $t =_{\varepsilon} s$, where ε is a non-negative real number representing the "maximum distance" between terms t and s. Here we begin exploring the incorporation of a metric equation for the case statements (*i.e.* conditionals). Our motivation for it is highly practical: in trying to reason quantitatively about higher-order programs, we often fell short when these involved conditionals.

Quantitative logics offer a way forward, extending beyond λ-calculus. They reflect a broader effort to move beyond rigid binary concepts, such as equality and bisimulation, and toward more flexible frameworks better suited to real-world computation.

Other works in the spirit of this dissertation include [14–17], which explore (generalized) metric universal algebras. In simple terms, a universal algebra is a set equipped with any number of operations, further defined by axioms typically expressed as identities or equational laws. In a (generalized) metric universal algebra, these axioms are relaxed into (generalized) metric equations rather than strict equalities. In the higher-order setting, [18], following the framework introduced by Mardare [14], investigates the problem of defining quantitative algebras that are capable of interpreting terms in higher-order calculi.

Probabilistic programs are quite ubiquitous: they control autonomous systems, verify security protocols, and implement randomized algorithms for solving computationally intractable problems. At their core, they aim to democratize probabilistic modeling by providing programmers with expressive, high-level abstractions for machine learning and statistical reasoning [19]. In this context, concerns such as developing more eco-friendly programs and algorithms could greatly benefit from a quantitative approach.

Quantum Computation

In 1994, Peter Shor demonstrated that a quantum computer with sufficiently many qubits could pose a significant threat to the security of confidential data transmitted over the Internet [20]. This breakthrough spurred widespread interest in quantum computing. Nevertheless, **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, quantitative reasoning is indispensable for the design, optimization, and assessment of NISQ computing.

1.2 Contributions

Our contributions fall into three categories: syntactic, semantic, and concerned with the connection between the two.

Syntatic

We build on the work of [1] by introducing a metric equation for conditionals. Next, to illustrate the utility of the metric equation introduced, we present a simple example: a metric version of the copairing's extensionality. Moreover, we illustrate the usefulness of our(metric) equational system by using it as a bridge to connect a certain type to Boolean algebra.

Semantic

Returning to our earlier elephant parable, we study various "perspectives" by proving that the corresponding categories are indeed models suitable for reasoning about approximate equivalence using this equation:

- The category of metric spaces Met;

- Cocompletion of a Met-category C;

- Category Ban of Banach spaces and short maps;
- Selinger's Q, the category of quantum operations (*i.e.*, completely positive, trace non-increasing superoperators) [21], and Cho's $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$, the opposite category of $\mathsf{W}^*_{\mathsf{CPSU}}$, the category of W*-algebras and normal, completely positive, subunital maps [22].

This demonstrates that our work is applicable across several domains. For the last two quantum models, we restrict our attention to the first-order fragment of the λ-calculus, noting that extensions to higher-order are possible using more advanced categorical tools, as in [1].
We investigate two computational paradigms in greater detail: probabilistic (via Ban) and quantum computation (via Q and $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$). In the probabilistic setting, we use a random walk to reason about approximate equivalence. In the quantum setting, we use three examples: quantum state discrimination, quantum teleportation, and quantum random walks.

Connection between syntax and semantics

We prove that the metric equation introduced is sound and complete. Soundness ensures that if a metric equation $t =_\varepsilon s$ can be derived in the calculus, then the distance between the interpretations of t and s is at most ε. Completeness guarantees that if ε is the maximum distance between the interpretations of two programs, then we can derive $t =_\varepsilon s$ in the calculus.

Across all these areas, we also prove several folklore results about conditionals, such as soundness and completeness, that, to our knowledge, are missing from the literature.

1.3 Document Structure

Chapter 2 introduces (metric) λ-calculus along with its categorical interpretation. One advantage of working with a metric equational system is the ability to reason syntactically about approximate equivalence. We leverage this idea in an interlude on booleans (i.e., terms of type $\mathbb{I} \oplus \mathbb{I}$) to illustrate the usefulness of the classical equational system. In Chapter 3, we introduce a metric equation for conditionals, prove its soundness and completeness, and present a few models in this setting, along with a few illustrative syntactic examples. Then, in Chapter 4 and Chapter 5, we focus on reasoning about higher-order probabilistic and first-order quantum programs, respectively, including the necessary background in each domain.

The thesis concludes with directions for future work in Chapter 6. An overview of the categorical concepts and results employed in this thesis is provided in Appendix A.
Although this work uses knowledge across multiple areas, the author's engagement with them is mostly limited to the scope of this thesis.

Chapter 2

Metric Lambda Calculus

This chapter introduces the metric lambda calculus as presented in [1], drawing also from [23–25]. After some intuitions about (metric) lambda calculus, the chapter overviews its syntax, metric equational system, and interpretation. Our presentation on lambda calculus will involve conditionals; and in this regard, we will include proofs of results that are folklore, but whose proof we could not find in the literature. Finally, we illustrate the usefulness of the (metric) equational system by using it as a bridge to connect a certain type to Boolean algebra. For a more detailed study of lambda calculus theory, the reader is referred to *e.g.*[26]. It is worth noting that this chapter includes minor contributions such as the proofs of results on conditionals that are folklore, but whose proof we could not find in the literature and the preliminary study about Booleans.

2.1 A first look at lambda Calculus

The concept of a function emerges naturally in lambda calculus. But what exactly is a function? In most mathematics, 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.

From a Computer Science perspective, this does not always suffice. We are typically just as concerned with how a function computes its result as we are with what it produces. For instance, consider sorting: every correct sorting algorithm produces the same output for a given input, from the simplest to the most sophisticated. Yet, entire books and research papers are devoted to analyzing different sorting techniques. Clearly, something important is being overlooked. The casual use of the term “algorithm” in that context is revealing: a function should be represented not by its graph, but by the rule or process that describes how its result is computed. This view gives rise to 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 *abstractions*. A function $f : x \mapsto f(x)$ is represented as $\lambda x.f(x)$. Applying a function to an argument is done by juxtaposing the abstraction with its argument. For instance, given the function $f : x \mapsto x + 1$, the term $f(2)$ is represented by $(\lambda x.x + 1)(2)$.

A major limitation of this notation appears to be that we can only define unary functions, that is, we can introduce only one argument at a time. However, this is not a true restriction. Suppose we have a binary function represented as an expression with formal arguments x and y, say $f(x, y)$. It can be represented as $g = \lambda y.\,(\lambda x.\, f(x, y))$. This function g is equivalent to the original binary function f, but it takes its arguments *one at a time*. This idea, based on *currying*, shows how functions of multiple arguments can be represented using only unary functions.

The expression of *higher-order functions*, functions whose inputs and/or outputs are themselves functions, in a simple manner, is another important 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 rule” 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 $X \to X$. 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.

However, this remarkable feature of the lambda calculus can also lead to complications. As previously mentioned, applying a function to itself, as in the term $(\lambda x.\, x\, x)(\lambda x.\, x\, x)$, results in non-termination. The typed variant of the lambda calculus, known as the *simply-typed lambda calculus*, addresses this issue by assigning a type to every expression. Here, a function may only be applied to an argument if the argument's type is the same as the function's expected domain. Consequently, terms such as $f(f)$ are not allowed, even if f represents the identity function.

2.2 Syntax

2.2.1 Type system

As previously mentioned, 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 a set. Let G represent a set of ground types. The **Backus-Naur Form** (**BNF**) grammar of types for affine lambda calculus is as follows:

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

Note that this is an inductive definition. Ground types can be such things as booleans, integers, and so forth. The type $\mathbb{I}$ is the so-called unit. The type $\mathbb{A} \otimes \mathbb{A}$ corresponds to the tensor of two types. The type $\mathbb{A} \oplus \mathbb{A}$ can be seen as the coproduct/disjunction of two types. Finally, the type $\mathbb{A} \multimap \mathbb{A}$, for instance, in a set-theoretical perspective, can be seen as the type of functions from 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 lambda terms*. The grammar of raw lambda terms is given by the BNF below.

$$\begin{aligned} v \quad ::= \quad & x \mid f(v_1, \dots, v_n) \mid \ * \ \mid (\lambda x.v) \mid v\, w \mid v \otimes w \mid \\ & \textsf{pm}\ v\ \textsf{to}\ x \otimes y.w \mid v\ \ \textsf{to}\ \ *\,.w \mid \textsf{dis}(v) \mid \textsf{inl}(v) \mid \textsf{inr}(v) \mid \\ & \ \textsf{case}\ v\ \{\textsf{inl}(x) \Rightarrow w;\ \textsf{inr}(y) \Rightarrow u\} \end{aligned}$$

Here x ranges over an infinite set of variables, and $f \in \Sigma$, where Σ corresponds to a set of sorted operation symbols. The expression $f(v_1, \ldots, v_n)$ corresponds to the application of the function f to the arguments $v_1, \ldots, v_n$. The symbol $*$ is the unit. The term $(\lambda x.v)$ is the lambda abstraction term, representing a function that takes an argument of type x and returns the value v. The term $v\,w$ 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 that deconstructs 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 terms inl(v) and inr(v) represent the left and right injections of v, respectively. Intuitively, the case statement executes w when v is a left injection, and u when v is a right injection, and a "mixture" of both otherwise.

Convention 2.2.1.

- Applications associate to the left; that is, the expression vwu is interpreted as $(vu)u$. This convention is convenient when applying a function to multiple arguments: for example, $fxyz$ is read as $(((fx)y)z)$.
- The body of a lambda abstraction, as well as pattern matching and discarding constructs (*i.e.*, the part after the dot), extends as far to the right as possible. For instance, $\lambda x.\, vw$ is interpreted as $\lambda x.\, (vw)$, not $(\lambda x.\, v)w$.

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 a *bound* variable. 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:

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

$$FV(\text{case } v\, \{\text{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \text{inr}_{\mathbb{A}}(y) \Rightarrow u\}) = FV(v) \cup (FV(w) \backslash \{x\}) \cup (FV(u) \backslash \{y\}).$$

2.2.4 α-equivalence

A natural notion of equivalence should stem from the fact that terms that differ only in the names of their bound variables represent the same program. For instance, the functions $\lambda x.x$ and $\lambda y.y$ have the same input-output behavior, despite being represented by different lambda terms. The equivalence we are referring to is called α-*equivalence*.

Definition 2.2.2 (α-renaming)**.** The α-equivalence is an equivalence relation on lambda terms that is used to, among other things, rename bound variables (se will see that such is crucial in defining crucial operations in lambda-calculus). 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 α-equivalent, written $=_\alpha$, if one can be derived from the other by a series of changes of bound variables.

Convention 2.2.3. Terms are considered up to α-equivalence from now on, *i.e.*, terms are treated as equal if they differ only by the renaming of bound variables.

2.2.5 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. In this context, it is necessary to avoid the unintended binding of free variables. For example, consider terms $v \triangleq \lambda x.\, yx$ and $w \triangleq \lambda z.\, xz$. Note that x is bounded in v and free in w, Consequently, the term $v[w/y]$ is not the same as $\lambda x. \quad (\lambda z.\, xz)x$. The proper thing to do is to rename the bound variable *before* the substitution:

$$v[w/y] = \lambda x'.\, yx'[w/y] = \quad \lambda x'. \quad (\lambda z.\, xz)x'.$$

Thus, the operation of substitution may require renaming bound variables. In such cases, it is preferable to select a *fresh* variable—that is, a variable that has not yet been used—as the new name for the bound variable. The assumption that the set of variables is infinite ensures that a fresh variable is always available when needed.

Definition 2.2.4. Given terms v and w, the substitution $v[w/x]$ is defined below.

$$\begin{aligned} x[w/x] &= w \\ y[w/x] &= y && \text{if } x \neq y, \\ *[w/x] &= *, \end{aligned}$$

$$
\begin{aligned}
(\lambda x.v)[w/x] &= \lambda x.v, && \\
(\lambda y.v)[w/x] &= \lambda y : \mathbb{B}.v[w/x], && \text{if } x \neq y \text{ and } y \notin FV(w) \\
\lambda y.v[w/x] &= \lambda y' : \mathbb{B}.v\{y'/y\}[w/y], && \text{if } x \neq y, y \in FV(w), \\
& && \text{and } y' \text{ is fresh} \\
(v\,u)[w/x] &= v[w/x]\,u[w/x] && \\
(f(v_1,\dots,v_n))[w/x] &= f(v_1[w/x],\dots,v_n[w/x]) && \\
(v \otimes u)[w/x] &= (v[w/x] \otimes u[w/x]) && \\
(\mathsf{pm}\ v\ \mathsf{to}\ y \otimes z.u)[w/x] &= \mathsf{pm}\ v[w/x]\ \mathsf{to}\ y \otimes z.u[w/x], && \text{if } y \notin FV(w), z \notin FV(w) \\
(\mathsf{pm}\ v\ \mathsf{to}\ y \otimes z.u)[w/x] &= \mathsf{pm}\ v[w/x]\ \mathsf{to}\ y' \otimes z. && \text{if } y \in FV(w), z \notin FV(w), \\
& \quad u\{y'/y\}[w/x], && \text{and } y' \text{ is fresh} \\
(\mathsf{pm}\ v\ \mathsf{to}\ y \otimes z.u)[w/x] &= \mathsf{pm}\ v[w/x]\ \mathsf{to}\ y \otimes z'. && \text{if } y \notin FV(w), z \in FV(w), \\
& \quad u\{z'/z\}[w/x], && \text{and } z' \text{ is fresh} \\
(\mathsf{pm}\ v\ \mathsf{to}\ y \otimes z.u)[w/x] &= \mathsf{pm}\ v[w/x]\ \mathsf{to}\ y' \otimes z'. && \text{if } y \in FV(w), z \in FV(w), \\
& \quad u\{y'/y\}\{z'/z\}[w/x], && \text{and } y', z' \text{ are fresh} \\
(v\ \mathsf{to}\ *.u)[w/x] &= v[w/x]\ \mathsf{to}\ *.u[w/x] && \\
(\mathsf{inl}(v))[w/x] &= \mathsf{inl}(v[w/x]), && \\
(\mathsf{inr}(v))[w/x] &= \mathsf{inr}(v[w/x]), && \\
\mathsf{case}\ v \left\{\begin{matrix} \mathrm{inl}(y) \Rightarrow p; \\ \mathrm{inr}(z) \Rightarrow q \end{matrix}\right\}[w/x] &= \mathsf{case}\ v[w/x] \left\{\begin{matrix} \mathrm{inl}(y) \Rightarrow \\ p[w/x]; \\ \mathrm{inr}(z) \Rightarrow \\ q[w/x] \end{matrix}\right\}, && \text{if } y \notin FV(w), z \notin FV(w) \\
& (\dots) && \\
(\mathsf{case}\ v\,\{\mathrm{inl}(y) \Rightarrow p; &= \mathsf{case}\ v[w/x] && \text{if } y \in FV(w), z \in FV(w), \\
\mathrm{inr}(z) \Rightarrow q\})[w/x] & \quad \{\mathrm{inl}(y) \Rightarrow p\{y'/y\}\{z'/z\} && \text{and } y', z' \text{ are fresh} \\
& \quad [w/x]; && \\
& \quad \mathrm{inr}(z) \Rightarrow q\{y'/y\} && \\
& \quad \{z'/z\}[w/x]\} &&
\end{aligned}
$$

The sequential substitutions $v[w_1/x_1]\dots[w_n/x_n]$ are writen as $v[w_1/x_1,\dots,w_n/x_n]$.

2.2.6 Typing rules

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

Typing rules are formulated using *typing judgments*. A typing judgment is an expression of the form $x_1 : \mathbb{A}_1, \dots, x_n : \mathbb{A}_n \triangleright v : \mathbb{A}$ (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 x_i has type $\mathbb{A}_i$, for $1 \leq i \leq n$. The list $x_1 : \mathbb{A}_1, \dots, x_n : \mathbb{A}_n$ 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 $x_1, \dots, x_n$. Typing contexts are denoted by Greek letters Γ, Δ, 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 linear lambda calculus, when different contexts appear sequenced (*e.g.*$\Gamma, \Delta, \dots$) they do not share variables amongst themselves. In other words, the typing system is linear: every variable is used exactly once.

There are certain typing rules that are not explicitly stated and whose validity follows the existing rules of the system. These are called *admissible rules*. The concept of *shuffling* is employed to construct a linear typing system that ensures the admissibility of the exchange rule (which allows reordering variables within the same context) and enables unambiguous reference to judgment's interpretation denoted $[\![\Gamma \triangleright v : \mathbb{A}]\!]$. Shuffling is defined as a permutation of typed variables in a sequence of contexts, $\Gamma_1, \dots, \Gamma_n$, preserving the relative order of variables within each Γ_i [27]. For instance, if $\Gamma_1 = x : \mathbb{A}, y : \mathbb{B}$ and $\Gamma_2 = z : \mathbb{D}$, then $z : \mathbb{D}, x : \mathbb{A}, y : \mathbb{B}$ is a valid shuffle of Γ_1, Γ_2. On the other hand, $y : \mathbb{B}, x : \mathbb{A}, z : \mathbb{D}$ is not a shuffle because it alters the occurrence order of x and y in Γ_1. The set of shuffles based on $\Gamma_1, \dots, \Gamma_n$ is denoted as $\mathsf{Sf}(\Gamma_1; \dots; \Gamma_n)$. A valid typing derivation is constructed using the inductive rules shown in Figure 1.

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

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

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

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

$$\dfrac{\Gamma \triangleright v : \mathbb{A}}{\Gamma \triangleright \mathrm{inl}_{\mathbb{B}}(v) : \mathbb{A} \oplus \mathbb{B}}\ (\text{inl}) \qquad \dfrac{\Gamma \triangleright v : \mathbb{B}}{\Gamma \triangleright \mathrm{inr}_{\mathbb{A}}(v) : \mathbb{A} \oplus \mathbb{B}}\ (\text{inr})$$

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

Figure 1: Term formation rules of linear lambda calculus.

A few straightforward programming examples are provided for a better understanding of the rules.

Example 2.2.5. For instance, the program that swaps the elements of a tensor product can be written as follows:

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

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

1	$x : \mathbb{A} \otimes \mathbb{B} \triangleright y : \mathbb{A} \otimes \mathbb{B}$	(hyp)
2	$b : \mathbb{B} \triangleright b : \mathbb{B}$	(hyp)
3	$a : \mathbb{A} \triangleright a : \mathbb{A}$	(hyp)
4	$b : \mathbb{B}, a : \mathbb{A} \triangleright b \otimes a : \mathbb{B} \otimes \mathbb{A}$	$(2, 3, \otimes_i)$
5	$x : \mathbb{A} \otimes \mathbb{B} \triangleright \mathsf{pm}\ x\ \mathsf{to}\ a \otimes b.b \otimes a : \mathbb{B} \otimes \mathbb{A}$	$(1, 4, \otimes_e)$

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

Example 2.2.6. Another example is the function that recieves a tensor product of type $\mathbb{I} \otimes \mathbb{I}$ and returns first element, discarding the second:

Dis2nd $\triangleq - \triangleright \lambda x : \mathbb{I} \otimes \mathbb{I}.\, \mathsf{pm}\ x\ \mathsf{to}\ a \otimes b.b\ \ \mathsf{to}\ *\,.\,a : \mathbb{I}$

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

1 $b : \mathbb{I} \rhd b : \mathbb{I}$ (hyp)

2 $a : \mathbb{I} \rhd a : \mathbb{I}$ (hyp)

3 $a : \mathbb{I}, b : \mathbb{I} \rhd b \ \text{to} \ * \, . \, a : \mathbb{I}$ $(1, 2, \mathbb{I}_e)$

4 $x : \mathbb{I} \otimes \mathbb{I} \rhd x : \mathbb{I} \otimes \mathbb{I}$ (hyp)

5 $x : \mathbb{I} \otimes \mathbb{I} \rhd \text{pm } x \text{ to } a \otimes b . b \ \text{to} \ * \, . \, a : \mathbb{I}$ $(3, 4, \otimes_e)$

6 $- \rhd \lambda x : \mathbb{I} \otimes \mathbb{I}. \text{ pm } x \text{ to } a \otimes b . b \ \text{to} \ * \, . \, a : \mathbb{I}$ $(5, \multimap_i)$

Example 2.2.7. Next, consider the following program which can be seen as the distributive property:

$$z : (\mathbb{A} \oplus \mathbb{B}) \otimes \mathbb{D} \rhd \text{pm } z \text{ to } a \otimes b. \text{ case } a \left\{ \begin{matrix} \mathrm{inl}_{\mathbb{B}}(x) \Rightarrow \mathrm{inl}_{\mathbb{B} \otimes \mathbb{D}}(x \otimes b); \\ \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow \mathrm{inr}_{\mathbb{A} \otimes \mathbb{D}}(y \otimes b) \end{matrix} \right\} : (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})$$

To prove that this program is well-typed, we reason as follows:

1 $x : \mathbb{A} \rhd x : \mathbb{A}$ (hyp)

2 $b : \mathbb{D} \rhd x : \mathbb{D}$ (hyp)

3 $b : \mathbb{D}, x : \mathbb{A} \rhd \mathrm{inl}_{\mathbb{B} \otimes \mathbb{D}}(x \otimes b) : (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})$ $(1, 2, \text{inl})$

4 $y : \mathbb{B} \rhd x : \mathbb{B}$ (hyp)

5 $b : \mathbb{D}, y : \mathbb{B} \rhd \mathrm{inr}_{\mathbb{A} \otimes \mathbb{D}}(x \otimes b) : (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})$ $(2, 4, \text{inr})$

6 $a : \mathbb{A} \oplus \mathbb{B} \rhd a : \mathbb{A} \otimes \mathbb{B}$ (hyp)

7 $a : \mathbb{A} \oplus \mathbb{B}, b : \mathbb{D} \rhd \text{ case } a \left\{ \begin{matrix} \mathrm{inl}_{\mathbb{B}}(x) \Rightarrow \mathrm{inl}_{\mathbb{B} \otimes \mathbb{D}}(x \otimes b); \\ \mathrm{inr}_{\mathbb{B}}(y) \Rightarrow \mathrm{inr}_{\mathbb{A} \otimes \mathbb{D}}(x \otimes b) \end{matrix} \right\} : (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})$ $(6, 3, 5, \text{case})$

8 $z : (\mathbb{A} \oplus \mathbb{B}) \otimes \mathbb{D} \rhd z : (\mathbb{A} \oplus \mathbb{B}) \otimes \mathbb{D}$ (hyp)

9 $z : (\mathbb{A} \oplus \mathbb{B}) \otimes \mathbb{D} \rhd \text{pm } z \text{ to } a \otimes b. \text{ case } a \ \{\ldots\} : (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})$ $(8, 7, \otimes_e)$

It should be noted that there are two distinct conventions for typing terms. One is the Church-style typing, in which all subterms are explicitly typed. This is the convention we adopt. The other is the Curry-style typing, where only the outermost term is assigned a type, and the types of subterms are left implicit. For instance, consider the Curry-style typing judgment:

$$x : \mathbb{A} \rhd (\lambda f. \, x)(\lambda y. \, y) : \mathbb{A}.$$

Here, the variable x has an explicitly assigned type, but the variable y does not. Its type is not constrained and could be anything. The consequence is that a typed term alone does not uniquely determine its typing derivation.

Convention 2.2.8.

- A judgment $\Gamma \triangleright v : \mathbb{A}$ will often be abbreviated into $\Gamma \triangleright v$ or even just v when no ambiguities arise.

- The type annotations in terms $\Gamma \triangleright \lambda x : \mathbb{A}.\, v$, $\Gamma \triangleright \mathrm{inl}_{\mathbb{B}}(v)$ and $\Gamma \triangleright \mathrm{inr}_{\mathbb{A}}(v)$ will also often be ommited when no ambiguities arise.

2.2.7 Properties

The calculus defined in Figure 1 possesses several desirable properties, which are listed below. Before detailing them, it is necessary to introduce some auxiliary notation. Given a context Γ, $te(\Gamma)$ denotes context Γ with all types erased. The expression $\Gamma \simeq_\pi \Gamma'$ means contexts Γ is a permutation of context Γ'. This notation also applies to non-repetitive lists of untyped variables $te(\Gamma)$.

Theorem 2.2.9. *The lambda calculus defined by the rules of Figure 1 has the following properties:*

1. *for all judgements $\Gamma \triangleright v$ and $\Gamma' \triangleright v$, te(Γ) $\simeq_\pi$ te(Γ');*

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. Since these properties are established in [28, Theorem 2.3] for the lambda calculus without conditionals, it suffices to consider the cases involving conditionals. It follows in all three cases from induction over the length of judgement derivation trees.

Let us focus first on Property (1). The case of the rules concerning injections is direct. As for rule (case) take two contexts E and E' for the same conditional. According to this rule we obtain contexts Γ, Γ', Δ, Δ' such that $E \in \mathrm{Sf}(\Gamma; \Delta)$ and $E' \in \mathrm{Sf}(\Gamma'; \Delta')$. It follows from induction that $\mathrm{te}(\Gamma) \simeq_\pi \mathrm{te}(\Gamma')$ and $\mathrm{te}(\Delta) \simeq_\pi \mathrm{te}(\Delta')$, and the proof is then obtained from the sequence of equivalences,

$$\begin{aligned}\text{te}(E) &\simeq_\pi \text{te}(\Gamma, \Delta)\\ &\simeq_\pi \text{te}(\Gamma', \Delta')\\ &\simeq_\pi \text{te}(E')\end{aligned}$$

Concerning Property (2), the case of the rules concerning injections is direct and the case of rule (case) is a corollary of Property (1). Finally let us consider Property (3). Again the case concerning injections is direct and we thus focus only on rule (case). According to this rule we obtain contexts Γ, Γ', Δ, Δ' such that $E \in \mathrm{Sf}(\Gamma; \Delta)$ and $E \in \mathrm{Sf}(\Gamma'; \Delta')$. By an appeal to Property (1) we also obtain $\Gamma \simeq_\pi \Gamma'$ and $\Delta \simeq_\pi \Delta'$, and thus since shuffling preserves relative orders we obtain $\Gamma = \Gamma'$ and $\Delta = \Delta'$. The proof then follows by induction. □

Lemma 2.2.10 (Exchange and Substitution)**.** *For every judgement* $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \rhd v : \mathbb{D}$ *the judgement* $\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \rhd v : \mathbb{D}$ *is derivable. Not only this, given judgements* $\Gamma, x : \mathbb{A} \rhd v : \mathbb{B}$ *and* $\Delta \rhd w : \mathbb{A}$ *the judgement* $\Gamma, \Delta \rhd v[w/x] : \mathbb{B}$ *is also derivable.*

Proof. Once again, these properties are established in [1, Theorem 2.1] for the lambda calculus without conditionals, so it suffices to consider the cases involving conditionals.

We start with the exchange property which follows by induction over the length of derivation trees. The rules that involve injections are direct. The rule (case) calls for case distinction, more specifically we distinguish between the cases in which both variables ($x : \mathbb{A}$ and $y : \mathbb{B}$) are in Γ, both are in Δ, and otherwise. The first two cases follow straightforwardly by induction and the definition of a shuffle. For the third case consider a judgement $E_1, x : \mathbb{A}, y : \mathbb{B}, E_2 \rhd \text{case}\ v \,\{\mathrm{inl}_{\mathbb{F}}(a) \Rightarrow w;\ \mathrm{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{D}$, and assume with no loss of generality that Γ is of the form $\Gamma_1, x : \mathbb{A}, \Gamma_2$ and Δ of the form $\Delta_1, y : \mathbb{B}, \Delta_2$. The proof now follows directly from the implication,

$$\begin{aligned}&E_1, x : \mathbb{A}, y : \mathbb{B}, E_2 \in \mathrm{Sf}(\Gamma_1, x : \mathbb{A}, \Gamma_2;\ \Delta_1, y : \mathbb{B}, \Delta_2) \Longrightarrow\\ &E_1, y : \mathbb{B}, x : \mathbb{A}, E_2 \in \mathrm{Sf}(\Gamma_1, x : \mathbb{A}, \Gamma_2;\ \Delta_1, y : \mathbb{B}, \Delta_2)\end{aligned}$$

(which holds by the definition of a shuffle).

Finally we now focus on the substitution rule which also follows by induction over the length of judgement derivation trees. Again the cases involving the injections are direct, and we thus only detail the proof of rule (case). Consider then judgements $E, x : \mathbb{A} \rhd \text{case}\ v \,\{\mathrm{inl}_{\mathbb{D}}(a) \Rightarrow$

$w;\ \mathrm{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{B}$ and $Z \triangleright t : \mathbb{A}$ with $E \in \mathrm{Sf}(\Gamma; \Delta)$. According to the definition of a shuffle either Γ is of the form $\Gamma_1, x : \mathbb{A}$ or Δ is of the form $\Delta_1, x : \mathbb{A}$. The first case follows directly and the second case is a corollary of the exchange rule. □

Convention 2.2.11. Given programs **A** $\triangleq \Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}$ and **B** $\triangleq \Delta \triangleright w : \mathbb{A}$, we will often abuse notation by writing $\Gamma, \Delta \triangleright$ **A**[**B**$/x] : \mathbb{B}$ to mean $\Gamma, \Delta \triangleq v[w/x] : \mathbb{B}$. Variants such as simply writing **A**[**B**$/x]$ to refer to the term $v[w/x]$ will also be used.

2.2.8 Equations-in-context

Ver historia de reduções vs eqs

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 equations which fall into two primary categories: the β-*equations*, which intuitively perform operations and enforce the intended meaning of the term, and η-*equations*, which simplify terms by exploiting extensionality. There is also a secondary class of equations known as *commuting conversions*, which serve to disambiguate terms that, while equivalent, have different representations. As a result, affine λ-calculus comes equipped with the so-called equations-in-context $\Gamma \triangleright v = w : \mathbb{A}$, which are often abbreviated as $v = w : \mathbb{A}$, or simply $v = w$ when the type is clear from context. These equations are illustrated in Figure 2.

(β)	$(\lambda x : \mathbb{A}.\, v)\, w = v[w/x]$	(η)	$\lambda x : \mathbb{A}.\, (v\, x) = v$
$(\beta_{\mathbb{I}_e})$	$*$ to $*\,.\, v = v$	$(\eta_{\mathbb{I}_e})$	v to $*\,.\, w[*/z] = w[v/z]$
$(\beta_{\otimes_e})$	pm $v \otimes w$ to $x \otimes y.\, u = u[v/x, w/y]$		
$(\eta_{\otimes_e})$	pm v to $x \otimes y.\, u[x \otimes y/z] = u[v/z]$		
$(c_{\mathbb{I}_e})$	$u[v$ to $*\,.w/z] = v$ to $*\,.u[w/z]$		
$(c_{\otimes_e})$	$u[$pm v to $x \otimes y.\, w/z] =$ pm v to $x \otimes y.\, u[w/z]$		
(β^{inl}_{case})	case $\mathrm{inl}_{\mathbb{B}}(v)\ \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow u\} = w[v/x]$		
(β^{inr}_{case})	case $\mathrm{inl}_{\mathbb{B}}(v)\ \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow u\} = u[v/y]$		
(η_{case})	case v $\{$inl$_{\mathbb{B}}(y) \Rightarrow w[inl_{\mathbb{B}}(y)/x];$ inr$_{\mathbb{A}}(z) \Rightarrow w[inr_{\mathbb{A}}(z)/x]\} = w[v/x]$		

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

It is evident that, for example, equation (β) enforces the meaning of application in $(\lambda x : \mathbb{A}.$

$v)w$, which is interpreted as "v with w in place of x". On the other hand, the equation (η) is a simplification rule exploring extensionality: it states that a function that applies another function v to an argument x can be simplified to the function v itself. The remaining β e η equations follow similar reasoning.

The following example demonstrates how these equations can be used in practice.

Example 2.2.12. For instance, consider a program that receives a tensor of terms whose second component is $*$ and discards it. This program can be simplified to the term corresponding to the first component of the tensor. In other words, we will show that the λ-term

$$- \triangleright \big(\lambda z : \mathbb{A} \otimes \mathbb{I}.\, \mathsf{pm}\ z\ \mathsf{to}\ x \otimes y.\, y\ \mathsf{to}\ *\, .\, x\big)\ (v \otimes *) : \mathbb{A}$$

can be simplified to $v : \mathbb{A}$.

Applying equation β , we have:

$$- \triangleright \big(\lambda z : \mathbb{A} \otimes \mathbb{I}.\, \mathsf{pm}\ z\ \mathsf{to}\ x \otimes y.\, y\ \mathsf{to}\ *\, .\, x\big)\ (v \otimes *) = \mathsf{pm}\ v \otimes\ *\ \mathsf{to}\ x \otimes y.\, y\ \mathsf{to}\ *\, .\, x : \mathbb{A}.$$

Next, applying equation $\beta_{\otimes_e}$, it follows:

$$\mathsf{pm}\ v \otimes\ *\ \mathsf{to}\ x \otimes y.\, y\ \mathsf{to}\ *\, .\, x = *\ \mathsf{to}\ *\, .\, v : \mathbb{A}$$

Finally, applying equation $\beta_{\mathbb{I}_e}$, we have:

$$*\ \mathsf{to}\ *\, .\, v = v : \mathbb{A}$$

Definition 2.2.13. Consider a pair (G, Σ), where G is a set of ground types and Σ is a set of sorted operation symbols. A *linear λ-theory* is a triple $((G, \Sigma), Ax)$, where Ax is a set of equations-in-context over λ-terms constructed from (G, Σ). The elements of Ax are called the *axioms* of the theory.

Let $Th(Ax)$ denote the smallest be the smallest congruence that containing Ax, the equations presented in Figure 2, and closed under exchange and substitution (Lemma 2.2.10). The elements of $Th(Ax)$ are called the *theorems* of the theory.

We will often denote the triple $((G, \Sigma), Ax)$ by T when referring to a linear λ-theory.

For instance recall Example 2.2.12:

$$- \triangleright \big(\lambda z : \mathbb{I} \otimes \mathbb{A}.\, \mathsf{pm}\ z\ \mathsf{to}\ x \otimes y.\, x\ \mathsf{to}\ *\, .\, y\big)\ (* \otimes v) = v : \mathbb{A}$$

is a theorem.

2.2.9 Interlude: Booleans - Part 1

In this subsection, we illustrate the usefulness of the classical equational system by showing how it can be used to connect the type $\mathbb{I} \oplus \mathbb{I}$ to Boolean algebra. More precisely, we use the previously introduced calculus to write programs corresponding to Boolean operations such as conjunction, disjunction, and negation. We then use the extensionality of the copairing and the equations-in-context to demonstrate that these operations satisfy the properties required by Boolean algebra.

The type $\mathbb{I} \oplus \mathbb{I}$ can be used to represent truth-values, in which case True $= \mathrm{inl}(*)$, False $= \mathrm{inr}(*)$ [25]. We will use the equations in Figure 2 to demonstrate that they possess certain properties typical of Boolean algebras.

Boolean operators

As a part of our λ-theory we consider an operation dis $: \mathbb{I} \oplus \mathbb{I} \to \mathbb{I}$ which discards its input, accompanied by the following axiom which we will denote by ax_{dis},

$$\mathsf{dis}(v) = \mathsf{case}\ v \left\{ \begin{matrix} \mathrm{inl}_{\mathbb{I}}(x) \Rightarrow \mathrm{inl}_{\mathbb{I}}(x); \\ \mathrm{inr}_{\mathbb{I}}(y) \Rightarrow \mathrm{inr}_{\mathbb{I}}(y) \end{matrix} \right\}.$$

Given variables $a : \mathbb{I} \oplus \mathbb{I}$ and $b : \mathbb{I} \oplus \mathbb{I}$, their conjunction and disjunction correspond to the following programs:

$$\textbf{Conjunction}\ (a, b) \triangleq a : \mathbb{I} \oplus \mathbb{I}, b : \mathbb{I} \oplus \mathbb{I} \triangleright \mathsf{case}\ a \left\{ \begin{matrix} \mathrm{inl}_{\mathbb{I}}(x) \Rightarrow x\ \mathsf{to}\ * .\ b; \\ \mathrm{inr}_{\mathbb{I}}(y) \Rightarrow y\ \mathsf{to}\ * .\ \mathsf{dis}(b)\ \mathsf{to}\ * .\mathrm{inr}_{\mathbb{I}}(*) \end{matrix} \right\}$$

$$\textbf{Disjunction}\ (a, b) \triangleq v : \mathbb{I} \oplus \mathbb{I}, w : \mathbb{I} \oplus \mathbb{I} \triangleright \mathsf{case}\ a \left\{ \begin{matrix} \mathrm{inl}_{\mathbb{I}}(x) \Rightarrow x\ \mathsf{to}\ * .\ \mathsf{dis}(b)\ \mathsf{to}\ * .\mathrm{inl}_{\mathbb{I}}(*); \\ \mathrm{inr}_{\mathbb{I}}(y) \Rightarrow y\ \mathsf{to}\ * .\ b \end{matrix} \right\}$$

Moreover, negation can be expressed by the following program:

$$\textbf{Negation}\ (a) \triangleq a : \mathbb{I} \oplus \mathbb{I} \triangleright \mathsf{case}\ a \left\{ \begin{matrix} \mathrm{inl}_{\mathbb{I}}(x) \Rightarrow x; \\ \mathrm{inr}_{\mathbb{I}}(y) \Rightarrow y \end{matrix} \right\}$$

To simplify notation, given terms $\Gamma \triangleright v :: \mathbb{I} \oplus \mathbb{I}$ and $\Delta \triangleright w : \mathbb{I} \oplus \mathbb{I}$, we define:

$$\begin{aligned} &\Gamma, \Delta \triangleright \textbf{Conjunction}\ (v, w) \triangleq \Gamma, \Delta \triangleright \textbf{Conjunction}\ (a, b)[v/a, w/b] \\ &\Gamma, \Delta \triangleright \textbf{Disjunction}\ (v, w) \triangleq \Gamma, \Delta \triangleright \textbf{Disjunction}\ (a, b)[v/a, w/b] \\ &\Gamma \triangleright \textbf{Negation}\ (a) \triangleq \Gamma \triangleright \textbf{Negation}\ (a)[v/a] \end{aligned} \tag{2.2}$$

This result will enable us to verify that the programs we define satisfy the desired properties in a systematic and straightforward manner.

Extensionality of the copairing

A λ-abstraction that receives inputs of a disjunctive type is determined by what it does to inputs "from the left and from the right", *i.e.*,

$$\begin{cases} (\lambda x.v)\,\mathsf{inl}(y) = (\lambda x.w)\,\mathsf{inl}(y) \\ (\lambda x.v)\,\mathsf{inr}(z) = (\lambda x.w)\,\mathsf{inr}(z) \end{cases} \implies \lambda x.v = \lambda x.w$$

The proof is as follows:
Using the β-equation we have

$$\begin{cases} (\lambda x.v)\,\mathsf{inl}(y) = (\lambda x.w)\,\mathsf{inl}(y) \\ (\lambda x.v)\,\mathsf{inr}(z) = (\lambda x.w)\,\mathsf{inr}(z) \end{cases} = \begin{cases} v\,[\mathsf{inl}(y)/x] = w\,[\mathsf{inl}(y)/x] \\ v\,[\mathsf{inr}(z)/x] = w\,[\mathsf{inr}(z)/x] \end{cases} \tag{2.3}$$

Next, considering the equations above and η_{case} we reason as follows:

$$\begin{aligned} v &= \mathsf{case}\ x \left\{ \mathrm{inl}(y) \Rightarrow v\,[\mathsf{inl}(y)/x]; \mathrm{inl}(z) \Rightarrow v\,[\mathsf{inr}(z)/x] \right\} && (\eta_{case}) \\ &= \mathsf{case}\ x \left\{ \mathrm{inl}(y) \Rightarrow w\,[\mathsf{inl}(y)/x]; \mathrm{inl}(z) \Rightarrow w\,[\mathsf{inr}(z)/x] \right\} = w && (\text{Equation 2.3}, \eta_{case}) \end{aligned}$$

Finally, we derive $\lambda x.v = \lambda x.w$ from the conjunture.

Properties

Next, we will show that the programs we have defined verify certain properties of their namesake operations in Boolean algebra. Given we have just established the extensionality of the copairing, it follows that if the desired properties hold for the injections, then they also hold for any terms of type $\mathbb{I} \oplus \mathbb{I}$.

Lemma 2.2.14. $\mathrm{inl}(*)$ *acts as the neutral element for conjunction, whereas* $\mathrm{inr}(*)$ *serves as the absorbing element,* i.e.*, for* $\Gamma \triangleright v : \mathbb{I} \oplus \mathbb{I}$

$$\begin{cases} \Gamma \triangleright \textbf{\textit{Conjunction}}\,(\mathrm{inl}(*), v) = v \\ \Gamma \triangleright \textbf{\textit{Conjunction}}\,(\mathrm{inr}(*), v) = \textit{dis}(v)\ \textit{to}\ *.\mathrm{inr}(*) \end{cases}$$

Proof. These properties follow from the equations β^{inl}_{case}, β^{inr}_{case}, and $\beta_{\mathbb{I}_e}$.

$$\begin{aligned}
&\textbf{Conjunction}\,(\mathrm{inl}(*), v)\\
&\triangleq \textsf{case}\ \mathrm{inl}(*)\ \{\mathrm{inl}(x) \Rightarrow x\ \textsf{to}\ *\,.\,v;\ \mathrm{inr}(y) \Rightarrow y\ \textsf{to}\ *\,.\,\textsf{dis}(w')\ \textsf{to}\ *\,.\mathrm{inr}(*)\}\\
&= *\ \textsf{to}\ *\,.\,v && (\beta^{inl}_{case})\\
&= v && (\beta_{\mathbb{I}_e})
\end{aligned}$$

For the second equality, by the extensionality of the coparing we need to prove that

$$\begin{cases}\textbf{Conjunction}\,(\mathrm{inr}(*), \mathrm{inl}(z)) = \mathrm{inr}(*)\\ \textbf{Conjunction}\,(\mathrm{inr}(*), \mathrm{inr}(z)) = \mathrm{inr}(*)\end{cases}$$

For the first equation, we reason as follows:

$$\begin{aligned}
&\textbf{Conjunction}\,(\mathrm{inr}(*), \mathrm{inl}(z))\\
&\triangleq \textsf{case}\ \mathrm{inr}(*)\ \begin{Bmatrix}\mathrm{inl}(x) \Rightarrow x\ \textsf{to}\ *\,.\,\mathrm{inl}(z);\\ \mathrm{inr}(y) \Rightarrow y\ \textsf{to}\ *\,.\,\textsf{dis}(\mathrm{inl}(z))\ \textsf{to}\ *\,.\mathrm{inr}(*)\end{Bmatrix}\\
&= *\ \textsf{to}\ *\,.\,\textsf{dis}(\mathrm{inl}(z))\ \textsf{to}\ *\,.\mathrm{inr}(*) && (\beta^{inr}_{case})\\
&= \textsf{dis}(\mathrm{inl}(z))\ \textsf{to}\ *\,.\mathrm{inr}(*) && (\beta_{\mathbb{I}_e})
\end{aligned}$$

The second equation is obtained through similar reasoning. □

Note that the idempotency property of conjunction, — that is, **Conjunction** $(\mathrm{inl}(*), \mathrm{inl}(*)) = \mathrm{inl}(*)$ and **Conjunction** $(\mathrm{inr}(*), \mathrm{inr}(*)) = \mathrm{inr}(*)$ — follows directly from the equalities above.

Proposition 2.2.15. *The conjunction of two terms is commutative ,* i.e.*, for* $\Gamma \triangleright v : \mathbb{I} \oplus \mathbb{I}$ *and* $\Delta \triangleright w : \mathbb{I} \oplus \mathbb{I}$

$$\Gamma, \Delta \triangleright \textbf{\textit{Conjunction}}\,(v, w) = \textbf{\textit{Conjunction}}\,(w, v)$$

Proof. Once again, by the extensionality of copairing, it suffices to prove the equality for the four base cases:

$$\begin{cases}
\textbf{Conjunction}\,(\mathrm{inl}(c), \mathrm{inl}(d)) = \textbf{Conjunction}\,(\mathrm{inl}(d), \mathrm{inl}(c))\\
\textbf{Conjunction}\,(\mathrm{inl}(c), \mathrm{inr}(d)) = \textbf{Conjunction}\,(\mathrm{inr}(d), \mathrm{inl}(c))\\
\textbf{Conjunction}\,(\mathrm{inr}(c), \mathrm{inl}(d)) = \textbf{Conjunction}\,(\mathrm{inl}(d), \mathrm{inr}(c))\\
\textbf{Conjunction}\,(\mathrm{inr}(c), \mathrm{inr}(d)) = \textbf{Conjunction}\,(\mathrm{inr}(d), \mathrm{inr}(c))
\end{cases}$$

These equalities follow from the equations β^{inl}_{case}, β^{inr}_{case}, $\eta_{\mathbb{I}_e}$, and ax_{dis}. We will explicitly prove the second equality below; the others follow by similar reasoning.

$$
\begin{aligned}
&\textbf{Conjunction}\,(\mathrm{inl}(c), \mathrm{inr}(d)) \\
&\triangleq \mathsf{case}\ \mathrm{inl}(c) \left\{ \begin{array}{l} \mathrm{inl}(x) \Rightarrow x\ \mathsf{to}\ *\,.\,\mathrm{inr}(d); \\ \mathrm{inr}(y) \Rightarrow y\ \mathsf{to}\ *\,.\,\mathsf{dis}(\mathrm{inr}(d))\ \mathsf{to}\ * .\mathrm{inr}(*) \end{array} \right\} \\
&= c\ \mathsf{to}\ *\,.\,\mathrm{inr}(*) && (\beta^{inl}_{case}) \\
&= \mathsf{dis}(\mathrm{inl}(c))\ \mathsf{to}\ * .\mathrm{inr}(*) && (ax_{\mathsf{dis}}, \beta^{inl}_{case}) \\
&= d\ \mathsf{to}\ *\,.\,\mathsf{dis}(\mathrm{inl}(c))\ \mathsf{to}\ * .\mathrm{inr}(*) && (\eta_{\mathbb{I}_e}) \\
&= \mathsf{case}\ \mathrm{inr}(d) \left\{ \begin{array}{l} \mathrm{inl}(x) \Rightarrow x\ \mathsf{to}\ *\,.\,\mathrm{inl}(c); \\ \mathrm{inr}(y) \Rightarrow y\ \mathsf{to}\ *\,.\,\mathsf{dis}(\mathrm{inl}(c))\ \mathsf{to}\ * .\mathrm{inr}(*) \end{array} \right\} && (\beta^{inr}_{case}) \\
&\triangleq \textbf{Conjunction}\,(\mathrm{inr}(d), \mathrm{inl}(c))
\end{aligned}
$$

□

Isto vai sair daqui e ir para a prova de soundeness classica

Proposition 2.2.16 (*Syntactic fusion law*)**.** *The following equality holds:*

$$v\left[(\mathit{case}\ a\ \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow u\})\,/z\right] = \mathit{case}\ a\ \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow v[w/z];\ \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow v[u/z]\}.$$

Proof. This equality follows from the extensionality of copairing and equations $\beta^{\mathsf{inl}}_{\mathsf{case}}$, and $\beta^{\mathsf{inr}}_{\mathsf{case}}$. By the extensionality of copairing, it suffices to prove the equility for inl(b) and inl(b). We present the explicit proof for inl(c); the proof for inl(d) follows similar reasoning.

$$
\begin{aligned}
&v\left[(\mathsf{case}\ \mathsf{inl}(b)\ \{\mathrm{inl}(x) \Rightarrow w;\ \mathrm{inr}(y) \Rightarrow u\})\,/z\right] \\
&= v[w[b/x]\,/z] && (\beta^{\mathsf{inl}}_{\mathsf{case}}) \\
&= v[[w/z]\,[b/x]] \\
&= \mathsf{case}\ \mathsf{inl}(b)\ \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow v[w/z];\ \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow v[u/z]\} && (\beta^{\mathsf{inl}}_{\mathsf{case}})
\end{aligned}
$$

□

Lemma 2.2.17. *The double negation of a term w is equivalent to that term,* i.e., *for* $\Gamma \triangleright v : \mathbb{I} \oplus \mathbb{I}$

$$\Gamma \triangleright \boldsymbol{\mathit{Negation}}(\boldsymbol{\mathit{Negation}}(v)) = v.$$

Proof. By the extensionality of copairing, it suffices to prove the equality for the base cases, *i.e.*, the injections. This follows directly from the equations β^{inl}_{case} and β^{inr}_{case}. Once again, we will prove one of the resulting equalities explicitly; the other follows by similar reasoning.

$$\begin{aligned}
&\textbf{Negation}(\textbf{Negation}(\mathrm{inl}(z))) \\
&\triangleq \textbf{Negation}(\textsf{case}\ \mathrm{inl}(z)\ \{\mathrm{inl}(x) \Rightarrow \mathrm{inr}(x);\ \mathrm{inr}(y) \Rightarrow \mathrm{inl}(y)\}) \\
&= \textbf{Negation}(\mathrm{inr}(z)) && (\beta^{inl}_{case}) \\
&= \mathrm{inl}(z) && (\beta^{inr}_{case})
\end{aligned}$$

□

Lemma 2.2.18. *De Morgan's laws hold for terms* $v : \mathbb{I} \oplus \mathbb{I}$ *and* $w : \mathbb{I} \oplus \mathbb{I}$, i.e., *for* $\Gamma \triangleright v : \mathbb{I} \oplus \mathbb{I}$ *and* $\Delta \triangleright w : \mathbb{I} \oplus \mathbb{I}$

$$\Gamma, \Delta \triangleright \textbf{\textit{Disjunction}}(\textbf{\textit{Negation}}(v), \textbf{\textit{Negation}}(w)) = \textbf{\textit{Negation}}(\textbf{\textit{Conjunction}}(v, w))$$

Proof. Once again, by the extensionality of copairing, it suffices to prove the equality for the four base cases. The corresponding equalities follow from the equations β^{inl}_{case}, β^{inr}_{case} and $c_{\mathbb{I}_e}$. We will explicitly prove one of the equalities below; the others follow by similar reasoning.

$$\begin{aligned}
&\textbf{Disjunction}(\textbf{Negation}(\mathrm{inl}(c)), \textbf{Negation}, (\mathrm{inr}(d))) \\
&\triangleq \textsf{case}\ \ \textsf{case}\ \mathrm{inl}(c) \left\{\begin{matrix} \mathrm{inl}(a) \Rightarrow \mathrm{inr}(a); \\ \mathrm{inl}(b) \Rightarrow \mathrm{inl}(b) \end{matrix}\right\} \left\{\begin{matrix} \mathrm{inl}(x) \Rightarrow \ldots; \\ \mathrm{inl}(y) \Rightarrow \ldots \end{matrix}\right\} \\
&= c\ \textsf{to}\ * \,.\, \textsf{case}\ \mathrm{inr}(d) \left\{\begin{matrix} \mathrm{inl}(x) \Rightarrow \mathrm{inr}(a); \\ \mathrm{inl}(y) \Rightarrow \mathrm{inl}(b) \end{matrix}\right\} && (\beta^{inl}_{case}, \beta^{inr}_{case}) \\
&= \textsf{case}\ c\ \textsf{to}\ * .\mathrm{inr}(d) \left\{\begin{matrix} \mathrm{inl}(a) \Rightarrow \mathrm{inr}(a); \\ \mathrm{inl}(b) \Rightarrow \mathrm{inl}(b) \end{matrix}\right\} && (c_{\mathbb{I}_e}) \\
&= \textsf{case}\ \textsf{case}\ \mathrm{inl}(c) \left\{\begin{matrix} \mathrm{inl}(x) \Rightarrow x\ \textsf{to}\ * \,.\, \mathrm{inr}(d); \\ \mathrm{inr}(y) \Rightarrow y\ \textsf{to}\ * \,.\, \textsf{dis}(\mathrm{inr}(d))\ \textsf{to}\ * .\mathrm{inr}(*) \end{matrix}\right\} \left\{\begin{matrix} \ldots; \\ \ldots \end{matrix}\right\} && (\beta^{inl}_{case}) \\
&\triangleq \textbf{Negation}(\textbf{Conjunction}(\mathrm{inl}(c), \mathrm{inr}(d)))
\end{aligned}$$

□

The remaining properties such can be proven through similar reasoning.

2.2.10 Metric equational system

Metric equations [14, 15] are a strong candidate for reasoning about approximate program equivalence. These equations take the form of $\Gamma \triangleright t =_\epsilon s$, meaning they are *at most* at a distance ε from each other. The metric equational system for linear lambda calculus is depicted in Figure 3. Note that the equations $\Gamma \triangleright v = w : A$ in Figure 2, which in this setting abbreviate $\Gamma \triangleright w =_0 v : A$, are also part of the metric equational system.

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

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

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

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

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

Figure 3: Metric equational system

Here, perm(Γ) denotes the set of possible permutations of context Γ. The rules (refl), (trans), and (sym) generalize the properties of reflexivity, transitivity, and symmetry of equality. The rule (weak) asserts that if two terms are at most at a distance q from each other, then they are also at most at a distance r for 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, then one can safely assume the minimum of these distances. In particular, it is always the case that $v =_\infty w$. Rule (sym) 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 $q + r$, *i.e.*, the distances compound additively. The remaining rules follow similar reasoning.

Example 2.2.19. To ilustrate the usefulness of these equations, consider the program P that receives a tensor product, swaps its elements and then applies a function $f : \mathbb{A} \to \mathbb{D} \in \Sigma$ to the new second element of the tensor pair:

$$\textbf{SwapTensorf} \triangleq x : \mathbb{A}, y : \mathbb{B} \triangleright \mathsf{pm}\ x \otimes y\ \mathsf{to}\ a \otimes b.b \otimes f(a) : \mathbb{B} \otimes \mathbb{D}$$

Let f^{ε} be an erroneous implementation of f. The program above is thus rewritten as:

$$\textbf{SwapTensorf}^{\varepsilon} \triangleq x : \mathbb{A}, y : \mathbb{B} \triangleright \mathsf{pm}\ x \otimes y\ \mathsf{to}\ a \otimes b.b \otimes f(a)^{\varepsilon} : \mathbb{B} \otimes \mathbb{D}$$

Consider we have the axiom $f^{\varepsilon}(a) =_{\varepsilon} f(a)$. Then it is possible to show that **SwapTensorf**$^{\varepsilon} =_{\varepsilon}$ **SwapTensorf** using our metric equational system. The proof is as follows. The types and contexts are omitted for brevity as no ambiguity arises.

1 $f(a)^{\varepsilon} =_{\varepsilon} f(a)$

2 $b =_0 b$ (refl)

3 $b \otimes f(a)^{\varepsilon} =_{\varepsilon} b \otimes f(a)$ $(1, 2, \otimes_i)$

4 $x \otimes y =_0 x \otimes y$ (refl)

5 pm $x \otimes y$ to $a \otimes b.b \otimes f(a)^{\varepsilon} =_{\varepsilon}$ pm $x \otimes y$ to $a \otimes b.b \otimes f(a)$ $(3, 4, \otimes_e)$

Definition 2.2.20. Consider a tuple (G, Σ), where G is a set of ground types and Σ is a set of sorted operation symbols of the form $f : A_1, \ldots, A_n \to A$ with $n \geq 1$. A *metric λ-theory* is a tuple $((G, \Sigma), Ax)$, where Ax is a set of *metric equations-in-context* over λ-terms constructed from (G, Σ).

The elements of Ax are called the *axioms* of the theory. Let $Th(Ax)$ denote the smallest class that contains Ax and is closed under the rules presented in Figure 2 (i.e., the classical equational system) and Figure 3. The elements of $Th(Ax)$ are called the *theorems* of the theory.

For instance, in Example 2.2.19, **SwapTensorf** $=_{\varepsilon}$ **SwapTensorf**$^{\varepsilon}$ is a theorem.

2.2.11 Interlude: Booleans - Part 2

We can now use the extended system to explore the booleans introduced in Section 2.2.9. For instance, given axioms

$$\Gamma \triangleright v =_{\varepsilon} v' \quad \text{and} \quad \Delta \triangleright w =_{\delta} w',$$

we can ask whether

$$\textbf{Conjunction}[v, w] =_{\varepsilon+\delta} \textbf{Conjunction}[v', w'],$$

which indeed follows from a double application of the substitution rule. An analogous reasoning applies to **Conjunction**$[v, w]$ and **Negation**$[w]$.

These derivations we have just established are interesting, for they hint at "quantitative laws" for the boolean connectives. For example, previously we have established that

$$\begin{cases} \Gamma \triangleright \textbf{Conjunction}\ (\mathrm{inl}(*), w) = w \\ \Gamma \triangleright \textbf{Conjunction}\ (\mathrm{inr}(*), w) = \mathsf{dis}(w) \text{ to } * .\mathrm{inr}(*). \end{cases}$$

Now we can assume the existence of a truth value $- \triangleright v : \mathbb{I} \oplus \mathbb{I}$ between "true" and "false", *i.e.*, $\mathrm{inr}(*) =_{\epsilon} v =_{\delta} \mathrm{inl}(*)$ and derive the quantitative laws

$$\begin{cases} \Gamma \triangleright \textbf{Conjunction}\ (v, w) =_{\delta} w \\ \Gamma \triangleright \textbf{Conjunction}\ (v, w) =_{\epsilon} \mathsf{dis}(w) \text{ to } * .\mathrm{inr}(*). \end{cases}$$

2.3 Semantics

Up to this point, we have discussed λ-calculus in abstract terms: we explored which programs can be written, but we have not yet assigned them any meaning. This process—assigning meaning to syntactic expressions—is known as the *interpretation* or *semantics* of the language. In fact, the word "semantics" comes from the Greek word for "meaning".

There are different kinds of semantics, in particular, *denotational semantics* interprets terms as mathematical objects. This is done by defining a function that maps syntactic entities (such as types and terms) to semantic entities (such as sets and functions). This mapping is called the *interpretation function*, typically denoted by $[\![-]\!]$. Thus, given a term v, we write $[\![v]\!]$ to denote its meaning under a specific interpretation.

Naturally, this raises important questions: what guarantees that the interpretation of terms respects calculus's classical equations? This leads us to the notions of *soundness* and *completeness*.

With respect to a given class of interpretations:

- *Soundness* is the property

$$v = w \ \Rightarrow\ [\![v]\!] = [\![w]\!] \quad \text{for all interpretations in the class.}$$

That is, if two terms are provably equal, then they are interpreted as equal.

- *Completeness* is the property

$$[\![v]\!] = [\![w]\!] \;\Rightarrow\; v = w \quad \text{for all interpretations in the class.}$$

That is, if two terms are interpreted as equal, then they are provably equal.

Soundness ensures that our equations are *correct*—all derivable equations are semantically valid. Completeness ensures that our equations are *sufficient*—we can derive all semantically valid equations. We note that, in the case of the metric equations, the underlying idea is similar, although soundness and completeness are defined differently.

In order to define the interpretation of judgments $\Gamma \triangleright v : \mathbb{A}$, it is necessary to establish some notation first. Let C be a symmetric monoidal closed category and A, B and C be objects of this category.
Recall that in a closed monoidal category C, we have a natural isomorphism:

$$\mathrm{Hom}_{\mathsf{C}}(A \otimes B, C) \cong \mathrm{Hom}_{\mathsf{C}}(A, B \multimap C).$$

This isomorphism is known as *currying*. For each morphism $f\colon A \otimes B \to C$, its *curried form* $\overline{f}\colon A \to (B \multimap C)$ is the morphism corresponding to f under this isomorphism. The inverse operation, called *application* or *evaluation*, is given by the *application morphism* $\mathsf{app}_{B,C}\colon (B \multimap C) \otimes B \to C$.
For all ground types $X \in G$ the interpretation of $[\![X]\!]$ is postulated to be an object of C. Types are interpreted inductively using the unit $\mathbb{I}$, the tensor $\otimes$, the coproduct $\oplus$, 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 Γ' is non-empty and $[\![\Gamma', x : \mathbb{A}]\!] = [\![\mathbb{A}]\!]$ otherwise. The empty context $-$ is interpreted as $[\![-]\!] = \mathbb{I}$. Given $A_1, ..., A_n \in$ C, the n-tensor $(\ldots(A_1 \otimes A_2) \otimes \ldots) \otimes A_n$ is denoted as $A_1 \otimes \ldots \otimes A_n$, and similarly for morphisms.

2.3.1 Semantics

"Housekeeping" morphisms are employed to handle interactions between context interpretation and the symmetric monoidal struture of C. Given $\Gamma_1, \ldots, \Gamma_n$, the morphism that splits $[\![\Gamma_1, \ldots, \Gamma_n]\!]$ into $[\![\Gamma_1]\!] \otimes \ldots \otimes [\![\Gamma_n]\!]$ is denoted by $\mathsf{sp}_{\Gamma_1;\ldots;\Gamma_n} : [\![\Gamma_1, \ldots, \Gamma_n]\!] \to [\![\Gamma_1]\!] \otimes \ldots \otimes [\![\Gamma_n]\!]$.

For $n = 1$, $\mathsf{sp}_{\Gamma_1} = \mathrm{id}$. Let Γ_1 and Γ_2 be two contexts, $\mathsf{sp}_{\Gamma_1,\Gamma_2} : [\![\Gamma_1 \otimes \Gamma_2]\!] \to [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!]$ is defined as:

$$\mathsf{sp}_{-;\Gamma_2} = \lambda^{-1} \qquad \mathsf{sp}_{\Gamma_1;-} = \rho^{-1} \qquad \mathsf{sp}_{\Gamma_1;x:\mathbb{A}} = \mathrm{id} \qquad \mathsf{sp}_{\Gamma_1;\Delta,x:\mathbb{A}} = \alpha \cdot (\mathsf{sp}_{\Gamma_1;\Delta} \otimes \mathrm{id})$$

For $n > 2$, $\mathsf{sp}_{\Gamma_1;\ldots;\Gamma_n} : [\![\Gamma_1;\ldots;\Gamma_n]\!] \to [\![\Gamma_1]\!] \otimes \ldots \otimes [\![\Gamma_n]\!]$ is is defined recursively based on the previous definition, using induction on n:

$$\mathsf{sp}_{\Gamma_1;\ldots;\Gamma_n} = (\mathsf{sp}_{\Gamma_1;\ldots;\Gamma_{n-1}} \otimes \mathrm{id}) \cdot \mathsf{sp}_{\Gamma_1,\ldots,\Gamma_{n-1};\Gamma_n}$$

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 morphism permuting x and y is denoted by $\mathsf{exch}_{\Gamma,x:\mathbb{A},y:\mathbb{B},\Delta} : [\![\Gamma, \underline{x : \mathbb{A}, y : \mathbb{B}}, \Delta]\!] \to [\![\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta]\!]$ and defined as:

$$\mathsf{exch}_{\Gamma,\underline{x:\mathbb{A},y:\mathbb{B}},\Delta} = \mathsf{jn}_{\Gamma;y:\mathbb{B},x:\mathbb{A};\Delta} \cdot (\mathrm{id} \otimes \mathsf{sw} \otimes \mathrm{id}) \cdot \mathsf{sp}_{\Gamma;x:\mathbb{A},y:\mathbb{B};\Delta}$$

The shuffling morphism $\mathsf{sh}_E : [\![E]\!] \to [\![\Gamma_1,\ldots,\Gamma_n]\!]$ is defined as a suitable composition of exchange morphisms.

For every operation symbol $f : \mathbb{A}_1,\ldots,\mathbb{A}_n \to \mathbb{A}$ it is assumed the existence of a morphism $[\![f]\!] : [\![\mathbb{A}_1]\!] \otimes \ldots \otimes [\![\mathbb{A}_n]\!] \to [\![\mathbb{A}]\!]$. The interpretation of judgments is defined by induction over derivations according to the rules in Figure 4.

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

$$\frac{}{\llbracket x : \mathbb{A} \rhd x : \mathbb{A} \rrbracket = \mathrm{id}_{\llbracket \mathbb{A} \rrbracket}} \qquad \frac{}{\llbracket - \rhd * : \mathbb{I} \rrbracket = \mathrm{id}_{\llbracket \mathbb{I} \rrbracket}}$$

$$\frac{\llbracket \Gamma \rhd v : \mathbb{A} \otimes \mathbb{B} \rrbracket = m \quad \llbracket \Delta, x : \mathbb{A}, y : \mathbb{B} \rhd w : \mathbb{D} \rrbracket = n \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{\llbracket E \rhd \mathsf{pm}\, v\, \mathsf{to}\, x \otimes y.w : \mathbb{D} \rrbracket = n \cdot \mathsf{jn}_{\Delta;\mathbb{A};\mathbb{B}} \cdot \alpha \cdot \mathsf{sw} \cdot (m \otimes \mathrm{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E}$$

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

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

$$\frac{\llbracket \Gamma \rhd v : \mathbb{A} \multimap \mathbb{B} \rrbracket = m \quad \llbracket \Delta \rhd w : \mathbb{A} \rrbracket = n \quad E \in \mathit{Sf}(\Gamma; \Delta)}{\llbracket E \rhd v\, w : \mathbb{B} \rrbracket = \mathsf{app} \cdot (m \otimes n) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E}$$

$$\frac{\llbracket \Gamma \rhd v : \mathbb{A} \rrbracket = m}{\llbracket \Gamma \rhd \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A} \oplus \mathbb{B} \rrbracket = \mathrm{inl} \cdot m} \qquad \frac{\llbracket \Gamma \rhd v : \mathbb{B} \rrbracket = m}{\llbracket \Gamma \rhd \mathsf{inr}_{\mathbb{A}}(v) : \mathbb{A} \oplus \mathbb{B} \rrbracket = \mathrm{inr} \cdot m}$$

$$\frac{\llbracket \Gamma \rhd v : \mathbb{A} \oplus \mathbb{B} \rrbracket = b \quad \llbracket \Delta, x : \mathbb{A} \rhd w : \mathbb{D} \rrbracket = p \quad \llbracket \Delta, y : \mathbb{B} \rhd u : \mathbb{D} \rrbracket = q \quad E \in \mathrm{Sf}(\Gamma; \Delta)}{\begin{aligned} \llbracket E \rhd \mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} : \mathbb{D} \rrbracket = [p, q] \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot \\ (b \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E \end{aligned}}$$

Figure 4: Judgment interpretation

The following diagrams are useful for a clearer understanding of the interpretation of judgements given in Figure 4.

$$\begin{aligned} \llbracket \mathsf{ax} \rrbracket : \qquad & \llbracket E \rrbracket \xrightarrow{\mathsf{sh}_E} \llbracket \Gamma_1, \ldots, \Gamma_n \rrbracket \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} \llbracket \Gamma_1 \rrbracket \otimes \ldots \otimes \llbracket \Gamma_n \rrbracket \\ & \xrightarrow{m_1 \otimes \ldots \otimes m_n} \llbracket \mathbb{A}_1 \rrbracket \otimes \ldots \otimes \llbracket \mathbb{A}_n \rrbracket \xrightarrow{\llbracket f \rrbracket} \llbracket \mathbb{A} \rrbracket \\ \llbracket \mathsf{hyp} \rrbracket : \qquad & \llbracket \mathbb{A} \rrbracket \xrightarrow{\mathrm{id}_{\llbracket \mathbb{A} \rrbracket}} \llbracket \mathbb{A} \rrbracket \\ \llbracket \mathbb{I}_i \rrbracket : \qquad & \llbracket \mathbb{I} \rrbracket \xrightarrow{\mathrm{id}_{\llbracket \mathbb{I} \rrbracket}} \llbracket \mathbb{I} \rrbracket \\ \llbracket \otimes_e \rrbracket : \qquad & \llbracket E \rrbracket \xrightarrow{\mathsf{sh}_E} \llbracket \Gamma, \Delta \rrbracket \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} \llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket \xrightarrow{m \otimes \mathrm{id}} (\llbracket \mathbb{A} \rrbracket \otimes \llbracket \mathbb{B} \rrbracket) \otimes \llbracket \Delta \rrbracket \\ & \xrightarrow{\mathsf{sw}} \llbracket \Delta \rrbracket \otimes (\llbracket \mathbb{A} \rrbracket \otimes \llbracket \mathbb{B} \rrbracket) \xrightarrow{\alpha} (\llbracket \Delta \rrbracket \otimes \llbracket \mathbb{A} \rrbracket) \otimes \llbracket \mathbb{B} \rrbracket \xrightarrow{\mathsf{jn}_{\Delta;\mathbb{A};\mathbb{B}}} \llbracket \Delta, \mathbb{A}, \mathbb{B} \rrbracket \\ & \xrightarrow{n} \llbracket \mathbb{D} \rrbracket \end{aligned}$$

$$\begin{aligned}
&[\![\otimes_i]\!] : && [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma,\Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{m \otimes n} [\![\mathbb{A}]\!] \otimes [\![\mathbb{B}]\!] \\
&[\![\mathbb{I}_e]\!] : && [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma,\Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{m \otimes \mathrm{id}} [\![\mathbb{I}]\!] \otimes [\![\Delta]\!] \xrightarrow{\lambda} [\![\Delta]\!] \xrightarrow{n} [\![\mathbb{A}]\!] \\
&[\![\multimap_i]\!] : && [\![\Gamma]\!] \xrightarrow{\overline{m \cdot \mathsf{jn}_{\Gamma;\mathbb{A}}}} [\![\mathbb{A}]\!] \multimap [\![\mathbb{B}]\!] \qquad\qquad ([\![\Gamma]\!] \otimes [\![\mathbb{A}]\!] \xrightarrow{\mathsf{jn}_{\Gamma;\mathbb{A}}} [\![\Gamma,\mathbb{A}]\!] \xrightarrow{m} [\![\mathbb{B}]\!]) \\
&[\![\multimap_e]\!] : && [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma,\Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{m \otimes n} ([\![\mathbb{A}]\!] \multimap [\![\mathbb{B}]\!]) \otimes [\![\mathbb{A}]\!] \xrightarrow{\mathsf{app}} [\![\mathbb{B}]\!] \\
&[\![\mathsf{inl}]\!] : && [\![\Gamma]\!] \xrightarrow{m} [\![\mathbb{A}]\!] \xrightarrow{\mathrm{inl}} [\![\mathbb{A} \oplus \mathbb{B}]\!] \\
&[\![\mathsf{inr}]\!] : && [\![\Gamma]\!] \xrightarrow{m} [\![\mathbb{B}]\!] \xrightarrow{\mathrm{inr}} [\![\mathbb{A} \oplus \mathbb{B}]\!] \\
&[\![\mathsf{case}]\!] : && [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma,\Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{b \otimes \mathrm{id}} ([\![\mathbb{A}]\!] \oplus [\![\mathbb{B}]\!]) \otimes [\![\Delta]\!] \\
& && \xrightarrow{\mathrm{dist}} ([\![\mathbb{A}]\!] \otimes [\![\Delta]\!]) \oplus ([\![\mathbb{B}]\!] \otimes [\![\Delta]\!]) \\
& && \xrightarrow{\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}} [\![\Delta,\mathbb{A}]\!] \oplus [\![\Delta,\mathbb{B}]\!] \xrightarrow{[p,q]} [\![\mathbb{D}]\!]
\end{aligned}$$

Regarding the interpretation of the exhange and substitution properties, we have the following lemma.

Lemma 2.3.1. *For any judgements* $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \rhd v : \mathbb{D}$*,* $\Gamma, x : \mathbb{A} \rhd v : \mathbb{B}$*, and* $\Delta \rhd w : \mathbb{A}$*, the following holds:*

$$[\![\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta \rhd v : \mathbb{D}]\!] = [\![\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta \rhd v : \mathbb{D}]\!] \cdot \mathsf{exch}_{\Gamma,\underline{x:\mathbb{A},y:\mathbb{B}},\Delta}$$

$$[\![\Gamma, \Delta \rhd v[w/x] : \mathbb{B}]\!] = [\![\Gamma, x : \mathbb{A} \rhd v : \mathbb{B}]\!] \cdot \mathsf{jn}_{\Gamma;\mathbb{A}} \cdot (\mathrm{id} \otimes [\![\Delta \rhd w : \mathbb{A}]\!]) \cdot \mathsf{sp}_{\Gamma;\Delta}$$

Proof. This lemma is proved in [1, Lemma 2.2] for the lambda calculus without conditionals, so we only need to address the conditional cases.

Passar para aqui as provas qd elas estiverem vistas

□

Definition 2.3.2 (Models of linear λ-theories)**.** Consider a linear λ-theory $((G, \Sigma), Ax)$ and a symmetric monoidal closed category with coproducts C. Suppose that for each $X \in G$, we have an interpretation $[\![X]\!]$, which is an object of C, and analogously for the operation symbols in Σ. This interpretation structure is a *model* of the theory if all axioms in Ax are satisfied by the interpretation.

Theorem 2.3.3. *The equations presented in Figure 2 are sound with respect to judgement interpretation. More specifically, if* $\Gamma \rhd v = w : \mathbb{A}$ *is one of the equations in Figure 2, then* $[\![\Gamma \rhd v : A]\!] = [\![\Gamma \rhd w : \mathbb{A}]\!]$.

Proof. Since the theorem is already proven in [1, Theorem 2.3] for the lambda calculus without conditionals, it suffices to consider the cases involving conditionals.

Passar para aqui as provas qd elas estiverem vistas

□

Theorem 2.3.4 (Completeness)**.** *Consider a linear λ-theory T. Then an equation $\Gamma \triangleright v = w : \mathbb{A}$ is a theorem of T if and only if it is satisfied by all models of the theory.*

Proof. This theorem is proved in [1, Lemma 2.6] for the lambda calculus without conditionals, so we only need to address the cases involving conditionals.

Passar para aqui as provas qd elas estiverem vistas

□

Example 2.3.5. We now illustrate how the programs presented in Examples 2.2.5 and 2.2.6, with slight modifications, are interpreted in Set. To this effect, we consider a type N representing the set of natural numbers, along with a family of operations $\{n : \mathbb{I} \to N \mid n \in \mathbb{N}\}$, each mapping the monoidal unit to a corresponding natural number n. We consider yet another operation dis that marks elements of of type N as discardable, dis $: N \to \mathbb{I}$. In Set we have $[\![\mathbb{I}]\!] = \{*\}$, define $[\![N]\!] = \mathbb{N}$, $[\![n]\!] = \{*\} \to \mathbb{N},\ * \mapsto n,$ and $[\![\text{dis}]\!] =!$ where ! denotes the terminal map. Consider the following λ-term:

$$x : N \otimes N \triangleright \text{pm } x \text{ to } a \otimes b.b \otimes a : N \otimes N$$

Attending to Figure 4 and the coherence theorem for symmetric monoidal categories this program is interpreted as follows:

$$\begin{aligned}
&[\![\text{pm } x \text{ to } a \otimes b.b \otimes a]\!] \\
&\triangleq [\![b \otimes a]\!] \cdot \mathrm{jn} \cdot \alpha \cdot \mathrm{sw} \cdot ([\![x]\!] \otimes \mathrm{id}) \cdot \mathrm{sp}_{N \otimes N;-} \\
&= \mathrm{sw} \cdot \mathrm{sp}_{N;N} \cdot \mathrm{jn}_{\mathbb{I};N;N} \cdot \alpha \cdot \mathrm{sw}_{N \otimes N;\mathbb{I}} \cdot \mathrm{sp}_{N \otimes N;-} \\
&= \mathrm{sw} && \text{(coherence theorem)}
\end{aligned}$$

Next, consider the λ-terms below.

$$\textbf{Dis2nd} \triangleq - \triangleright \lambda x : N \otimes \mathbb{I}.\text{pm } x \text{ to } a \otimes b.b \text{ to } *.a : N \otimes N \multimap N$$

$$\textbf{Dis2nd}\ (1(*) \otimes *)$$

In this case, we will coordinate the use of the equational system with the semantics, thus illustrating the synergy that both create. First, applying equations β and $\beta_{\otimes_e}$ we have:

$$\begin{aligned}\textbf{Dis2nd}\,(1(*)\otimes 2(*)) &\triangleq \textbf{Dis2nd}\,[1(*)\otimes 2(*)/x]\\ &= \mathsf{pm}\ 1(*)\otimes 2(*)\ \mathsf{to}\ a\otimes b.\,\mathsf{dis}(b)\ \ \mathsf{to}\ *.a\\ &= \mathsf{dis}(2(*))\ \ \mathsf{to}\ *.1(*)\end{aligned}$$

The resulting program is interpreted as follows:

$$\begin{aligned}&[\![\mathsf{dis}(2(*))\ \ \mathsf{to}\ *.1(*)]\!]\\ &= [\![1(*)]\!]\cdot\lambda\cdot[\![\mathsf{dis}(2(*))]\!]\otimes\mathrm{id}\cdot\lambda^{-1}\\ &= [\![1(*)]\!]\cdot\lambda\cdot(!\cdot[\![2(*)]\!]\otimes\mathrm{id})\cdot\lambda^{-1}\\ &= [\![1(*)]\!]\cdot\lambda\cdot(!\cdot\otimes\mathrm{id})\cdot\lambda^{-1}\\ &= [\![1(*)]\!]\cdot\lambda\cdot\mathrm{id}\cdot\lambda^{-1}\\ &= [\![1(*)]\!]\end{aligned}$$

2.3.2 Semantics of metric equations

We will now turn our attention to the semantics of the metric equations. First, we recall the definitions of a metric space and of the category of metric spaces.

Definition 2.3.6. A *metric space* is a pair (X, d) where X is a set and $d : X \times X \to [0, \infty]$ is a function known as *distance* satisfying:

1. $0 \leq d(x, y)$, with equality if and only if $x = y$,
2. $d(x, y) = d(y, x)$,
3. $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in X$.

A *pseudometric space* satisfies the same axioms, except that the first condition condition is weakened: $d(x, y) = 0$ may hold even when $x \neq y$.

Definition 2.3.7. Met denotes the category whose objects are metric spaces and whose morphisms are non-expansive maps, *i.e.*, functions that do not increase the distance between points. More precisely, for two metric spaces (X, d_X) and (Y, d_Y), a morphism $f : (X, d_X) \to (Y, d_Y)$ is a function $f : X \to Y$ such that

$$d_Y(f(x), f(y)) \leq d_X(x, y) \quad \text{for all } x, y \in X.$$

Here, we equip each hom-set $\mathsf{C}(A,B)$ of a category C with a metric $d_{A,B}$, and impose that both postcomposition and precomposition are non-expansive. That is, for all morphisms $f, f_1, f_2 \in \mathsf{C}(A,B)$ and any $g, g_1, g_2 \in \mathsf{C}(B,C)$, the following inequalities holds:

$$d_{A,C}(g \circ f_1, g \circ f_2) \leq d_{A,B}(f_1, f_2) \qquad d_{A,C}(g_1 \circ f, g_2 \circ f) \leq d_{B,C}(g_1, g_2).$$

Note that, given the triangle inequality, we have:

$$d_{A,C}(g_1 \circ f_1, g_2 \circ f_2) \leq d_{A,C}(g_1 \circ f_1, g_1 \circ f_2) + d_{A,C}(g_1 \circ f_2, g_2 \circ f_2) \leq d_{A,B}(f_1, f_2) + d_{B,C}(g_1, g_2).$$

This is known as *enriching* the category C *over metric spaces*. Accordingly, we often refer to such a category as being *enriched over metric spaces*, or simply as a Met-category.

Following the same principle, we require that the tensor product be non-expansive, *i.e.*,

$$d_{A\otimes C, B\otimes D}(f_1 \otimes g_1, f_2 \otimes g_2) \leq d_{A,B}(f_1, f_2) + d_{C,D}(g_1, g_2).$$

In the literature, such a tensor product is typically referred to as a *functor enriched over metric spaces*, or simply a Met-*functor*. Similarly, we require the currying functor $A \multimap (-) \colon \mathsf{C} \to \mathsf{C}$ to be non-expansive.

Coproducts are not discussed in this context (for now), as they relate to the interpretation of the metric equation for conditionals, which will be the subject of the next chapter.

In this context, *soundness* and *completeness* concepts are extended to encompass not only the classical equations but also the metric equations. Recall classical equations $v = w$ can be written as $v =_0 w$. As a result, in this metric setting, we define

- *Soundness* as the property

  $$M =_\varepsilon N \;\Rightarrow\; d(\llbracket N \rrbracket, \llbracket M \rrbracket) \leq \varepsilon \quad \text{for all interpretations in the class.}$$

  That is, if two terms are provably at a maximum distance ε, so are their respective interpretations

- *Completeness* as the property

  $$d(\llbracket N \rrbracket, \llbracket M \rrbracket) \leq \varepsilon \;\Rightarrow\; M =_\varepsilon N \quad \text{for all interpretations in the class.}$$

  That is, if ε is the maximum distance between the interpretations of two programs, then they are provably at a maximum distance ε.

Definition 2.3.8. Consider a metric λ-theory $((G,\Sigma), Ax)$ and a symmetric monoidal closed Met-category C, in which both the tensor product and the internal hom-functor (currying) are non-expansive. Suppose that for each $X \in G$ we have an interpretation $[\![X]\!]$ as a C-object and analogously for the operation symbols. This interpretation structure is a model of the theory if all axioms in Ax are satisfied by the interpretation.

Theorem 2.3.9 (Soundness). *[1, Theorem 3.14] The rules in Figures 2 and 3 are sound for a symmetric monoidal closed* Met-*category* C*, in which both the tensor product and the internal hom-functor (currying) are non-expansive. Specifically, if* $\Gamma \triangleright v =_q w : \mathbb{A}$ *results from the rules in Figures 2 and 3 then* $q \geq d([\![\Gamma \ \ \triangleright v : \mathbb{A}]\!], [\![\Gamma \triangleright w : \mathbb{A}]\!])$.

Next, we will provide a proof sketch of the completeness result in [1] so the reader gets a general feeling of what it requires.

For two types $\mathbb{A}$ and $\mathbb{B}$ of a metric λ-theory T, consider the set $(\texttt{Values}\mathbb{A}, \mathbb{B})$ of values v such that $x : \mathbb{A} \triangleright v : \mathbb{B}$. We equip $\texttt{Values}(\mathbb{A}, \mathbb{B})$ with the function $d : \texttt{Values}(\mathbb{A}, \mathbb{B}) \times \texttt{Values}(\mathbb{A}, \mathbb{B}) \to [0, \infty]$ defined by,

$$d(v, w) = \inf\{q \,|\, v =_q w \text{ is a theorem of } T\}.$$

Given that the equations $\Gamma \triangleright v = w : \mathbb{A}$ are abbreviations of $\Gamma \triangleright v =_0 w : \mathbb{A}$, $\texttt{Values}((\mathbb{A}, \mathbb{B}), d)$ is a pseudometric space, *i.e.*, it allows distinct terms to have distance zero. Consequently, we quotient this space by the relation $\sim$ (identifying elements at distance zero) to obtain a metric space, denoted by $(\texttt{Values}(\mathbb{A}, \mathbb{B}), d)/\sim$, which is a Met-category.

Completeness arises from constructing the syntactic category Syn(T) of the underlying theory T and then showing that provability of $\Gamma \triangleright v =_q w : \mathbb{A}$ in T is equivalent to $d([\![v]\!], [\![w]\!]) \geq q$ in the category Syn(T). We use the category $(\texttt{Values}(\mathbb{A}, \mathbb{B}), d)/\sim$ to this end. Note that the quotienting process identifies all terms $x : \mathbb{A} \triangleright v : \mathbb{B}$ and $x : \mathbb{A} \triangleright w : \mathbb{B}$ such that $v =_0 w$ and $w =_0 v$. This relation includes the equations-in-context from Figure 2. Then, the next step is to prove that this quotienting procedure is compatible with the term formation rules of the extended calculus. To this effect, one generally uses the fact that $\otimes$ distributes over suprema. This yields the desired category Syn(T) which will correspond to a symmetric monoidal closed Met-category C, in which both the tensor product and the internal currying are non-expansive. The final step is to show that if an equation $\Gamma \triangleright v =_q v' : \mathbb{A}$ with $q \in [0, \infty]$ is satisfied by Syn(T), then it is a theorem of the linear metric λ-theory. Which follows from the strictly greater relation and rules (join), (weak), and (arch).

Theorem 2.3.10 (Completeness)**.** *[1, Theorem 3.16] Consider a metric λ-theory. A metric equation in context $\Gamma \triangleright v \ =_q \ w \ : \ \mathbb{A}$ is a theorem if and only if it holds in all models of the theory.*

Chapter 3

A Metric Equational System for Conditionals

A metric equational system for the conditionals would be extremely helpful for reasoning about approximate equivalence in the setting of programming and beyond. In this chapter, we address this gap by introducing such a system and corresponding models, and we then establish a soundness and completeness result in the same style as before. Additionally, we illustrate our system at work via a small example: a metric version of copairing's extensionality. Next, we introduce different models of our equational system. These include the category of metric spaces and all categories arising from a Met-enriched version of coproduct cocompletion. We conclude the chapter with a brief discussion about our system.

3.1 System

Our system for conditionals is presented in Figure 5.

$$\frac{v =_q w}{\mathrm{inl}_{\mathbb{B}}(v) =_q \mathrm{inl}_{\mathbb{B}}(w)} \qquad \frac{v =_q w}{\mathrm{inr}_{\mathbb{A}}(v) =_q \mathrm{inl}_{\mathbb{A}}(w)}$$

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

Figure 5: Equational system for condicionals

Firstly, we observe that our equational system encompasses both Figure 5 and Figure 3. Consequently, the inl and inr equations are redundant, as they can be derived from the substitution rule in Figure 3. Nevertheless, we have chosen to include them explicitly to emphasize that the injections preserve (*i.e.*, do not increase) the distance between terms. Moreover, note that, strictly speaking, we can always use substitution to reason about case statements; however, the introduced equation provides a tighter bound.

Intuitively, the equation for the case statement provides a bound for the worst-case scenario: the only branch executed (either w or u) is the one that is at the greatest distance from its counterpart (w' or u', respectively). Therefore, the maximum distance between the two branches is taken as the bound.

Additionally, observe that while the original metric system gives the operation "$+$" a predominant role, the extended version assigns similar importance to the $\sup$ operator.

Recall Definition 2.2.20 where we presented the notion of metria λ-theory. Now, we extend the set of theorems $Th(Ax)$ to denote the smallest set that contains Ax and is closed under the rules presented in Figure 2, Figure 3, and Figure 5.

3.1.1 Extensionality of the copairing

We establish a metric version of copairing extensionality. Just as the classical extensionality principle for copairings served as the foundation for demonstrating that terms of type $\mathbb{I} \oplus \mathbb{I}$ satisfy certain axioms of a Boolean algebra, our metric copairing extensionality will play an analogous role in metric-based reasoning.

Assume that

$$\begin{cases} (\lambda x.v)\, \mathsf{inl}(y) =_{\varepsilon_1} (\lambda x.v)\, \mathsf{inl}(y) \\ (\lambda x.w)\, \mathsf{inr}(z) =_{\varepsilon_2} (\lambda x.w)\, \mathsf{inr}(z) \end{cases} \implies \lambda x.v = \lambda x.w$$

Following the same reasoning as before in Section 2.2.9 and applying the metric equation for condicionals we obtain:

$$\begin{aligned} v &= \mathsf{case}\; x \left\{ \mathrm{inl}(y) \Rightarrow v\, [\mathsf{inl}(y)/x]; \mathrm{inl}(z) \Rightarrow v\, [\mathsf{inr}(z)/x] \right\} && (\eta_{case}) \\ &=_{\max\{\varepsilon_1, \varepsilon_2\}} \mathsf{case}\; x \left\{ \mathrm{inl}(y) \Rightarrow w\, [\mathsf{inl}(y)/x]; \mathrm{inl}(z) \Rightarrow w\, [\mathsf{inr}(z)/x] \right\} = w && (\text{Figure 5}, \eta_{case}) \end{aligned}$$

As a result, we have $v =_{\max\{\varepsilon_1, \varepsilon_2\}} w$ and derive $\lambda x.v =_{\max\{\varepsilon_1, \varepsilon_2\}} \lambda x.w$ using the metric equational system-

3.2 Interpretation

In this subsection, we extend the concepts introduced in Section 2.3.2 to include the interpretation of the extended equational system.

Definition 3.2.1. A *symmetric monoidal closed* Met-*category with binary coproducts* C is a Met-category that is symmetric monoidal closed and has binary coproducts such that, for all morphisms $f_1, f_2 \in \mathsf{C}(A, C)$ and $g_1, g_2 \in \mathsf{C}(B, C)$, we have:

$$d_{A \oplus B, C}([f_1, g_1], [f_2, g_2]) \leq \sup\{d_{A,C}(f_1, f_2), d_{B,C}(g_1, g_2)\}.$$

We present the category of metric spaces as an example of Definition 3.2.1.

Proposition 3.2.2. *The category* Met *is a symmetric monoidal closed with binary coproducts* Met-*category*

Proof. By [1, Example 3.8], the category Met is a symmetric monoidal closed Met-category. As a result, it suffices to show that for all morphisms $f_1, f_2 \in \mathsf{Met}(A, C)$ and $g_1, g_2 \in \mathsf{Met}(B, C)$, the following inequality holds:

$$d_{A \oplus B, C}([f_1, g_1], [f_2, g_2]) \leq \sup\{d_{A,C}(f_1, f_2), d_{B,C}(g_1, g_2)\}.$$

In this category, the coproduct is defined as in Set. The distance function d on the coproduct $A \oplus B$ is given by:

$$\begin{cases} d_{A \oplus B}(\mathrm{inl}(a_1), \mathrm{inl}(x_2)) = d_A(a_1, a_2) \\ d_{A \oplus B}(\mathrm{inr}(b_1), \mathrm{inr}(b_2)) = d_B(b_1, b_2) \\ d_{A \oplus B}(\mathrm{inl}(a), \mathrm{inr}(b)) = d_{A \oplus B}(\mathrm{inr}(b), \mathrm{inl}(a)) = \infty \end{cases}$$

The co-pairing is defined as in Set. The inequality we aim to prove follows directly from the fact that, given two morphisms $f, g \in \mathsf{Met}(A, B)$ the distance between them is defined as

$$\sup\{d_A(fa, ga) \mid a \in A\},$$

together with Lemma 3.3.5. For $f_1, f_2 \in \mathsf{Met}(A, C)$ and $g_1, g_2 \in \mathsf{Met}(B, C)$, we calculate:

Não estou a perceber o problema de $(a, b) \in A \oplus B$? Nós usamos $A \otimes B$ acima tb, não podemos usar (a, b)?

$$\begin{aligned}
&d_{A\oplus B,C}([f_1, g_1], [f_2, g_2]) \\
&\quad = \sup\{d_{A\oplus B}([f_1, g_1](x), [f_2, g_2](x)) \mid x \in A \oplus B\} \\
&\quad = \sup\{\{d_{A\oplus B}([f_1, g_1](\mathrm{inl}(a)), [f_2, g_2](\mathrm{inl}(a))) \mid a \in A\} \\
&\qquad \cup \{d_{A\oplus B}([f_1, g_1](\mathrm{inr}(b)), [f_2, g_2](\mathrm{inr}(b))) \mid b \in B\}\} \\
&\quad = \sup\{\{d_A(f_1(a), f_2(a)) \mid a \in A\} \cup \{d_B(g_1(b), g_2(b)) \mid b \in B\}\} \\
&\quad = \sup\{\sup\{d_A(f_1(a), f_2(a)) \mid a \in A\}, \sup\{d_B(g_1(b), g_2(b)) \mid b \in B\}\} \\
&\quad = \sup\{d_{A,C}(f_1, f_2), d_{B,C}(g_1, g_2)\}.
\end{aligned}$$

□

Definition 3.2.3. Consider a metric λ-theory $((G, \Sigma), Ax)$ and a symmetric monoidal closed Met-category with binary coproducts C, in which both the tensor product and the currying are non-expansive. Suppose that for each $X \in G$ we have an interpretation $[\![X]\!]$ as a C-object and analogously for the operation symbols. This interpretation structure is a model of the theory if all axioms in Ax are satisfied by the interpretation.

3.3 Soundeness and Completeness

In this section we establish a soundness and completeness result in the same style as before.

Lattice Theory Preliminaries

First, we introduce a few concepts from lattice theory that will be useful for the completeness proof and for a broader discussion of our results.

Definition 3.3.1. A *lattice* is a partial order in which every finite subset has both a meet and a join. A *complete lattice* is a partial order in which every subset, finite or infinite, has a meet and a join.

Definition 3.3.2. A subset D of a lattice L is called *directed* if it is nonempty and every finite subset of D has an upper bound in D. A partially ordered set is said to be *directed complete* if every directed subset has a supremum. A directed complete poset is commonly referred to as a *dcpo*.

Definition 3.3.3. A lattice L is called *meet continuous* if it is directed complete, *i.e.*a dcpo, and satisfies the condition

$$\inf\{x, \sup D\} = \sup\{\inf\{x, d\} \mid d \in D\}$$

for all $x \in L$ and all directed subsets $D \subseteq L$.

Lemma 3.3.4. *The* $[0, \infty]$ *lattice is meet continuous.*

Proof. Follows from [1] and [29, Proposition I-1.8] □

Note that, since the order is reversed in this quantale, suprema in general lattices correspond to infima here, and vice versa.

Lemma 3.3.5. *[30, Lemma 2.23] Let L be a lattice, let $A, B \subseteq L$ and assume that* $\sup A$*,* $\sup B$*,* $\inf A$ *and* $\inf B$ *exist in L. Then*

$$\sup\{A \cup B\} = \sup\{\sup A, \sup B\} \quad \textit{and} \quad \inf\{A \cup B\} = \inf\{\inf A, \inf B\}.$$

Soundeness and Completeness

The proofs in this section are based on the proofs of Theorem 2.3.9 and Theorem 2.3.10 in [1].

Theorem 3.3.6. *The rules in Figures 2, 3 and 5 are sound for a symmetric monoidal closed* Met*-category with coproducts* C*, in which both the tensor product and the internal hom-functor (currying) are non-expansive. Specifically, if* $\Gamma \triangleright v =_q w : \mathbb{A}$ *results from the rules in Figures 2, 3 and 5 then* $q \geq d([\![\Gamma \ \ \triangleright v : \mathbb{A}]\!], [\![\Gamma \triangleright w : \mathbb{A}]\!])$*.*

Proof. We follow the same strategy as in [1]. This proof uses induction over the depth of proof trees that arise from the metric deductive system. The general strategy for each inference rule is to use the definition of a symmetric monoidal closed Met-category with coproducts. More concretely, first, consider the equations on Figure 2 which abbreviate equations $\Gamma \triangleright w =_0 v : \mathbb{A}$. By Theorem 2.3.3, these equations are sound for symmetric monoidal closed categories with binary coproducts, *i.e.*if $v = w$, then $[\![v]\!] = [\![w]\!]$ in C. Then by the definition of metric space we obtain $d([\![v]\!], [\![w]\!]) = d([\![w]\!], [\![v]\!]) = 0$. The rules in Figure 3 follow from the definition of Met-category, and the fact that the tensor product and the internal hom-functor (currying) are non-expansive. Finally, rules in Figure 5 follow from the non-expansive law imposed on binary coproducts Definition 3.2.1. Specifically,

$$
\begin{aligned}
& d(\llbracket \mathsf{case}\ v\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\}\rrbracket, \llbracket \mathsf{case}\ v'\ \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w';\ \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u'\}\rrbracket) \\
& \quad = d([\llbracket w\rrbracket, \llbracket u\rrbracket] \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v\rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E, \\
& \qquad\quad [\llbracket w'\rrbracket, \llbracket u'\rrbracket] \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v'\rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E) \\
& \quad \leq d([\llbracket w\rrbracket, \llbracket u\rrbracket] \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v\rrbracket \otimes \mathsf{id}), [\llbracket w'\rrbracket, \llbracket u'\rrbracket] \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \\
& \qquad\quad \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v'\rrbracket \otimes \mathsf{id})) \\
& \quad \leq d(\llbracket v\rrbracket \otimes \mathsf{id}, \llbracket v'\rrbracket \otimes \mathsf{id}) + d([\llbracket w\rrbracket, \llbracket u\rrbracket] \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist}, [\llbracket w'\rrbracket, \llbracket u'\rrbracket] \cdot \\
& \qquad\quad (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist}) \\
& \quad \leq q + d([\llbracket w\rrbracket, \llbracket u\rrbracket] \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist}, [\llbracket w'\rrbracket, \llbracket u'\rrbracket] \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \\
& \qquad\qquad \cdot \mathrm{dist}) \\
& \quad \leq q + d([\llbracket w\rrbracket, \llbracket u\rrbracket], [\llbracket w'\rrbracket, \llbracket u'\rrbracket]) \\
& \quad \leq q + \sup(d(\llbracket w\rrbracket, \llbracket w'\rrbracket), d(\llbracket u\rrbracket, \llbracket u'\rrbracket)) \\
& \quad \leq q + \sup\{r, s\}
\end{aligned}
$$

The second step follows from the fact that $\mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E$ is a morphism in C and that C is a Met-category. The third and fifth step follow from an analogous reasoning. The fourth step follows from the premises of the rule in question and the fact that C is a symmetric monoidal Met-category. The sixth step follows from the fact that C is a symmetric monoidal Met-category with binary coproducts. Finally, the last step follows from the premise of the rule in question.

□

We will now focus on completeness. We extend $\mathtt{Values}((\mathbb{A}, \mathbb{B}), d)$, by incorporating the new theorems (those concerning the metric equations for conditionals) and quotient this pseudo-dometric space into a metric space $(\mathtt{Values}(\mathbb{A}, \mathbb{B}), d)/\!\sim$ (in the same spirit as before, see Section 2.3.2).

Theorem 3.3.7 (Completeness)**.** *Consider a metric λ-theory. A metric equation in context $\Gamma \triangleright v =_q w : \mathbb{A}$ is a theorem if and only if it holds in all models of the theory.*

Proof. We will focus only on conditionals, as the remaining cases are proven in [1]. Completeness arises from constructing the syntactic category $\mathsf{Syn}(T)$ of the underlying theory T

and then showing that provability of $\Gamma \triangleright v =_q w : \mathbb{A}$ in T is equivalent to $d(\llbracket v \rrbracket, \llbracket w \rrbracket) \geq q$ in the category Syn(T). We use the category $(\texttt{Values}(\mathbb{A}, \mathbb{B}), d)/\sim$ to this end. Note that the quotienting process identifies all terms $x : \mathbb{A} \triangleright v : \mathbb{B}$ and $x : \mathbb{A} \triangleright w : \mathbb{B}$ such that $v =_0 w$ and $w =_0 v$. This relation includes the equations-in-context from Figure 2. Then, we remark that all sets of the form $\{q \mid v =_q w\}$ are directed: they are non-empty, since by rule (join) we always have at least $v =_\infty w$, and again by (join), every finite subset of $\{q \mid v =_q w\}$ has a lower bound in the set. This will be useful for applying Lemma 3.3.4.

Next, we need to prove that the copairing is well defined in (Syn(T)), *i.e.*, for any v, v', w, w' if $v \sim v'$ and $w \sim w'$, then $[v, w] \sim [v', w']$.

This is equivalent to demonstrating the following implication:

$$\begin{cases} \inf\{q \mid v =_q w\} \leq 0 \\ \inf\{r \mid v' =_r w'\} \leq 0 \end{cases} \implies \inf \left\{ q \;\middle|\; \begin{array}{l} \mathsf{case}\ z\,\{\mathsf{inl}(x) \Rightarrow v;\ \mathsf{inr}(y) \Rightarrow w\} =_q \\ \mathsf{case}\ z\,\{\mathsf{inl}(x) \Rightarrow v';\ \mathsf{inr}(y) \Rightarrow w'\} \end{array} \right\} \leq 0.$$

Let L be a lattice, and let $D, F \subseteq L$ be directed sets. Observe the following:

$$\begin{aligned} &\sup\{\inf D, \inf F\} \\ &= \inf\{\sup\{\inf D, f\} \mid f \in F\} && \text{(Lemma 3.3.4)} \\ &= \inf\{\inf\{\sup\{\inf d, f\} \mid d \in D\} \mid f \in F\} && \text{(Lemma 3.3.4)} \\ &= \inf\{\sup\{d, f\} \mid d \in D, f \in F\} && \text{(Lemma 3.3.5)} \end{aligned} \tag{3.1}$$

With the equality above, we have

$$\begin{aligned} &\begin{cases} \inf\{q \mid v =_q w\} \leq 0 \\ \inf\{r \mid v =_r w\} \leq 0 \end{cases} \\ &\implies \sup\{\inf\{q \mid v =_q w\}, \inf\{r \mid v =_r w\}\} \leq 0 \\ &\implies \inf\{\sup\{q, r\} \mid v =_q w, v =_r w\} \leq 0 && \text{(Equation 3.1)} \\ &\implies \left\{ q \;\middle|\; \begin{array}{l} \mathsf{case}\ z\,\{\mathsf{inl}(x) \Rightarrow v;\ \mathsf{inr}(y) \Rightarrow w\} =_q \\ \mathsf{case}\ z\,\{\mathsf{inl}(x) \Rightarrow v';\ \mathsf{inr}(y) \Rightarrow w'\} \end{array} \right\} \leq 0 \end{aligned}$$

Thus, we obtain a category Syn(T) with binary coproducts. The next step is to prove the required non-expansivity law concerning copairing. To this effect, we reason as follows:

$$\begin{aligned}
&\sup\{d([v],[w]),d([v'],[w'])\}\\
&=\sup\{d(v,w),d(v',w')\}\\
&=\sup\{\inf\{q\,|\,v=_q v'\},\inf\{r\,|\,w=_r w'\}\}\\
&=\inf\{\sup\{q,r\}|\,v=_q v',w=_r w'\}\\
&\geq\inf\{q\,|\ \mathsf{case}\ z\,\{\mathsf{inl}(x)\Rightarrow v;\ \mathsf{inr}(y)\Rightarrow w\}=_q\ \mathsf{case}\ z\,\{\mathsf{inl}(x)\Rightarrow v';\ \mathsf{inr}(y)\Rightarrow w'\}\}\\
&=d(\mathsf{case}\ z\,\{\mathsf{inl}(x)\Rightarrow v;\ \mathsf{inr}(y)\Rightarrow w\},\mathsf{case}\ z\,\{\mathsf{inl}(x)\Rightarrow v';\ \mathsf{inr}(y)\Rightarrow w'\})\\
&=d([\mathsf{case}\ z\,\{\mathsf{inl}(x)\Rightarrow v;\ \mathsf{inr}(y)\Rightarrow w\}],[\mathsf{case}\ z\,\{\mathsf{inl}(x)\Rightarrow v';\ \mathsf{inr}(y)\Rightarrow w'\}])\\
&=d([[v],[v']],[[w],[w']])
\end{aligned}$$

The third step follows from Equation 3.1, and the fourth step follows from the fact that for any sets A and B, if $A\subseteq B$, then $\inf\{A\}\geq\inf\{B\}$.

The final step is to show that if an equation $\Gamma\triangleright v\ =_q\ v'\ :\ \mathbb{A}$ with $q\ \in\ [0,\infty]$ is satisfied by Syn(T), then it is a theorem of the linear metric λ-theory. By assumption, $d([v],[v'])\ =\ d(v,v')=\ \ \inf\{r\,|\,v=_r v'\}\leq q$. It follows from the definition of the strictly greater relation that for all $x\in[0,\infty]$ with $x>q$ there exists a finite set $A\subseteq\{r\,|\,v=_r v'\}$ such that $x\geq\inf A$. Then by an application of rule (join) we obtain $v\ =_{\inf A}\ v'$, and consequently, rule (weak) provides $v=_x v'$ for all $x>q$. Finally, by applying rule (arch), we deduce that $v=_q v'$ is part of the theory. □

3.4 Coproduct cocompletion

The idea behind the coproduct completion of a category C is to create a new category where all small coproducts exist by formally adding them to C in the simplest way possible. We will show that this construction is compatible with our metric equational system.

Definition 3.4.1. A *(free) coproduct completion* of a category C, denoted C$^+$, is the category whose objects are families $(A_i)_{i\in I}$ of objects of C, where I is a set; an arrow $(A_i)_{i\in I}\to(B_j)_{j\in J}$ consists of a pair $(f,(\phi_i)_{i\in I})$, where $f\ :\ I\ \to\ J$ is a function between the index sets, and $(\phi_i)_{i\in I}$ is a family of morphisms $\phi_i\ :\ X_i\ \to\ Y_{f(i)}$ in C. Given morphisms $(f,\phi_i)\ :\ (A_i)_{i\in I}\ \to\ (B_j)_{j\in J}$ and $(g,\psi_i)\ :\ (Y_j)_{j\in J}\ \to\ (Z_k)_{k\in K}$, their composition is defined as the morphism, given by the pair $(g\cdot f,(\theta_i)_{i\in I})$, where $\theta_i:=\psi_{f(i)}\circ\varphi_i:X_i\to Z_{g(f(i))}$. From now on, unless ambiguities arise, we will omit the indexing function and use letters Φ, Ψ, ξ to refer to families of morphisms.

If C is a Met-category, one may define a metric on the morphims of its coproduct cocompletion C$^+$ as follows:

$$d(\Phi, \Psi) = \sup\{d'(\phi_i, \psi_i) \mid i \in I\}, \text{ where } d'(\phi_i, \psi_i) = \begin{cases} \infty, & \text{if } f(i) \neq g(i), \\ d(\phi_i, \psi_i), & \text{otherwise.} \end{cases}$$

Proposition 3.4.2. *The coproduct completion of a* Met*-category* C *is a* Met*-category.*

Proof. This follows from the fact that C is a Met-category. Considering the definition of a Met-category, we need to show that for all objects A, B, C in C$^+$, and for any morphisms $\Phi, \Phi' : \mathsf{C}^+(B, C), \Psi, \Psi' \in \mathsf{C}^+(A, B)$, it holds that:

$$d(\Phi \cdot \Psi, \Phi' \cdot \Psi') \leq d(\Phi, \Phi') + d(\Psi, \Psi').$$

Given our choice of metric, we have:

$$d(\Phi \cdot \Psi, \Phi \cdot \Psi') = \sup\{d'(\phi_{f(i)} \cdot \psi_i, \phi_{g(i)} \cdot \psi') \mid i \in I\}$$

First, suppose $f(i) = g(i)$ for all $i \in I$,

$$\begin{aligned} &\sup\{d'(\phi_{f(i)} \cdot \psi_i, \phi_{g(i)} \cdot \psi') \mid i \in I\} \\ &= \sup\{d(\phi_{f(i)} \cdot \psi_i, \phi_{f(i)} \cdot \psi_i') \mid i \in I\} \\ &\leq \sup\{d(\psi_i, \psi_i') \mid i \in I\} && (\text{C is a Met-category}) \\ &= d(\Psi, \Psi') \end{aligned}$$

Next, assume $f(i) \neq g(i)$ for any $i \in I$, it is direct that

$$\sup\{d'(\phi_{f(i)} \cdot \psi_i, \phi_{g(i)} \cdot \psi')\} \leq \infty = \sup\{d'(\psi_i, \psi_i')\} = d(\Psi, \Psi')$$

The proof for precomposition follows a similar reasoning. □

Proposition 3.4.3. *The coproduct completion of a* Met*-category* C *is a* Met*-category with binary coproducts.*

Proof. This follows from Proposition 3.4.2 and the definition of the metric in C$^+$. For any objects A, B, C in C$^+$ and morphisms $\Phi, \Phi' \in$ C$^+(A, C)$ and $\Psi, \Psi' \in$ C$^+(B, C)$, we may regard the copairings

$$[(f, (\phi_i)_{i\in I'}),\ (g, (\psi_i)_{i\in I''})] \quad \text{and} \quad [(f', (\phi'_i)_{i\in I'}),\ (g', (\psi'_i)_{i\in I''})]$$

as morphisms $(h, (\xi_i)_{i\in I})$ and $(h', (\xi'_i)_{i\in I})$ from $A \oplus B$ to C, where $I = I' \cup I''$. Specifically, the indexing function h underlying $(\xi_i)_{i\in I}$ is defined as

$$h(i) = \begin{cases} f(i) & \text{if } i \in I', \\ g(i) & \text{if } i \in I'', \end{cases}$$

the copairing of f and g in Set. Then, we have:

$$\begin{aligned}
&d([\Phi, \Psi], [\Phi', \Psi']) \\
&= d(\xi, \xi') = \sup\{d'(\xi_i, \xi'_i) \mid i \in I' \cup I''\} \\
&= \sup\{\{d'(\phi_i, \phi'_i) \mid i \in I'\} \cup \{d'(\psi_i, \psi'_i) \mid i \in I''\}\} \\
&= \sup\{\sup\{d'(\phi_i, \phi'_i) \mid i \in I'\}, \sup\{d'(\psi_i, \psi'_i) \mid i \in I''\}\} && \text{(Lemma 3.3.5)} \\
&= \sup\{d(\Phi, \Phi'), d(\Psi, \Psi')\}
\end{aligned}$$

□

We note that more powerful machinery, based on advanced categorical structures such as presheaves, could be employed to prove the fact that the coproduct cocompletion of a category C forms a Met-category with binary products. However, since categories are used here as a tool rather than being the main focus of the thesis, we have opted for this more down-to-earth description, which assumes from the reader less advanced knowledge of category theory.

3.5 Discussion: generalizing to quantales

Que referencia é que dou para o artigo se ele (ainda) não está em lado nenhum?

This work can be generalized to other quantales. Indeed, initial steps in this direction have already been established in **[BMQL25]**. Readers unfamiliar with quantales may also consult [31] for an accessible introduction to the concept. This generalization also explains why in the equational system that we introduced, we chose to use the expression $q+\sup\{r, s\}$ rather than $\sup\{q + r, q + s\}$, given that these are equal.

The fact is that at the level of arbitrary quantales they need not to be the same. For instance, consider the quantale $\mathcal{P}(\Sigma^*)$, where Σ is a finite non-empty set of symbols (*i.e.*, the powerset of all finite lists over Σ) [31]. In this quantale, the associative operation $\otimes$ and the infimum/meet are defined as follows:

$$I \otimes J = \{i +\!\!+\, j \,|\, i \in I, j \in J\} \quad \text{and} \quad \inf\{I, J\} = I \cap J$$

where $I, J \in \mathcal{P}(\Sigma^*)$ and $+\!\!+$ denotes list concatenation. Consider the sets $X = \{\varepsilon, a\}$, $Y = \{a\}$, and $Z = \{aa\}$, where ε denotes the empty string over Σ (i.e., for any $s \in \Sigma^*$, we have $\varepsilon +\!\!+\, s = s = s +\!\!+\, \varepsilon$). Then:

$$X \otimes \inf\{Y, Z\} = X \otimes \emptyset = \emptyset,$$
$$\inf\{X \otimes Y,\ X \otimes Z\} = \{a, aa\} \cap \{aa, aaa\} = \{aa\}.$$

Consequently, $X \otimes \inf\{Y, Z\} \neq \inf\{X \otimes Y, X \otimes Z\}$.

Moreover, it becomes clear after inspecting the soundness proof that it is the expression $q + \sup\{r, s\}$ that arises naturally.

Part II

Applications

Chapter 4

Probabilistic Programming

Computer science and probability theory have shared a fruitful relationship since the early days [32]. Over the years, probabilistic algorithms have emerged as powerful tools across diverse domains—from machine learning [33] and robotics [34] to computational linguistics [35]. These algorithms also play a pivotal role in modern cryptography, particularly in public-key systems [36], and tackle computationally intractable problems [37].

The growing influence of probabilistic methods has also spurred the development of probabilistic programming languages, both concrete and abstract. Early examples include higher-order probabilistic languages like Church [38], while more recent innovations, such as Anglican [39], continue to expand the expressive power and practicality of probabilistic programming.

In this setting, Crubillé and Dal Lago introduced the notion of a *context distance* in [40, 41], as a metric analogue of Morris' context equivalence. In Morris's framework, two programs are said to be *context equivalent* if their observable behavior—that is, what an external observer can measure during execution—is identical in any context. This distance was first developed for an affine λ-calculus and later extended to a more general setting that, for instance, allows copying. In [1], the authors reason about approximate equivalence in the probabilistic setting using the *operator norm*. The latter induces a metric on the space of probabilistic programs (which are interpreted as short maps between Banach spaces).

This chapter begins with an introduction of Banach spaces based on [42–44]. We then show that Ban, the category of Banach spaces and short maps, is a model of our metric lambda calculus (Definition 3.2.3). The next section introduces the fundamentals of measure theory, a key component in defining the semantics of probabilistic programs, drawing inspiration primarily from [43, 45, 46]. Finally, using our calculus, we study approximate equivalence in the context of a random walk on the real line.

4.1 Banach spaces

Before introducing the category of Banach spaces and proving it is a model, we first introduce some preliminary concepts in this setting.
The letters V, W will often be used to refer to vector spaces and $\mathcal{F}$ denotes the field of scalars of a vector space.

Definition 4.1.1. 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: $\|av\| = |\alpha|\|v\|$ for all $v \in V$ $\alpha \in \mathcal{F}$;

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

Definition 4.1.2. A vector space together with a norm is called a *normed vector space*.

Every normed space may be regarded as a metric space (Definition 2.3.6), in which the distance $d(v, w)$ between vectors v and w is $\|v - w\|$.

Definition 4.1.3. Let V and W be normed vector spaces, and let $T : V \to W$ be a linear operator. The *operator norm*, denoted by $\|\cdot\|_{\mathsf{op}}$, is defined as

$$\|T\|_{\mathsf{op}} = \sup\{\|T(v)\| : v \in V, \ \|v\| = 1\}.$$

If $\|T\|_{\mathsf{op}} < \infty$ we say that T is a *bounded operator*; otherwise, if $\|T\|_{\mathsf{op}} = \infty$ we say that T is *unbounded*. We denote by $\mathcal{B}(V, W)$ the vector space of all bounded linear operators from V to W, and we write $\mathcal{B}(V)$ for $\mathcal{B}(V, V)$.

Lemma 4.1.4. *[43, Lemma 6.4] Let $T : V \to W$ be a bounded linear operator between normed spaces. Then the following statements hold:*

1. *For every $v \in V$, we have $\|T(v)\| \leq \|T\|_{\mathsf{op}} \cdot \|v\|$.*

2. *The operator T is continuous (w.r.t the metric) if and only if it is bounded.*

Definition 4.1.5. Let V and W be normed vector spaces, and let $T : V \to W$ be a linear operator. T is called a *short map* if $\|T\|_{\mathsf{op}} \leq 1$.

Definition 4.1.6. (*Cauchy sequence*) Suppose d is a metric on a set X. A sequence $\{x_n\} \subset X$ is called a *Cauchy sequence* if, for every $\varepsilon > 0$, there exists an integer $N \in \mathbb{N}$ such that $d(x_m, x_n) < \varepsilon$ for all $m, n > N$. The metric d is said to be *complete* if every Cauchy sequence in X converges to a point in X

Definition 4.1.7. A *Banach space* is a normed vector space that is complete with respect to the metric induced by its norm. In other words, every Cauchy sequence in the space must converge to a limit within the space.

Definition 4.1.8 (Algebraic Tensor Product)**.** The tensor product of vector spaces V and W is a vector space $V \odot W$ together with a bilinear function (*i.e.*, linear in both variables) $\otimes : V \times W \to V \odot W$ such that for every bilinear function $g : V \times W \to R$, there exists a unique linear function $h : V \odot W \to R$ such that $g = h \circ \otimes$.

$$\begin{tikzcd}
V \times W \arrow[r, "\otimes"] \arrow[dr, "g"'] & V \odot W \arrow[d, dashed, "h"] \\
& R
\end{tikzcd}$$

The function $\otimes$ usually remains anonymous and is written as $(a, b) \mapsto a \otimes b$.
It follows that arbitrary elements of $V \odot W$ take the form

$$\sum_{i=1}^{n} \alpha_i (v_i \otimes w_i)$$

for $\alpha_i \in \mathcal{F}$, $v_i \in V$, and $w_i \in W$.
The tensor product extends in particular to linear maps. If $f_1 : V_1 \to W_1$ and $f_2 : V_2 \to W_2$ are linear maps, then there is a unique linear map $f_1 \odot f_2 : V_1 \odot V_2 \to W_1 \odot W_2$ that satisfies

$$(f_1 \odot f_2)(v_1 \otimes v_2) = f_1(v_1) \otimes f_2(v_2)$$

for all $v_1 \in V_1, v_2 \in V_2$.

$$\begin{tikzcd}
V_1 \times V_2 \arrow[r] \arrow[d] & V_1 \odot V_2 \arrow[d, dashed] \\
W_1 \times W_2 \arrow[r] & W_1 \odot W_2
\end{tikzcd}$$

Ver cena de o que é que precisa de refs.

Definition 4.1.9. [44, Chapter 2.1] Let V and W be Banach spaces. Let u be any element of $V \odot W$. The *projective norm*, denoted $\|\cdot\|_\pi$, is defined by:

$$\|u\|_\pi = \inf\left\{ \sum_{i=1}^{n} \|v_i\| \, \|w_i\| \;\middle|\; u = \sum_{i=1}^{n} v_i \otimes w_i \right\}.$$

Definition 4.1.10. [44, Chapter 2.1] Let V and W be Banach spaces. The *projective tensor product* of V and W, denoted $V \widehat{\otimes}_\pi W$, is the completion ([47]) of the algebraic tensor product $V \otimes W$ with respect to the projective norm $\|\cdot\|_\pi$.

4.2 The category Ban

In this section, leveraging results from [1], we prove that the category of Banach spaces with short maps forms a suitable model of our calculus.

Definition 4.2.1. The category Ban is the category of Banach spaces and short maps. It has a (symmetric) monoidal structure where the tensor product is the projective tensor $\widehat{\otimes}_\pi$. The (binary) coproduct of two Banach spaces V, W is given by their direct sum equipped with the norm $\|(v, w)\| = \|v\| + \|w\|$.

Lemma 4.2.2. *Let V, W and U be Banach spaces. Let $T : V \to U$ and $S : W \to U$ be short maps. Then, it holds that*

$$\|[T, S]\|_{\mathit{op}} \leq \sup\{\|T\|_{\mathit{op}}, \|S\|_{\mathit{op}}\}$$

Não sei se chega, mas tb não sei justificar muito bem sem me meter em cenas convexas (extreme points) e não me apetecia muito.

Proof. We calculate,

$$\begin{aligned}
\|[T, S]\|_{\mathrm{op}} &= \sup\left\{\|[T, S](v_0)\| \mid \|(v_0)\| = 1\right\} \\
&= \sup\left\{\|[T, S](v, w)\| \mid \|(v, w)\| = 1\right\} \\
&= \sup\left\{\|T(v) + S(w)\| \mid \|v\| + \|w\| = 1\right\} \\
&\leq \sup\left\{\|T(v)\| + \|S(w)\| \mid \|v\| + \|w\| = 1\right\} \\
&= \sup\left\{\|v\| \cdot \|T\left(v/\|v\|\right)\| + \|w\| \cdot \|S\left(w/\|w\|\right)\| \mid \|v\| + \|w\| = 1\right\} \\
&= \sup\left\{\sup\left\{\|T(v)\| \mid \|v\| = 1\right\}, \sup\left\{\|S(w)\| \mid \|w\| = 1\right\}\right\} \\
&= \sup\{\|T\|_{\mathrm{op}}, \|S\|_{\mathrm{op}}\}
\end{aligned}$$

In the second to last step we observe that the expression

$$\|v\| \cdot \|T\,(v/\,\|v\|)\| + \|w\| \cdot \|S\,(w/\,\|w\|)\|$$

is maximized when either $\|v\| = 1$ or $\|w\| = 1$.

□

Theorem 4.2.3. *[1, Theorem 4.3] The category* Ban *is a symmetric monoidal* Met*-category.*

Theorem 4.2.4. *The category* Ban *is a symmetric monoidal* Met*-category with binary coproducts.*

Proof. Considering the definition of a symmetric monoidal Met-category with binary coproducts, this follows from Theorem 4.2.3 and Lemma 4.2.2. □

4.3 Measure theory

Probabilistic computation involves running programs that incorporate randomness, leading to output behaviors characterized by probability distributions rather than deterministic outcomes. To effectively understand and analyze these programs, it is essential to have a solid foundation in reasoning about probability distributions. That is where measure theory comes into play. In this work, we only consider finite measures; hence, the term "measure" implicitly refers to a finite measure unless stated otherwise.

4.3.1 What is measure theory?

Throughout history, mathematicians sought to extend the ideas of length, area, and volume. The most effective known way to generalizing these concepts is through the idea of a measure. Abstractly, a measure is a function defined on subsets of a set with additive properties mirroring length, area, and volume.

We begin with a simple example inspired by [48] to develop an intuition for the concepts of measure and measure space. Imagine an open field S covered in snow after a storm. Suppose we wish to measure the amount of snow accumulated in as many field regions as possible. Assume we have accurate tools for measuring snow over standard geometric shapes like triangles, rectangles, and circles. We can approximate irregularly shaped regions using combinations of these standard shapes and then apply a limiting process to assign a consistent measure to such regions. Let $\mathcal{B}$ denote the collection of subsets of S that are deemed *measurable*, let $\lambda(A)$ represent the amount of snow in each $A \in \mathcal{B}$, and let A^c denote the complement of a set A.

For this framework to make sense, it is reasonable to require that $\mathcal{B}$ and $\lambda(\cdot)$ satisfy the following properties:

Properties of $\mathcal{B}$:

1. If $A \in B$, then the complement $A^c \in \mathcal{B}$ (*i.e.*, if we can measure the snow on a set A, and we know the total amount on S, then we can determine the snow on the remaining part A^c).

2. If $A_1, A_2 \in \mathcal{B}$, then $A_1 \cup A_2 \in \mathcal{B}$ (*i.e.*, if we can measure the snow on two regions A_1 and A_2, we should also be able to measure it on their union).

3. If $\{A_n\}_{n\geq 1} \subset \mathcal{B}$ is an increasing sequence, *i.e.*, $A_n \subset A_{n+1}$ for all n, then $\bigcup_{n=1}^{\infty} A_n \in \mathcal{B}$ (*i.e.*, if each set in a increasing sequence of regions is measurable, then their limit—the union—should also be measurable).

4. The collection $\mathcal{B}$ contains a base class C of simple, well-behaved sets (*e.g.*, triangles, rectangles, circles) for which measurement is initially defined.

Properties of $\lambda(\cdot)$:

1. $\lambda(A) \geq 0$ for all $A \in \mathcal{B}$ (*i.e.*, the amount of snow on any set must be nonnegative).

2. If $A_1, A_2 \in \mathcal{B}$ and $A_1 \cap A_2 = \emptyset$, then $\lambda(A_1 \cup A_2) = \lambda(A_1) + \lambda(A_2)$ (*i.e.*, The total amount of snow over two non-overlapping regions is just the sum of the snow in each region. This characteristic of λ is known as *finite additivity*.)

3. If $\{A_n\}_{n\geq 1} \subset \mathcal{B}$ is an increasing sequence, *i.e.*, $A_n \subset A_{n+1}$ then $\lambda(\lim_{n\to\infty} A_n) = \lambda\left(\bigcup_{n=1}^{\infty} A_n\right) = \lim_{n\to\infty} \lambda(A_n)$ (*i.e.*, if a set can be approximated by an increasing sequence of measurable sets $\{A_n\}_{n\geq 1}$, then $\lambda(A) = \lim_{n\to\infty} \lambda(A_n)$. This is known as *monotone continuity from below*).

4.3.2 Measurable spaces and measures

Remarkably, these intuitive conditions give rise to a profoundly versatile and far-reaching theoretical framework. The requirements imposed on $\mathcal{B}$ and λ can more formally be stated as follows:

Properties of $\mathcal{B}$:

1. $\emptyset \in \mathcal{B}$.
2. $A \in \mathcal{B} \implies A^c \in \mathcal{B}$.
3. $A_1, A_2, \dots \in \mathcal{B} \implies \bigcup_i A_i \in \mathcal{B}$ (this is known as *closure under countable unions*).

Properties of λ:

1. $\lambda(\cdot) \geq 0$ and $\lambda(\emptyset) = 0$.
2. If $\{A_n\}_{n\geq 1} \subset \mathcal{B}$ is a sequence of pairwise disjoint sets (i.e., $A_i \cap A_j = \emptyset$ for $i \neq j$), then $\lambda\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \lambda(A_n)$ (this is known as *countable additivity*).

A collection $\mathcal{B}$ of subsets of S satisfying the above conditions for B is designated a *σ-algebra*. Similarly, a set function λ defined on a σ-algebra $\mathcal{B}$ that fulfills the above properties for λ qualifies as a *measure*.

Definition 4.3.1. A *σ-algebra $\mathcal{B}$* on a set S is a collection of subsets of S that includes the empty set, is closed under complementation with respect to S, and is closed under countable unions.

Definition 4.3.2. The *Borel σ-algebra* of a metric space is the smallest family of sets that includes the closed sets and is closed under countable intersections and countable unions. Elements of the Borel σ-algebra are known as *Borel sets*.

Definition 4.3.3. The pair $(S, \mathcal{B})$ where $\mathcal{B}$ is a σ-algebra constitutes a *measurable space*. The elements of $\mathcal{B}$ are called the *measurable sets* of the space.

Definition 4.3.4. A *measure* on the measurable space $(S, \mathcal{B})$ is a function $\mu : \mathcal{B} \to \mathbb{R}$ that is countably additive and satisfies $\mu(\emptyset) = 0$. A measure on $(S, \mathcal{B})$ is called a *probability measure* if $\mu(S) = 1$.

In the context of probability theory, such measures are often referred to as *distributions*. In what follows, we will use the terms measure and distribution interchangeably.
One of the most important measures is the Lebesgue measure on the real line (*i.e.*, the length in $\mathbb{R}$), and its generalizations to $\mathbb{R}^n$. It is characterized as the unique measure on the Borel sets

whose value on every interval is its length. That is, for any interval $[a, b] \subset \mathbb{R}$, $\lambda([a, b]) = b - a$. The Lebesgue measure is *translation-invariant*, meaning that shifting a set $B \in \mathbb{R}^n$ by a fixed vector, does not change its measure.

For any $s \in S$, the *Dirac measure* (also known as the *Dirac delta* or *point mass* at s) is the probability measure defined by

$$\delta_s(B) = \begin{cases} 1, & \text{if } s \in B, \\ 0, & \text{if } s \notin B. \end{cases} \quad \text{for all } B \in \mathcal{B}$$

A measure is called *discrete* if it is represented as a countable weighted sum of Dirac measures. In particular, a *convex combination* of Dirac measures yields a discrete probability measure. These are of the form $\sum_i \alpha_i \delta_i$, where $\alpha_i \geq 0$, and the weights satisfy $\sum_i \alpha_i = 1$.

On the other hand, a measure μ on a measurable space $(S, \mathcal{B})$ is called *continuous* if it assigns zero measure to all singleton sets, i.e., $\mu(\{s\}) = 0$ for every $s \in S$. An example of a continuous measure is the *Lebesgue measure* on $\mathbb{R}^n$ (for $n \in \mathbb{N}$).

Definition 4.3.5. Let $(S, \mathcal{B}_S)$ and $(T, \mathcal{B}_T)$ be measurable spaces. A function $f : S \to T$ is said to be *measurable* if for every measurable subset $B \in \mathcal{B}_T$, the preimage $f^{-1}(B) \in \mathcal{B}_S$.

Definition 4.3.6. Let $(S, \mathcal{B}_S)$ and $(T, \mathcal{B}_T)$ be measurable spaces. Given a measurable function $f : (S, \mathcal{B}_S) \to (T, \mathcal{B}_T)$ and a measure μ on $\mathcal{B}_S$, the *pushforward measure* $f_*(\mu)$ on $\mathcal{B}_T$ is defined by:

$$f_*(\mu)(B) = \mu(f^{-1}(B)), \quad B \in \mathcal{B}_T.$$

Definition 4.3.7. The *Lebesgue integral* generalizes the familiar Riemann integral. Consider a measurable space $(S, \mathcal{B})$ and a bounded measurable function $f \colon S \to \mathbb{R}$, with upper and lower bounds M and m, respectively. The *Lebesgue integral* of f with respect to a measure $\mu \colon \mathcal{B} \to \mathbb{R}$, denoted $\int f \, d\mu$, is defined as the limit of finite weighted sums of the form:

$$\sum_{i=0}^{n} f(s_i)\mu(B_i),$$

where $\{B_0, \ldots, B_n\}$ forms a measurable partition of S, and within each B_i, the variation of f does not exceed $(M - m)/n$. Here, $s_i \in B_i$ for each i, and the limit is taken over increasingly refined partitions.

In the case of a finite discrete space $n = \{1, 2, \ldots, n\}$, the Lebesgue integral simplifies to a weighted sum:

$$\int f \, d\mu = \sum_{i=1}^{n} f(i)\mu(i).$$

Given two measurable spaces $(S_1, \mathcal{B}_1)$ and $(S_2, \mathcal{B}_2)$, their product is the measurable space $(S_1 \times S_2, \mathcal{B}_1 \otimes \mathcal{B}_2)$, where $S_1 \times S_2$ is the cartesian product and $\mathcal{B}_1 \otimes \mathcal{B}_2$ is the σ-algebra generated by all measurable rectangles $B_1 \times B_2$ with $B_1 \in \mathcal{B}_1$ and $B_2 \in \mathcal{B}_2$:

$$\mathcal{B}_1 \otimes_{\text{meas}} \mathcal{B}_2 := \sigma\left(\{B_1 \times B_2 \mid B_1 \in \mathcal{B}_1,\ B_2 \in \mathcal{B}_2\}\right).$$

Measures on this product space are called *joint distributions*, and are uniquely determined by their values on measurable rectangles due to the inductive structure of the product σ-algebra. *Product measures* are a particular class of joint distributions and are defined from measures, as detailed next.

Definition 4.3.8. Let $(S_1, \mathcal{B}_1)$ and $(S_2, \mathcal{B}_2)$ be measurable spaces, and let μ_1 and μ_2 be measures on these spaces, respectively. The *product measure* $\mu_1 \otimes_{\textbf{meas}} \mu_2$ is defined on measurable rectangles by

$$(\mu_1 \otimes_{\text{meas}} \mu_2)(B_1 \times B_2) = \mu_1(B_1)\mu_2(B_2).$$

This definition extends uniquely to a joint distribution $\mu_1 \otimes_{\text{meas}} \mu_2 : \mathcal{B}_1 \otimes_{\text{meas}} \mathcal{B}_2 \to \mathbb{R}$, and reflects the notion of probabilistic independence: sampling from $\mu_1 \otimes_{\text{meas}} \mu_2$ is equivalent to independently sampling from μ_1 and μ_2.

4.3.3 Spaces of Measures

The set of all finite measures on a measurable space $(S, \mathcal{B})$ will be denoted by $\mathcal{M}(S, \mathcal{B})$, or simply $\mathcal{M}S$ when the σ-algebra $\mathcal{B}$ is clear from context. In particular, $\mathcal{M}\mathbb{R}$ denotes the Banach space of finite Borel measures on $\mathbb{R}$.

$\mathcal{M}S$ forms a real vector space, where addition and scalar multiplication are defined pointwise. Specifically, for $B \in \mathcal{B}$, $\mu, \nu \in \mathcal{M}S$, and $\alpha \in \mathbb{R}$, the operations are given by

$$(\mu + \nu)(B) = \mu(B) + \nu(B), \quad (a\mu)(B) = \alpha\mu(B).$$

$\mathcal{M}S$ is also a normed space when equipped with the *total variation norm*.

Definition 4.3.9. A *partition* of a set $B \in \mathcal{B}$ is any finite collection $\{B_1, \ldots, B_n\}$ of pairwise disjoint subsets of B satisfying $\bigcup_{i=1}^n B_i = B$. For a measure μ, the *total variation norm* is defined as

$$\|\mu\| := \sup\left\{\sum_{i=1}^n |\mu(B_i)| : \{B_1, \ldots, B_n\} \text{ is a finite measurable partition of } S\right\}.$$

For positive measures, this reduces to $\mu(S)$, and for probability measures, the norm is 1. The total variation norm turns $\mathcal{M}S$ into a Banach space, meaning it is complete under this norm. The following alternative definition is useful to compute the total variation norm between measures.

Theorem 4.3.10. *[46, Theorem 3.1.1] Let μ be a measure on a measurable space $(S, \mathcal{B})$. Then, there exist disjoint sets $S^-, S^+ \in \mathcal{A}$ such that $S^- \cup S^+ = S$ and for all $B \in \mathcal{B}$, one has*

$$\mu(B \cap S^-) \leq 0 \quad \textit{and} \quad \mu(B \cap S^+) \geq 0.$$

Corollary 4.3.11. *[46, Corollary 3.1.2] Attending to the theorem above, define:*

$$\mu^+(B) := \mu(B \cap S^+), \quad \mu^-(B) := -\mu(B \cap S^-).$$

Then μ^+ and μ^- are nonnegative measures, and we have:

$$\mu = \mu^+ - \mu^-.$$

The measures μ^+ and μ^- have the following properties:

$$\mu^+(B) = \sup\{\mu(B_i) : B_i \subset B,\ B_i \in \mathcal{B}\},$$

$$\mu^-(B) = \sup\{-\mu(B_i) : B_i \subset B,\ B_i \in \mathcal{B}\},$$

for all $B \in \mathcal{B}$.

Definition 4.3.12. Let μ^+ and μ^- be defined as in the corollary above. Then, for a measure μ, the *total variation norm* is defined as

$$\|\mu\| = \mu^+(S) + \mu^-(S).$$

4.4 Case-study : Random Walk

We proceed by presenting a metric λ-theory on which to reason about random walks, as previously discussed, and this will briefly illustrate the synergy between syntax and semantics that our framework provides. Our (only) ground type will be `real` to represent measures over real numbers, *i.e.* we set $[\![\texttt{real}]\!]$ to be the space of measures over the real line, $\mathcal{M}\mathbb{R}$. Concerning operations we take a pre-determined set of coin toss functions *CoinToss*$_p : \mathbb{I} \to \mathbb{I} \oplus \mathbb{I}$

whose interpretation takes the form $[\![\mathit{CoinToss}_p]\!] : \mathbb{R} \to \mathbb{R} \oplus \mathbb{R},\ 1 \mapsto (p, 1-p)$. We also take a pre-determined set of measures

$$m = \{\texttt{unif}(a,b) : \mathbb{I} \to \texttt{real}\} \cup \{\texttt{delta}_{p_1,\ldots,p_n;x_1,\ldots,x_n} : \mathbb{I} \to \texttt{real}\}$$

which are interpreted as

$$[\![\texttt{unif}(a,b)]\!]\,(1) = \texttt{unif}(a,b) \quad \text{and} \quad [\![\texttt{delta}_{p_1,\ldots,p_n;x_1,\ldots,x_n}]\!]\,(1) = \sum_{i=1}^{n} p_i \cdot \delta_{x_i},$$

for all $a, b, p_1, \ldots, p_n, x_1, \ldots, x_n \in \mathbb{Q}$, such that $\sum_i p_i = 1$. Here $\texttt{unif}(0,1) \in \mathcal{M}\mathbb{R}$ is the uniform distribution on the interval $[a,b]$. Note that we are slightly abusing notation by using $\texttt{unif}(a,b)$ both as syntactic and semantic objects. We consider yet addition $+ : \texttt{real}, \texttt{real} \to \texttt{real}$ whose interpretation is given by $\mu \otimes_{\mathsf{meas}} \nu \mapsto +_*(\mu \otimes_{\mathsf{meas}} \nu)$. Finally, we consider a pre-determined set of jumps $j : \mathbb{I} \to (\texttt{real} \multimap \texttt{real})$ interpreted as

$$[\![j]\!]\,(1) = \mu \mapsto +_*(\mu \otimes_{\mathsf{meas}} [\![m_0]\!]\,(*)),$$

where $m_0 \in m$.

Example 4.4.1 (Random walk)**.** In general terms, a *random walk* on $\mathbb{R}$ is a stochastic process in which a particle—the walker—starts at an initial position and repeatedly "tosses a coin" (possibly biased) to decide whether to move left or right.
Given jumps jl, jr we can describe a single step of the random walks a follows,

$$\textbf{step} = - \triangleright \text{case } \mathit{CoinToss}_p(*)\{\text{inl}(x) \Rightarrow jl(x); \text{inr}(y) \Rightarrow jr(y)\} : \texttt{real} \multimap \texttt{real}$$

Given an argument r representing the walker's current position, the program **step** performs a random jump: with probability p, the jump is sampled from the underlying distribution of jl; with probability $1-p$, it is sampled from the distribution of jr.
Now, consider the λ-term **apply-n** $= \lambda f_1, \ldots, f_n, r.\ f_1(f_2(\ldots(f_n(r))))$ which operationally speaking sequences n terms given as input. The term that follows represents a n-step walk.

$$\textbf{rwalk} = \textbf{apply-n step} \ldots \textbf{step}\ (\texttt{delta}_{1;0}) : \texttt{real}$$

Recall the interpretation of the jump operations. It is straightforward to prove that the following axiom is sound for each of them:

$$j(*) =_0 \lambda x.\ +(x, m_0(*)).$$

Now, consider we set the interpretations of $[\![jl]\!], [\![jr]\!] : \mathbb{R} \to (\mathcal{M}\mathbb{R} \multimap \mathcal{M}\mathbb{R})$ to be:

$$[\![jl]\!](1) = \mu \mapsto +_*(\mu \otimes \texttt{unif}(0,-1)) \qquad [\![jr]\!](1) = \mu \mapsto +_*(\mu \otimes \texttt{unif}(1,0))$$

Operationally jl corresponds to a jump to the left with magnitude between 0 and 1, and analogously for jr. Suppose we have another jump $[\![jr^\delta]\!] : \mathbb{R} \to (\mathcal{M}\mathbb{R} \multimap \mathcal{M}\mathbb{R})$ whose interpretation is that of jr except for the fact that $\texttt{unif}(0,1)$ is replaced by $\texttt{unif}(0, 1+\delta)$. What will be the effect on the random walk when replacing jr by jr^δ? Observe that one can put an upper bound between $jr(*)$ and $jr^\delta(*)$ via the previous axioms and an upper bound between the terms $\texttt{unif}(0,1)(*)$ and $\texttt{unif}(0,1+\delta)(*)$. The latter upper bound is obtained *semantically* by computing the norm $\|\texttt{unif}(0,1) - \texttt{unif}(0,1+\delta)\|$. First, attending to Definition 4.3.12, we have,

$$\begin{aligned}&\|\texttt{unif}(0,1) - \texttt{unif}(0,1+\delta)\| \\ &= (\texttt{unif}(0,1) - \texttt{unif}(0,1+\delta))^+(\mathbb{R}) + (\texttt{unif}(0,1) - \texttt{unif}(0,1+\delta))^-(\mathbb{R})\end{aligned}$$

and proceed by computing the left-hand side of the addition,

$$\begin{aligned}&(\texttt{unif}(0,1) - \texttt{unif}(0,1+\delta))^+(\mathbb{R}) \\ &= \sup\{\texttt{unif}(0,1)(U) - \texttt{unif}(0,1+\delta)(U) \mid U \subseteq \mathbb{R}\} \\ &= \sup\{\texttt{unif}(0,1)(U \cap [0,1]) - \texttt{unif}(0,1+\delta)(U \cap [0,1]) \\ &\quad - \texttt{unif}(0,1+\delta)(U \cap (1, 1+\delta]) \mid U \subseteq \mathbb{R}\} \\ &= \sup\left\{\left(1 - \frac{1}{1+\delta}\right)\texttt{unif}(0,1)(U \cap [0,1]) - \texttt{unif}(0,1+\delta)(U \cap (1,1+\delta]) \mid U \subseteq \mathbb{R}\right\} \\ &= 1 - \frac{1}{1+\delta}\end{aligned}$$

It follows from an analogous reasoning the right-hand side of the addition will be $\frac{\delta}{1+\delta}$ and therefore the norm will be $2 \cdot \frac{\delta}{1+\delta}$.

Additionally, suppose $\mathit{CoinToss}_p$ is replaced by $\mathit{CoinToss}_q$. We calculate:

$$\begin{aligned}\|[\![\mathit{CoinToss}_p(*)]\!] - [\![\mathit{CoinToss}_q(*)]\!]\| &= \|(p, 1-p) - (q, 1-q)\| \\ &= \|(p-q, q-p)\| = 2|p-q|\end{aligned}$$

Then as our final step we proceed *syntactically* via our metric deductive system, as follows.

$$
\begin{aligned}
&\text{case } \mathit{CoinToss}_p(*) \text{ of } \mathrm{inl}(x) \Rightarrow jl(x); \mathrm{inr}(y) \Rightarrow jr(y) \\
&=_0 \text{ case } \mathit{CoinToss}_p \text{ of } \mathrm{inl}(x) \Rightarrow jl(y); \mathrm{inr}(y) \Rightarrow x \text{ to } * \,.\, jr(*) \\
&=_{2\cdot\left(|p-q|+\frac{\delta}{1+\delta}\right)} \text{ case } \mathit{CoinToss}_q \text{ of } \mathrm{inl}(x) \Rightarrow jl(y); \mathrm{inr}(y) \Rightarrow x \text{ to } * \,.\, jr^{\delta}(*) \\
&=_0 \text{ case } \mathit{CoinToss}_q \text{ of } \mathrm{inl}(x) \Rightarrow jl(x); \mathrm{inr}(y) \Rightarrow jr^{\delta}
\end{aligned}
$$

Thus if **rwalk** is the random walk that involves jump jl and *CoinToss*$_p$ and **rwalk'** the random walk that involves jump jl^{δ} and *CoinToss*$_q$ we deduce from the framework the metric equation,

$$\textbf{rwalk} =_{2n\cdot\left(|p-q|+\frac{\delta}{1+\delta}\right)} \textbf{rwalk'}$$

which will converge to 0 as δ and $|p-q|$ tends to 0.

The same reasoning applies to alternative interpretations of jl and jr. For example, consider:

$$[\![jl]\!]\,(1) = \mu \mapsto +_*\left(\mu \otimes \sum_{i=1}^{n} p_i \cdot \delta_{-x_i}\right) \qquad [\![jr]\!]\,(1) = \mu \mapsto +_*\left(\mu \otimes \sum_{i=1}^{n} p_i \cdot \delta_{x_i}\right).$$

Operationally, this means that jl performs a jump to the left, landing at position $-x_i$ with probability p_i, and jl behaves analogously, jumping to x_i with the same probabilities. Now, consider another jump jl^{q_i} whose interpretation corresponds to that of jl xcept for the fact that $\sum_{i=1}^{n} p_i \cdot \delta_{-x_i}$ is replaced with $\sum_{i=1}^{n} q_i \cdot \delta_{-x_i}$. Given Definition 4.3.9, we compute,

$$
\begin{aligned}
&\left\|\sum_i p_i\delta_{-x_i} - \sum_i q_i\delta_{-x_i}\right\| \\
&= \sup\left\{\sum_{i=1}^{n}\left|\left(\sum_i p_i\delta_{-x_i} - \sum_i q_i\delta_{-x_i}\right)(B_i)\right| \;\middle|\; B_i \in \mathbb{R}, B_i \cap B_j = \emptyset i, \neq j, n \in \mathbb{N}\right\} \\
&= \sum_i |p_i - q_i|
\end{aligned}
$$

Here, we use the inequality

$$\left|\sum_{i=1}^{n} \alpha_i\right| \leq \sum_{i=1}^{n} |\alpha_i|, \qquad \text{for all } n \in \mathbb{N}.$$

Applying the same reasoning as above, if **rwalk** is the random walk that involves jump jl and *CoinToss*$_p$ and **rwalk'** the random walk that involves jump jl^{q_i} and *CoinToss*$_q$, we obtain

$$\textbf{rwalk} =_{n\cdot\left(2|p-q|+\sum_i |p_i-q_i|\right)} \textbf{rwalk'}.$$

Chapter 5

Quantum computation

Quantum computing dates back to 1982 when the Nobel laureate Richard Feynman proposed the idea of constructing computers based on quantum mechanics principles to efficiently simulate quantum phenomena [49]. 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 [20]. 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. This result underscores the promise of the quantum computing paradigm.

Quantum computing and the need for quantitative reasoning. While hardware advancements have brought the scientific community closer to realizing the transformative potential of quantum computing, the ultimate goal is yet to be accomplished. A 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 [50]. Unfortunately, general-purpose error correction techniques [51–53] 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 [54]. 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 [55] and noise-adaptive optimization [56]. Additionally, current quantum computers only support a restricted, albeit universal, set of quantum operations. As a result, non-native operations must be decomposed into sequences of native operations before execution [57, 58]. The assessment of these compiler transformations necessitates a comparison of the error bounds between the source and compiled quantum programs, which calls for the development of appropriate notions of approximate program equivalence.

As previously noted, Shor's algorithm has played a pivotal role in sparking heightened interest within the scientific community towards quantum computing research. Several quantum programming languages have surfaced over the past 25 years [59, 60]. Among them, we highlight Selinger's and Valirion's work. In 2004, Selinger introduced a first-order functional language for quantum computation, QPL, along with its denotational semantics [21]. Building on this, Selinger and Valiron later developed a higher-order functional language for quantum computation—commonly referred to as a quantum lambda calculus. They first presented a version with classical control and its operational semantics in [61]. This was followed by a denotational semantics for a fragment of the language in [62]. In subsequent work, they extended the quantum lambda calculus to include recursion and infinite types, along with its operational semantics [63]. Later, they proposed an alternative approach to its denotational semantics [64].

These works adopt Schrödinger's picture, in which quantum programs are interpreted as maps between quantum states (*i.e.*, density operators). In contrast, [22, 65] consider Heisenberg's picture, in which programs are modeled as maps between observables (*i.e.*, self-adjoint operators). Particularly, [22] presents a model based on W^*-algebras, which can be viewed as an infinite-dimensional extension of [21]. Moreover, [22] presents a model of Selinger and Valiron's quantum lambda calculus [61, 63, 66], also based on W^*-algebras, and proves the model's adequacy.

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 [67]. 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. For example, [68] and [69] 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,λ)-diamond norm, which analyzes the output error given that the input quantum state satisfies some quantum predicate Q to degree λ. However, it does not specify any practical method for obtaining non-trivial quantum predicates. In fact, the methods used in [68] 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 [1], using linear λ-calculus as basis. A notion of approximate equivalence is then integrated in the calculus via the so-called diamond norm, which induces a metric on the space of quantum programs (seen semantically as completely positive trace-preserving super-operators) [70].

The first two sections of this chapter present mathematical and quantum computing preliminaries necessary for understanding the theory of quantum computation. This introduction to quantum computing draws primarily from [70, 71], while the mathematical foundations are also based on [72–74]. The next section introduces core concepts from functional analysis that are essential for understanding W^*-algebras, based on [42, 43]. This is followed by a section presenting the fundamentals of W^*-algebras, drawing primarily from [75–77]. We then show that both Selinger's category Q of quantum operations (i.e., completely positive, trace-nonincreasing super-operators) [21], and Cho's category $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$, the opposite category of W^*-algebras with normal, completely positive, subunital maps [22], are first-order models of our calculus. Finally, the last section provides a few illustrative examples in the setting of quantum information.

5.1 Hilbert Spaces

It is impossible to present the theory of quantum computation without introducing some concepts of theory of Hilbert spaces and operators. This section briefly overwiews of the aspects of Hilbert spaces that are most pertinent to the study of quantum computation.

Convention 5.1.1. In this section and the one that follows, vector spaces are assumed to be finite-dimensional, unless otherwise stated.

5.1.1 Inner product

Definition 5.1.2. An *inner product* $\langle \cdot, \cdot \rangle$ on a vector space V is a function from a mapping $V \times V$ to the field of scalars, $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{C}$, that satisfies the following properties for all $v, w, w_1, \dots, w_n \in V$ and $\alpha_1, \dots, \alpha_n \in \mathbb{C}$.

1. Linearity in the second argument,

$$\left\langle v, \sum_{i=1}^{n} \alpha_i w_i \right\rangle = \sum_{i=1}^{n} \alpha_i \langle v, w_i \rangle.$$

2. $\langle v, w \rangle = \overline{\langle w, v \rangle}$, where $\overline{(-)}$ is the complex conjugate operation.

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

Example 5.1.3. For instance, the inner product $\langle v, w \rangle$ of two vectors $v = (\alpha_1, \dots, \alpha_n), w = (\beta_1, \dots, \beta_n) \in \mathbb{C}^n$ is defined as

$$\langle v, w \rangle = \sum_i \overline{\alpha}_i \beta_i.$$

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

Definition 5.1.4. A *Hilbert space* $\mathcal{H}$ is an inner product space.

The letters $\mathcal{H}, \mathcal{K}$ will often be used to refer to Hilbert spaces.

Definition 5.1.5. *Positive (semidefinite) operators.* A square operator $A \in \mathcal{B}(\mathcal{H})$ is *positive*, denoted $A \geq 0$, if $\langle v, Av \rangle \geq 0$ for all $v \in \mathcal{B}(\mathcal{H})$.

5.1.2 Trace

Definition 5.1.6. Let $\mathcal{H}$ be an Hilbert space and $A \in \mathcal{B}(\mathcal{H})$ a positive operator (Definition 5.1.5). The trace of A is defined as

$$\mathsf{Tr}(A) := \sum_i \langle Av_i, v_i \rangle \in [0, \infty],$$

where $\{v_i\}$ is an orthonormal basis for $\mathcal{H}$.

The trace is *linear*, $\mathsf{Tr}(A + B) = \mathsf{Tr}(A) + \mathsf{Tr}(B)$, $\mathsf{Tr}(\alpha \cdot A) = \alpha \cdot \mathsf{Tr}(A)$, where $A, B \in \mathcal{B}(\mathcal{H})$, and α is a complex number.

The trace of a square matrix can alternatively be defined as follows.

Definition 5.1.7. The trace of a square matrix $A \in \mathbb{C}^{n \times n}$ is defined to be the sum of its diagonal elements,

$$\mathsf{Tr}(A) = \sum_i A_{ii}.$$

By means of the trace, one defines the inner product of two operators $A, B \in \mathbb{C}^{m \times n}$ as follows

$$\langle A, B \rangle = \mathsf{Tr}(A^\dagger B),$$

where $(-)^\dagger$ denotes the adjoint operation.

5.1.3 Important classes of operators

In a finite-dimensional Hilbert space $\mathcal{H}$ every linear mapping is continuous, hence a bounded operator. For an n-dimensional Hilbert space $\mathcal{H}$, we can identify $\mathcal{B}(\mathcal{H})$ with the space $\mathbb{C}^{n \times n}$ of $n \times n$ complex matrices known as square matrices. As a result, linear operators mapping a Hilbert space to itself are known as *square operators*.

The following classes of operators are of particular interest in quantum information theory.

Definition 5.1.8. *Normal operators.* A square operator $A \in \mathcal{B}(\mathcal{H}, \mathcal{K})$ is *normal* if $AA^\dagger = A^\dagger A$.

Definition 5.1.9. *Hermitian operators.* A square operator $A \in \mathcal{B}(\mathcal{H})$ is *hermitian* if $A = A^\dagger$. Every Hermitian operator is a normal operator.

Definition 5.1.10. *Unitary operators.* A square operator $U \in \mathcal{B}(\mathcal{H})$ is *unitary* if $U^\dagger U = UU^\dagger = \mathrm{id}$. 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.

Definition 5.1.11. A *density operator* is a positive (semidefinite) operator with unit trace. By convention, density operators are denoted by the lowercase Greek letter ρ, often accompanied with subscripts or primes to indicate the system or state, *e.g.*, ρ_A, ρ', etc.

Definition 5.1.12. *Isometries.* An operator $A \in \mathcal{B}(\mathcal{H}, \mathcal{K})$ is as isometry if $\|Av\| = \|v\|$ for all elements all elements $v \in \mathcal{H}$.

Definition 5.1.13. *Projectors.* A positive operator $P \in \mathcal{B}(\mathcal{H})$ is a projector if $P^2 = P$.

5.1.4 Spectral theorem

Theorem 5.1.14. *[70, Corollary 1.4] Let $\mathcal{H}$ be a Hilbert space. Every normal operator $A \in \mathcal{L}(\mathcal{H})$ can be expressed as a linear combination $\sum_{i=1}^{n} \lambda_i v_i v_i^\dagger$ where the set $\{v_1, \ldots, v_n\}$ is an orthonormal basis on $\mathcal{H}$.*

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

$$f(A) = \sum_i f(\lambda_i) v_i v_i^\dagger \tag{5.1}$$

where $A = \sum_i \lambda_i v_i v_i^\dagger$ is the spectral decomposition of A.
Positive operators are hermitian, and consequently, by the spectral decomposition, have diagonal representation $A = \sum_i \lambda_i v_i v_i^\dagger$, with non-negative eigenvalues λ_i.

5.1.5 Tensor Products and Direct Sums of Hilbert Spaces

Definition 5.1.15. The *direct sum* of two finite-dimensinal Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$, denoted $\mathcal{H} \oplus \mathcal{K}$, is the space of all pairs (v, w) where $v \in \mathcal{H}$ and $w \in \mathcal{K}$.

The inner product in $\mathcal{H} \oplus \mathcal{K}$ is defined as follows:

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

The notation $(-)^{\oplus n}$ will be used to denote the direct sum of a vector space with itself n times.

Definition 5.1.16. Consider two finite dimensinal Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$ with respective basis $v = (\alpha_1, \ldots, \alpha_n)$ and $w = (\beta_1, \ldots, \alpha_m)$. Then $\mathcal{H} \otimes \mathcal{K}$ is an mn dimensional vector space and $v \otimes w$ corresponds to the vector

$$(\alpha_1\beta_1, \ldots, \alpha_1\beta_m, \ldots, \alpha_n\beta_1, \ldots, \alpha_n\beta_m),$$

which is a basis for $\mathcal{H} \otimes \mathcal{K}$. The tensor product of two elements $v = \sum_i \alpha_i v_i$ and $w = \sum_j \beta_j \;\, w_j$ is:

$$v \otimes w = \sum_{i,j} \alpha_i \beta_j \cdot v_i \otimes w_j.$$

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

$$\langle v_1 \otimes w_1, v_2 \otimes w_2 \rangle = \langle v_1, v_2 \rangle \langle w_1, w_2 \rangle,$$

extending to all vectors by linearity.

Definition 5.1.17. Consider two Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$. The tensor product of two operators $A \in \mathcal{B}(\mathcal{H})$ and $B \in \mathcal{B}(\mathcal{K})$ is an operator $A \otimes B \in \mathcal{B}(\mathcal{H} \otimes \mathcal{K})$ defined by the equation

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

The definition of $A \otimes B$ is extended to all elements of $\mathcal{H} \otimes \mathcal{K}$ in the natural way to ensure linearity of the tensor product operator. That is,

$$(A \otimes B) \left(\sum_i \alpha_i \, v_i \otimes w_i \right) = \sum_i \alpha_i \, (Av_i \otimes Bw_i).$$

Suppose A is an $n \times n$ matrix, and B is a $m \times m$ matrix. Then we have the matrix representation:

$$A \otimes B := \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}B & A_{n2}B & \cdots & A_{nn}B \end{bmatrix}$$

In this representation, each block $A_{ij}B$ is a $p \times q$ submatrix obtained by scaling the entire matrix B by the scalar A_{ij}.

For example the tensor product of the matrices $A = \left(\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\right)$ and $B = \mathrm{id}$ is

$$A \otimes B = \begin{bmatrix} 1 \cdot \mathrm{id} & 2 \cdot \mathrm{id} \\ 3 \cdot \mathrm{id} & 0 \cdot \mathrm{id} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \\ 3 & 0 & 4 & 0 \\ 0 & 3 & 0 & 4 \end{bmatrix}. \tag{5.2}$$

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

5.1.6 Useful norms

In this section we only consider finite dimensional Hilbert spaces.

Definition 5.1.18. The *euclidean norm*, $\|\cdot\|_2$, of a vector $v \in \mathcal{H}$ is defined as:

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

Definition 5.1.19. The *trace norm*, $\|\cdot\|_1$, of a matrix $A \in \mathcal{B}(\mathcal{H})$ 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 $d(\rho, \rho') = \|\rho - \rho'\|$.

5.1.7 Infinite-dimensional Hilbert Spaces

In this subsection we lift the restriction to finite-dimensional Hilbert spaces. The definition of inner product (Definition 5.1.2) extends naturally to infinite-dimensional vector spaces, as stated, and the same applies to the definition of trace (Definition 5.1.6).

Definition 5.1.20. A *Hilbert space* $\mathcal{H}$ is an inner product space that is complete with respect to the norm induced by the inner product.

Definition 5.1.21. Let $\mathcal{H}$ be a Hilbert space. An operator $A \in \mathcal{B}(\mathcal{H})$ is *trace class* if Tr$(|A|) < \infty$, where $|A| = \left(A^\dagger A\right)^{1/2}$. We denote by $\mathcal{T}(\mathcal{H})$ the set of trace class operators on $\mathcal{H}$.

If $\mathcal{H}$ is infinite-dimensional, the set $\mathcal{T}(\mathcal{H})$ forms a proper subset of $\mathcal{B}(\mathcal{H})$. In the finite-dimensional case, however, the two spaces coincide and can be identified with one another.

Definition 5.1.22. Let $\mathcal{H}$and $\mathcal{K}$ be Hilbert spaces. We denote by $\mathcal{H} \otimes_2 \mathcal{K}$ the Hilbert space tensor product that is obtained by completing $\mathcal{H} \otimes \mathcal{K}$ w.r.t. the standard inner product

$$\langle w_1 \otimes v_1,\, w_2 \otimes v_2\rangle = \langle w_1, v_2\rangle \cdot \langle w_1, v_2\rangle.$$

5.2 Quantum Computing Preliminaries

The basic unit of information in quantum computation is a *quantum bit* or *qubit* [78]. While a classical bit can be in one of two states, a qubit can be in one of a continuum of states. Qubits are represented using *Dirac notation*, where the ket symbol $|\psi\rangle$ denotes a quantum state ψ. The corresponding bra symbol $\langle\psi|$ denotes the conjugate transpose of the state ψ. In this setting, the inner product of two states $|\psi\rangle$ and $|\phi\rangle$ is denoted $\langle\psi|\phi\rangle$ and is the same as $\langle\psi|\,|\phi\rangle$. The outer product of two states $|\psi\rangle \in \mathcal{H}$ and $|\phi\rangle \in \mathcal{K}$ is the linear operator $|\psi\rangle\langle\phi| : \mathcal{K} \to \mathcal{H}$, defined by

$$(|\psi\rangle\langle\phi|)(|\phi'\rangle) = |\psi\rangle\langle\phi|\phi'\rangle = \langle\phi|\phi'\rangle\,|\psi\rangle\,.$$

Definition 5.2.1. Each isolated quantum system is associated with a Hilbert space, known as the system's *state space*. The system's state is fully characterized by a *state vector*, which is a unit vector within this state space.

5.2.1 The 2-Dimensional Hilbert Space

Definition 5.2.2. The *state* of a single qubit is described by a normalized vector in the 2-dimensional Hilbert space $\mathbb{C}^2$. The states

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

correspond to the classical states 0 and 1, respectively. These states, known as the *computational basis* states, form an orthonormal basis for this vector space.

Definition 5.2.3. Unlike classical bits, a qubit is not restricted to the basis states $|0\rangle$ and $|1\rangle$. It can be in a linear combination of these states, known as a *superposition*. A general qubit state can be written as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

where $\alpha, \beta \in \mathbb{C}$ are called *amplitudes*, and must satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. The values $|\alpha|^2$ and $|\beta|^2$ represent the probabilities of measuring the qubit in the states $|0\rangle$ and $|1\rangle$, respectively.

Informally, a *measurement* (in the computational basis) of a single qubit is an (irreversible) process that projects the qubit state onto $|0\rangle$ with probability $|\alpha|^2$, or $|1\rangle$ with probability $|\beta|^2$, yielding the classical outcome 0 or 1, respectively.

Any normalized qubit state $|\psi\rangle$ can be written (up to a global phase) as

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

where $\theta, \phi, \gamma \in \mathbb{R}$. The global phase factor $e^{i\gamma}$ has no observable effect on the outcome of measurements and is often disregarded. Thus, the state is usually represented as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle . \tag{5.3}$$

The above parametrization defines a point on the unit sphere in $\mathbb{R}^3$, known as the *Bloch sphere*. The angles θ and ϕ represent the polar and azimuthal angles, respectively. Each pure qubit state corresponds to a point on this sphere, with associated *Bloch vector* given by $(\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$.

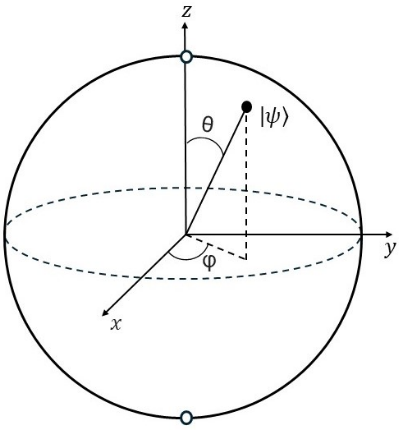


Figure 6: Bloch sphere representation of a qubit

The (trace) distance between two quantum states $|\psi\rangle$ and $|\psi'\rangle$, $\||\psi\rangle - |\psi'\rangle\|_1$, is their Euclidean distance in the Bloch sphere [71].

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 θ. 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 α and β, 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.

5.2.2 Multi-qubit States

Definition 5.2.4. 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 $|\phi\rangle$. As for any complex vector, $|\psi\rangle^{\otimes n}$ denotes the n-fold tensor product of state $|\psi\rangle$ with itself. The computational basis states of an n-qubit system are of the form $|x_1 \dots x_n\rangle$ 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 is known for multiple qubits.

Entanglement

Definition 5.2.5. An interesting aspect of multi-qubit states is the phenomenon of *entanglement*. This term indicates strong intrinsic correlations between two (or more) particles when the quantum state of each of them cannot be described independently of the state of the other (*i.e.*, it cannot be written as a product of states of the individual qubits). Measuring one qubit of the entangled pair affects the state of the other qubit. This must happen even if the particles are far apart.

In order to better understand this concept, consider the follow *Bell state* or *EPR pair*:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Upon measuring the first qubit, there are two possible outcomes: 0 with probability $1/2$ and 1 with probability $1/2$. Remarkably, if the first qubit is measured to be 0, the second qubit will also be 0 with probability 1; and if the first qubit is measured to be 1, the second qubit will also be 1 with probability 1. Therefore, the measurement outcomes are correlated.

These correlations prompted Einstein, Podolsky, and Rosen to publish a paper [79] questioning the completeness of quantum mechanics in 1935. The EPR paradox presented a dilemma: the existence of entanglement (i.e., correlations that persist regardless of distance) versus local realism and hidden variables. Einstein argued that if two objects, which have interacted in the past but are now separated, exhibit perfect correlation, they must possess a set of properties determined before their separation. These properties would persist in each object, dictating the outcomes of measurements on both sides. Einstein believed that the strong correlations predicted by quantum mechanics necessitate the existence of additional properties not accounted for by the quantum formalism that determine the measurement results. Therefore, he argued that quantum mechanics might require supplementation, as it may not represent a complete or ultimate description of reality.

In 1964, John Bell made a remarkable discovery: the measurement correlations in the Bell state are stronger than those that could ever occur between classical systems [80]. He explored the idea that each entangled particle might possess hidden properties — unaccounted for by quantum mechanics—that determine the measurement outcomes. Then, through mathematical reasoning, Bell demonstrated that the correlations predicted by any local hidden variable theory cannot exceed a specific level. There is an upper limit of correlations fixed by what today is called the “Bell inequalities”. He found that quantum theory sometimes predicts correlations that exceed this limit. Consequently, an experiment could settle the debate by testing whether or not correlations surpass the bounds he had found following Einstein’s position.

In 1982, Alain Aspect conducted an experiment that confirmed the violation of the Bell inequalities [81]. In this experiment, polarizers were placed more than twelve meters apart. This meant that the correlation obtained could not be explained by the fact that the particles carry within them unmeasured properties. Moreover, it proved that the outcome of the measurement is not determined until the moment of measurement. There seemed to be an instantaneous exchange between two particles at the time of measurement when they were twelve meters apart.

Sixteen years later, Nicolas Gisin [82] and Anton Zeilinger [83] conducted similar experiments, demonstrating that entanglement persists over distances of several kilometers. More recently, [84] extended these tests using entangled photon pairs sent from a satellite to verify Bell’s inequalities over a distance of one thousand kilometers, further confirming that, regardless of the distance, entangled particles behave as an indivisible, inseparable whole. The connection between them is so profound that it appears to challenge the principles of relativity. This phenomenon is known as *quantum nonlocality*.

5.2.3 Unitary operators

Pauli Matrices

Definition 5.2.6. The Pauli matrices are a set of three 2×2 hermitian matrices that are defined as follows:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The eigenvectors and eigenvalues of the Pauli matrices are as follows:

$$\sigma_x \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad \sigma_y \begin{pmatrix} 1 \\ i \end{pmatrix} = \begin{pmatrix} 1 \\ i \end{pmatrix}, \qquad \sigma_z \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\sigma_x \begin{pmatrix} 1 \\ -1 \end{pmatrix} = -\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad \sigma_y \begin{pmatrix} 1 \\ -i \end{pmatrix} = -\begin{pmatrix} 1 \\ -i \end{pmatrix}, \quad \sigma_z \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -\begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The normalized eigenvectors of σ_x are $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, and normalized eigenvectors of σ_y are $|+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle)$ and $|-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle)$. The eigenvectors of σ_z are $|0\rangle$ and $|1\rangle$. These eigenvectors correspond to the $\hat{x}, \hat{y}$ and $\hat{z}$ axes of the Bloch sphere in Figure 6, respectively.

When matrices σ_x, σ_y or σ_z are applied to a state on the Bloch sphere, they rotate the state by π radians around the $\hat{x}, \hat{y}$ or $\hat{z}$ axis, respectively. For example, the action of σ_x on the state $|0\rangle$ is to rotate it to $|1\rangle$, and vice versa. Note that for the eigenstates of these matrices with eigenvalue -1, this still applies if considering a global phase of $-1 = e^{i\pi}$, given that two quantum states $|\psi\rangle$ and $e^{i\phi}\,|\psi\rangle$ are indistinguishable by any quantum measurement.

Matrices σ_x and σ_z will also be referred to as X and Z, respectively.

Unitary operators

Definition 5.2.7. *Closed systems*, i.e., systems that do not interact with other systems evolve according to unitary operators. In quantum computation, these unitary operators are also known as *gates*. For a state $|\psi\rangle$, a *unitary operator* U describes an evolution from $|\psi\rangle$ to $U\,|\psi\rangle$.

Example 5.2.8. Pauli matrices are examples of unitary operators. The X and Z gates are often referred to as the *not* and *phase flip* gates, respectively. Other important unitary operators include *Hadamard gate*, denoted H, which maps $|0\rangle$ to $|+\rangle$ and $|1\rangle$ to $|-\rangle$, and the *phase-shift gate*, denoted P, which leaves $|0\rangle$ unaltered applies a phase shift of θ to the state $|1\rangle$:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad P = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}.$$

When the Pauli matrices are exponentiated, they result in three valuable classes of unitary matrices, corresponding to the rotation operators around the $\hat{x}$, $\hat{y}$, and $\hat{z}$ axes, which are defined as follows:

$$R_x(\theta) = e^{-i\theta\sigma_x/2} = \cos\left(\frac{\theta}{2}\right) \mathrm{id} - i\sin\left(\frac{\theta}{2}\right) \sigma_x = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix},$$

$$R_y(\theta) = e^{-i\theta\sigma_y/2} = \cos\left(\frac{\theta}{2}\right) \mathrm{id} - i\sin\left(\frac{\theta}{2}\right) \sigma_y = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix},$$

$$R_z(\theta) = e^{-i\theta\sigma_z/2} = \cos\left(\frac{\theta}{2}\right) \mathrm{id} - i\sin\left(\frac{\theta}{2}\right) \sigma_z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}.$$

Theorem 5.2.9. *[71] Suppose U is a unitary operation on a single qubit. Then there exist real numbers α, β, γ and δ such that*

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta).$$

Example 5.2.10. There are also multi-qubit gates, such as the *controlled-not* gate, denoted ***CNOT***, in which the state of the first qubit determines whether the X gate is applied to the second qubit. The first qubit is called the *control qubit*, and the second is the *target qubit*. The gate is defined by the following matrix:

$$\mathit{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

In this case, the states $|00\rangle$ and $|01\rangle$ remain unchanged, while $|10\rangle$ and $|11\rangle$ are mapped to each other.

There is an "extension" of the controlled-not gate, the controlled-U gate, where U is a unitary gate acting on a single qubit. This gate applies the gate U to the target qubit if the control qubit is in state $|1\rangle$ and does nothing otherwise. It is defined as:

$$\begin{aligned}
&\mathit{CU}(|0\rangle \otimes |\psi\rangle) = |0\rangle \otimes |\psi\rangle \\
&\mathit{CU}(|1\rangle \otimes |\psi\rangle) = |1\rangle \otimes U\,|\psi\rangle\,.
\end{aligned}$$

It should be noted that no completely closed systems exist in the universe. Nevertheless, for many systems, the approximation of a closed system is valid.

5.2.4 Measurements

There are times when it necessary to observe the system to extract information. This interection leaves the system no longer closed and, consequently, the evolution of the system is no longer unitary.

Definition 5.2.11. The act of measuring a qubit is represented by a set of operators called *measurement operators*, denoted $\{M_m\}$. These operators act on the state space of the system being measured. The index m refers possible measurement outcomes. These measurement operators must satisfy the completeness equation $\sum_m M_m^\dagger M_m = \mathrm{id}$, which ensures that the probabilities of all possible outcomes sum to 1. If a measurement M_m is performed on a state $|\psi\rangle$ the outcome m is observed with probability $p_m = \langle\psi|\, M_m^\dagger M_m\, |\psi\rangle$ for each m. Moreover, after a measurement yielding outcome m, the state collapses to

$$|\psi'\rangle = \frac{M_m\,|\psi\rangle}{\sqrt{p_m}}.$$

Definition 5.2.12. A measurement is called a *projective measurement* if its measurement operators are projectors.

Example 5.2.13. In the case of the computational basis, the measurement operators are the projectors onto the basis states $|0\rangle$ and $|1\rangle$ denoted by $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$, respectively. Considering an arbitrary state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, the probabilities of measuring 0 and 1 are $p_0 = \langle\psi|\, M_0 M_0^\dagger\, |\psi\rangle = \langle\psi|\, M_0\, |\psi\rangle = |\alpha|^2$, and $p_1 = \langle\psi|\, M_1 M_1^\dagger\, |\psi\rangle = \langle\psi|\, M_1\, |\psi\rangle = |\beta|^2$, respectively. Consequently the states after measurement are

$$\begin{aligned}
|\psi'\rangle &= \frac{M_0\,|\psi\rangle}{|\alpha|} = \frac{\alpha}{|\alpha|}\,|0\rangle = |0\rangle \text{ (with } p = p_0) \quad \text{and} \\
|\psi''\rangle &= \frac{M_1\,|\psi\rangle}{|\beta|} = \frac{\beta}{|\beta|}\,|1\rangle = |1\rangle \text{ (with } p = p_1)
\end{aligned}$$

From now on, unless stated otherwise, any reference to measurement should be understood as pertaining to the computational basis.

As previously mentioned, any states $|\psi\rangle$ and $e^{i\phi}|\psi\rangle$ are indistinguishable by any quantum measurement. Consider a measurement operator M_m, the probabilities of obtaining outcome m are $\langle\psi| M_m^\dagger M_m |\psi\rangle$ and $\langle\psi| e^{-i\theta} M_m^\dagger M_m e^{i\theta} |\psi\rangle = \langle\psi| M_m^\dagger M_m |\psi\rangle$. For this reason, it is said that these states are equal from an observational point of view.

5.2.5 Density operators

Until now the state vector formalism was used. However there is an alternative formulation using density operators. The density operator is often known as the *density matrix*, the two terms will be used interchangeably.

Definition 5.2.14. A quantum state $|\psi\rangle$ is said to be a *pure state* if it is completely known, *i.e.*if it can be written as a ket. In this case, the state can be written in the density operator formalism as $\rho = |\psi\rangle\langle\psi|$.

Definition 5.2.15. A state that is a probabilistic mixture of pure states is designated a *mixed state*. A mixed state can be represented by a density operator $\rho = \sum_i |p_i|^2 |\psi_i\rangle\langle\psi_i|$, where $|p_i|^2$ is the probability of the system being in state $|\psi_i\rangle$.

Definition 5.2.16 (Unitary Evolution of a Density Operator)**.** When a unitary operator U is applied to a mixed quantum state described by a density matrix ρ, the resulting state is given by $\rho' = U\rho U^\dagger$.

Definition 5.2.17 (Measurement of a Density Operator)**.** Given a collection of measurement operators $\{M_m\}$, the probability of obtaining outcome m when measuring a state ρ is $p_m = \text{Tr}(M_m \rho M_m^\dagger)$. After observing outcome m, the post-measurement state collapses to:

$$\rho' = \frac{M_m \rho M_m^\dagger}{\text{Tr}(M_m \rho M_m^\dagger)}.$$

Definition 5.2.18. In subsection 5.2.1 it was shown how to determine the cartesian coordenates of a pure state in the Bloch sphere from the state vector. For an arbitrary 2×2 density matrix, the following holds

$$\rho = \frac{1}{2}(\text{id} + r_x\sigma_x + r_y\sigma_y + r_z\sigma_z), \tag{5.4}$$

where $r = (r_x, r_y, r_z)$ is a real three-dimensional vector such that $\|r\|_2 \leq 1$. This vector is known as the *Bloch vector for the state* ρ. Since ρ is Hermitian, r_x, r_y and r_z are always real. To derive the inverse map $r_\mu = \text{Tr}(\rho\sigma_\mu)$, consider the properties of Pauli matrices:

$$\text{Tr}(\sigma_\mu) = 0, \quad \text{Tr}(\sigma_\mu\sigma_\nu) = 2\delta_{\mu\nu}.$$

Consequently,

$$\text{Tr}(\rho\sigma_\mu) = \frac{1}{2}\sum_\nu r_\nu \text{Tr}(\sigma_\nu\sigma_\mu) = \frac{1}{2} \cdot 2r_\mu = r_\mu.$$

Thus, the inverse map of Equation 5.4 is

$$r_\mu = \text{Tr}(\rho\sigma_\mu). \tag{5.5}$$

Note that given that the trace is linear and matrix multiplication distributes over matrix addition, the cartesian coordinates of an operator consisting of the sum or subtraction of density operators can also be determined by Equation 5.5.

Reduced density operator

Density operators are particularly well-suited for describing individual subsystems of a composite quantum system. This type of description is provided by the *reduced density operator*

Definition 5.2.19. Consider Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ of systems A and B, respectively. The *partial trace over* B,$\text{Tr}_B \colon \mathcal{T}(\mathcal{H}_A \otimes \mathcal{H}_B) \to \mathcal{T}(\mathcal{H}_A)$, is defined by

$$\text{Tr}_B = \text{id}_{\mathcal{T}(\mathcal{H}_A)} \otimes \text{Tr}.$$

Similarly, the *partial trace over* B corresponds to the map $\text{Tr}_A \colon \mathcal{T}(\mathcal{H}_A \otimes \mathcal{H}_B) \to \mathcal{T}(\mathcal{H}_B)$, defined by

$$\text{Tr}_B = \text{Tr} \otimes \text{id}_{\mathcal{T}(\mathcal{H}_B)}.$$

Definition 5.2.20. Given physical systems A and B whose composite system is given by the density operator ρ_{AB}, the *reduced density operator* for subsystem A is $\rho_A = \text{Tr}_B(\rho_{AB})$. Similarly, the *reduced density operator for subsystem* B is $\rho_B = \text{Tr}_A(\rho_{AB})$.

Recall that in this section we restrict ourselves to the finite-dimensional setting, where the set of trace class operators on $\mathcal{H}$, $\mathcal{T}(\mathcal{H})$, can be identified with the set of bounded operators on $\mathcal{H}$, $\mathcal{B}(\mathcal{H})$. Nevertheless, we use the notation $\mathcal{T}(\mathcal{H})$, as it reflects the natural setting of the density operator formalism — a density operator must be trace-class, even in the infinite-dimensional case [72].

5.2.6 Quantum Channels

Thus far, only two types of quantum operations have been discussed: unitary operators, which describe the evolution of a closed quantum system, and measurements, which describe the act of observing a quantum system. Now, a new type of quantum operation that accounts for the more realistic notion of interaction between a quantum system and an environment will be introduced. Nonetheless, it is necessary to first introduce a few key concepts.

Definition 5.2.21. Operators that map operators to other operators are known as *super-operators*.

Definition 5.2.22. A super-operator $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ is called *positive* if it sends positive operators to positive operators, *i.e.* $A \geq 0 \Rightarrow QA \geq 0$.

Definition 5.2.23. The tensor product of two super-operators $\Phi : \mathcal{T}(\mathcal{H}_1) \to \mathcal{T}(\mathcal{K}_1)$ and $\Psi : \mathcal{T}(\mathcal{H}_2) \to \mathcal{T}(\mathcal{K}_2)$ is an operator $\Phi \otimes \Psi : \mathcal{T}(\mathcal{H}_1 \otimes \mathcal{K}_1) \to \mathcal{T}(\mathcal{H}_2 \otimes \mathcal{K}_2)$ defined by the equation:

$$(\Phi \otimes \Psi)(A \otimes B) = \Phi(A) \otimes \Psi(B).$$

Definition 5.2.24. A linear mapping $\Phi : \mathcal{T}(\mathcal{H}_1) \to \mathcal{T}(\mathcal{H}_2)$ is *completely positive* if the mapping $\Phi \otimes \mathrm{id}_{\mathcal{T}(\mathcal{K})}$ on $\mathcal{T}(\mathcal{H}_1 \otimes \mathcal{K}) \to \mathcal{T}(\mathcal{H}_2 \otimes \mathcal{K})$ is positive for any Hilbert space $\mathcal{K}$.

Definition 5.2.25. A super-operator Φ is called *trace-preserving* (resp. *trace-nonincreasing*) if Tr $(\Phi A) =$ Tr(A) (resp. (Tr $(\Phi A) \leq$ Tr(A))).

Since density matrices are positive, any physically allowed transformation must be represented by a positive operator. Nonetheless, this is not sufficient on its own: since one can always extend the space $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n} \otimes \mathbb{C}^{m\times m}$ by adjoining a new quantum system, any physically allowed transformation must be completely positive. Finally, since the trace of a density matrix is always 1, any physically allowed transformation must be trace-preserving. A **Completely Positive Trace-Preserving** (**CPTP**) operator is traditionally called a *quantum channel*.

It is sometimes convenient to relax the trace-preserving condition to a trace-non-increasing condition, resulting in what is known as a *quantum operation*. This accounts for phenomena such as *qubit leakage* (the unintended loss of quantum information from computational basis states, $|0\rangle$ and $|1\rangle$, into higher-energy non-computational states, e.g., $|2\rangle$, $|3\rangle$, breaking the idealized two-level qubit assumption) and operations such as *postselection* (a technique in quantum algorithms where operations are conditioned on measurement outcomes, often leading to non-trace-preserving maps) [85].

Kraus operator sum representation

Assume that there is a quantum system S of interest which is a subsystem of a larger system which also includes an environment E. These systems have a joint unitary evolution described by a unitary operator U acting on the composite system, $U(\rho_{SE}) = U\rho_{SE}U^\dagger$. Given that density matrices are positve operators, and therefore Hermitian with non-negative eigenvalues, the density operator of the environment ρ_E initially can be written as

$$\rho_E = \sum_i p_i \left|i\right\rangle \left\langle i\right|$$

where $\left|i\right\rangle$ form an orthonormal basis for the state space of E and p_i are positive.
The state of the subsystem S after the unitary evolution corresponds to the partial trace of the joint state over the environment,

$$\begin{aligned}\rho'_S =& \mathrm{Tr}_E(U\rho_{SE}U^\dagger)\\ =& \sum_\mu \left\langle\mu\right| U\rho_{SE}U^\dagger \left|\mu\right\rangle\end{aligned}$$

where $\{\left|\mu\right\rangle\}$ span the state space of E.
Considering that initially both systems are completely decoupled, the initial state of the system can be written as $\rho_{SE} = \rho_S \otimes \rho_E$. Thus,

$$\begin{aligned}\rho'_S &= \sum_\mu \left\langle\mu\right| U\rho_S \otimes \sum_i p_i \left|i\right\rangle \left\langle i\right| U^\dagger \left|\mu\right\rangle\\ &= \sum_{\mu i} \sqrt{p_i} \left\langle\mu\right| U \left|i\right\rangle \rho_S \sqrt{p_i} \left\langle i\right| U^\dagger \left|\mu\right\rangle\\ &= \sum_{\mu i} \mathsf{K}_{\mu i}\rho_S \mathsf{K}^\dagger_{\mu i}\end{aligned}$$

where the set of operators $\{\mathsf{K}_{\mu i}\}$ is designated *Kraus operators* and $\mathsf{K}_{\mu i} = \sqrt{p_i}\left\langle\mu\right| U \left|i\right\rangle$. Note that $\{\left|\mu\right\rangle\}$ and $\{\left|i\right\rangle\}$, act only in the state space of E.

Definition 5.2.26. The equation $\rho'_S = \sum_{\mu i} \mathsf{K}_{\mu i}\rho_S\mathsf{K}^\dagger_{\mu i}$ is called an ***Operator Sum Representation (OSR)***. An OSR can be thought of as a quantum channel that maps ρ_S to $\sum_{\mu i} \mathsf{K}_{\mu i}\rho_S\mathsf{K}^\dagger_{\mu i}$, given this map is CPTP [70, 86].

Non-selective measurements

In the previously presented formalism to represent all the possible outcomes of a measurement, described by a set of operators $\{M_m\}$, on a state ρ, it would be necessary to write that state ρ collapse to state $\rho_m = \frac{M_m^\dagger \rho M_m}{\text{Tr}(M_m \rho M_m^\dagger)}$ with probability $p_m = \text{Tr}(M_m \rho M_m^\dagger)$, for each possible outcome m. Although the selective description above is useful conceptually, it is often impractical for calculations. Instead, one uses *non-selective measurements*, in which the possible outcomes are not explicitly stated.

Definition 5.2.27. A *non-selective measurement* is a quantum measurement in which the post-measurement state of the system is then given by the weighted sum over all possible outcomes:

$$\rho = \sum_m p(m)\rho_m = \sum_m M_m \rho M_m^\dagger.$$

This last equality corresponds to an Kraus operator sum representation, where the set of Kraus operators is $\{M_m\}$.

5.2.7 Norms on quantum channels

Definition 5.2.28. The trace norm of a super-operator $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ is defined as:

$$\|\Phi\|_1 = \sup\{\|\Phi A\|_1 \mid \|A\|_1 = 1\},$$

where $A \in \mathcal{T}(\mathcal{H})$.

Unfortunately, this norm is not stable under tensoring , given that the inequation $\|\Phi \otimes I_{\mathcal{T}(\mathcal{H})}\|_1 \geq \|\Phi\|_1$ does not hold [70]. As a result, the diamond norm, which is based on the trace norm, is used instead in the context of quantum channels.

Definition 5.2.29. Given a super-operator $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$, the diamond norm, $\|\cdot\|_\diamond$, is defined as:

$$\|\Phi\|_\diamond = \|\Phi \otimes \text{id}_{\mathcal{T}(\mathcal{H})}\|_1$$

Since the diamond norm is generally difficult to compute, we will rely on the following properties:

Theorem 5.2.30. *Let $\Phi \in \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ be a positive map. Then it holds that*

$$\|\Phi\|_1 = \sup\{\text{Tr}\left(\Phi(vv^*)\right) \mid \|v\|_2 = 1, v \in \mathcal{H}\}.$$

Proposition 5.2.31. *[70, Proposition 3.48] For all maps $\Phi \in \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ and $\Psi \in \mathcal{T}(\mathcal{K}) \to \mathcal{T}(\mathcal{L})$, it holds that*

$$\|\Psi\Phi\|_\diamond \leq \|\Psi\|_\diamond \|\Phi\|_\diamond .$$

Theorem 5.2.32. *[70, Theorem 3.49] Let $\Phi \in \mathcal{T}(\mathcal{H}_1) \to \mathcal{T}(\mathcal{K}_1)$ and $\Psi \in \mathcal{T}(\mathcal{H}_2) \to \mathcal{T}(\mathcal{K}_2)$ be maps. Then it holds that*

$$\|\Phi \otimes \Psi\|_\diamond = \|\Phi\|_\diamond \|\Phi\|_\diamond$$

Theorem 5.2.33. *[70, Theorem 3.55] Let $n \leq m$, let $V_0, V_1 : \mathcal{T}(\mathcal{H}, \mathcal{K})$ be isometries, and define CPTP operators $\Phi_0, \Phi_1 : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ as*

$$\Phi_0(\rho) = V_0 \rho V_0^\dagger \quad \text{and} \quad \Phi_1(\rho) = V_1 \rho V_1^\dagger$$

for all $\rho \in \mathcal{T}(\mathcal{H})$. There exists a unit vector $u \in \mathcal{H}$ such that

$$\left\|\Phi_0(uu^\dagger) - \Phi_1(uu^\dagger)\right\|_1 = \|\Phi_0 - \Phi_1\|_\diamond .$$

Theorem 5.2.34. *[70, Theorem 3.56] Let $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ be a quantum channel, let $\varepsilon \in [0, 2]$, and suppose that*

$$\|\phi(\rho) - \rho\|_1 \leq \varepsilon$$

for every density operator $\rho \in \mathcal{T}(\mathcal{H})$. It holds that

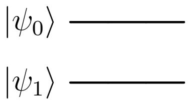
$$\|\phi - \mathrm{id}_{\mathcal{T}(\mathcal{H})}\|_1 \leq \sqrt{2\varepsilon}.$$

5.2.8 Quantum circuits

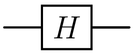
As quantum computation remains in its early stages of development, programming is primarily based on the use of *quantum circuits*.

Definition 5.2.35. A *quantum circuit* consists of wires and quantum gates, which serve to transmit and manipulate quantum information. Each wire corresponds to a qubit, while the gates represent operations that can be applied to these qubits.

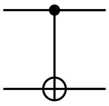
In this subsection the notation for the quantum gates used in this work will be introduced. Wires in parallel represent the tensor product of the respective qubits. For instance, $\psi_0 \otimes \psi_1$ corresponds to



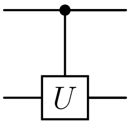
The single bit gates presented in Section 5.2.3 are represented as a box with the symboL of the gate inside. For example, the Hadamard gate is represented as



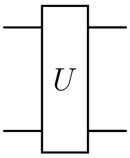
The controlled-not gate, which is a two-qubit gate, is represented as



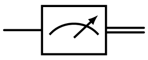
Similarly, the controlled-U gate, where U is an unitary single-qubit gate, is represented as



An arbitrary unitary operator acting on n qubits is represented as a box acting on n wires. For instance, the operator U acting on two qubits is represented as



CPTP maps are depicted as boxes containing the corresponding map symbols.
The measurement operation is representes by a “meter” symbol. Given that output of a measurement is a classical bit, the wire representing the output of a measurement is a classical wire, represented by a double line.



5.2.9 No-cloning theorem

The no-cloning theorem states that it is impossible to duplicate an unknown quantum bit [87]. In this subsection, an elementary proof of this theorem will be presented.

Suppose that there exists a cloning machine, C, that produces a clone (a duplicate) of any unknown state. It recieves a qubit $|\psi\rangle$ and some standard pure state $|s\rangle$ as input and returns the state $|\psi\rangle \otimes |\psi\rangle$.

takes one state as input and returns two of the same kind. The second is a duplicate of the first in the sense that no experiment could distinguish between them. Hence, the action of a clonning machine can be written as

$$|\psi\rangle \otimes |s\rangle \mapsto |\psi\rangle \otimes |\psi\rangle$$

for all states $|\psi\rangle$.

However, due to its violation of linearity, this kind of transformation is not a valid quantum operation. Namely, let $|\psi\rangle = \sum_i \alpha_i |\psi_i\rangle$ be a mixed state. Then:

$$C\left(\sum_i \alpha_i|\psi_i\rangle\right) \otimes |s\rangle = \left(\sum_i \alpha_i|\psi_i\rangle\right) \otimes \left(\sum_i \alpha_i|\psi_i\rangle\right) = \sum_{i,j} \alpha_i\alpha_j|\psi_i\rangle \otimes |\psi_j\rangle,$$

but assuming linearity of the cloning transformation, we would get:

$$C\left(\sum_i \alpha_i|\psi_i\rangle\right) \otimes |s\rangle = \sum_i \alpha_i C\left(|\psi_i\rangle\right) = \sum_i \alpha_i\left(|\psi_i\rangle \otimes |\psi_i\rangle\right).$$

These two expressions generally do not coincide. For instance, let $\{|\psi_i\rangle\}$ be a set of orthogonal pure states. In this case, the coefficients $\alpha_i\alpha_j$ and α_i correspond exactly to the eigenvalues of the final states in the equations above. Since $\alpha_i\alpha_j < \alpha_i$ for all $\alpha_j < 1$, the two resulting states are distinct.

It should be noted that 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).

5.3 Functional Analysis

In this section, we are no longer restricted to finite-dimensional vector spaces; the term "vector space" now also encompasses infinite-dimensional ones. Due to this, we will introduce topological concepts and results necessary for the next two sections.

Topology is the abstract mathematical study of concepts like convergence and approximation, among other things, generalizing familiar notions from calculus and analysis. Note that, for instance, in a metric space, a sequence $\{x_n\}$ with $n \in \mathbb{N}$ converges to a point x if the distance $d(x_n, x)$ tends to zero; that is, for every $\varepsilon > 0$, there exists n_0 such that $d(x_n, x) < \varepsilon$ for all $n \geq n_0$. However, metric spaces are not sufficient to describe all types of convergence. An example is the pointwise convergence of all real-valued functions on the interval $[0, 1]$. In fact, there is no metric on the space of all real functions on the interval $[0, 1]$ for which one can define a distance function $d(f_n, f)$ such that $d(f_n, f) \to 0$ if and only if $f_n(x) \to f(x)$ for every $x \in [0, 1]$. A foundational idea in topology is that of a *neighborhood*—a collection of points considered "sufficiently close" to a given point. From this arises the concept of *open sets*, which are sets that serve as neighborhoods for all their points. The collection of all such open sets defines a *topology*, and a set equipped with a topology becomes a *topological space*. This framework introduces some subtleties: for example, traditional sequences are often inadequate for capturing convergence, requiring the more general notion of *nets*, which are indexed over broader structures than the natural numbers.

Definition 5.3.1. A *topology* τ on a set S is a collection of subsets of S satisfying the following properties:

1. $\varnothing \in \tau$ and $S \in \tau$.

2. τ is closed under finite intersections: if $U_1, U_2, \ldots, U_n \in \tau$, then $\bigcap_{i=1}^{n} U_i \in \tau$.

3. τ is closed under arbitrary unions: if $\{U_\alpha\}_{\alpha \in A} \subseteq \tau$, then $\bigcup_{\alpha \in A} U_\alpha \in \tau$.

A nonempty set S equipped with a topology τ is called a *topological space*, and is denoted by (S, τ) (or simply S when no ambiguity arises). A member of τ is called an *open set* in S. The complement of an open set is a *closed set*.

A set S can have many different topologies. The family of all topologies on S is partially ordered by set inclusion. If $\tau_1 \subset \tau_2$, that is, if every τ_1-open set is also τ_2-open, then we say that τ_1 is *weaker* or *coarser* than τ_2, and that τ_2 is *stronger* or *finer* than τ_1.

Example 5.3.2. Standard examples of topologies are presented below:

1. *Trivial (or indiscrete) topology:* On a set S, the trivial topology consists only of the sets $\varnothing$ and S. These are also the only closed sets.

2. *Discrete topology:* The discrete topology on a set S consists of all possible subsets of S. In this topology, every set is both open and closed.

3. *Standard topology on $\mathbb{R}$:* The metric $d(v, w) = |v - w|$ on $\mathbb{R}$ induces a topology where open sets are unions of open intervals. This is known as the standard topology on $\mathbb{R}$.

Definition 5.3.3. A *neighborhood* of a point $s \in S$ in a topological space (S, τ) is any subset $N \subseteq S$ that contains s in its interior. In this case, s is called an *interior point* of N.

Definition 5.3.4. The *norm topology* induced by a norm $\| \cdot \|$ is the topology generated by the metric $d(v, w) = \|v - w\|$.

Topology is about convergence and also about continuity. Consider a map $f \colon V \to W$, the idea behind continuity is that if we move $v \in V$ only slightly, then $f(v)$ should change by a small amount as well. The less we move v, the less $f(v)$ should change. We begin, with a more intuitive definition restricted to the setting of metric spaces.

Definition 5.3.5. Let (V, d_V) and (W, d_W) be metric spaces. A mapping $f \colon V \to W$ is *sequentially continuous* if for every convergent sequence $(x_n)_{n \in \mathbb{N}}$ in V with $v_n \to v$, the image sequence $(f(v_n))_{n \in \mathbb{N}}$ converges to $f(v)$ in W. That is,

$$v_n \to n \text{ in } V \;\Rightarrow\; f(v_n) \to f(v) \text{ in } W.$$

More generally, continuity may be defined as follows:

Definition 5.3.6. Let (S_1, τ_1) and (S_2, τ_2) be topological spaces. A map $f \colon S_1 \to S_2$ is *continuous* if and only if for every open subset $N \subseteq S_2$, the preimage $f^{-1}(N)$ is open in S_1.

Definition 5.3.7. A net in a set S is a function $s \colon D \to S$, where D is a directed set. The directed set D is called the *index set* of the net and the members of D are *indexes*.

Definition 5.3.8. Let S_1 and S_2 be two topological spaces, and let s_1 be a point in S_1. A map $f : S_1 \to S_2$ is said to be *continuous at* s_1 if and only if, for every open neighborhood S_1 of $f(s_1)$, there exists an open neighborhood N of s_1 such that $\{f(n) \,|\, n \in N\} \subseteq N$.

The proposition and theorem below present continuity in a more intuitive way:

Proposition 5.3.9. *[88, Theorem 2.27] Let S_1 and S_2 be two topological spaces. A map $f : S_1 \to S_2$ is continuous if and only if it is continuous at every point $s_1 \in S_2$.*

Theorem 5.3.10. *[88, Theorem 2.28] Let $f \colon S_1 \to S_2$ be a function between topological spaces, and let $s_1 \in S_2$. The following statements are equivalent:*

1. *The function f is continuous at s.*

2. *For every net (s_α) in S_1 converging to s, the net $(f(s_\alpha))$ converges to $f(s)$ in S_2.*

Definition 5.3.11. A *topological vector space* is a vector space V equipped with a linear topology τ such that:

1. every singleton $\{v\} \subset V$ is a closed set, and

2. the vector space operations (addition and scalar multiplication) are continuous with respect to τ. That is, the addition map $(x, y) \mapsto x + y$, from the Cartesian product $V \times V$ into V, is continuous, and the scalar multiplication map $(r, x) \mapsto rx$, from $\mathcal{F} \times V$ into V, is also continuous.

Definition 5.3.12. Let V be a vector space. Linear maps from V to its scalar field are called *linear functionals*. The set of all continuous linear functionals on V forms a vector space, called the *(topological) dual space* of V, and is denoted by V^*. It is common to designate elements of the dual space V^* by v^*.

Theorem 5.3.13. *[42, Theorem 4.3] Let V be a normed vector space. For each $v^* \in V^*$, define its norm by*

$$\|v^*\| := \sup\left\{|v^*(v)| : \|v\| = 1\right\}.$$

This defines a norm on V^* under which V^* is a Banach space. Moreover, for every $v \in V$, we have

$$\|v\| = \sup\left\{|v^*(v)| : \|v^*\| = 1\right\}.$$

As a consequence, the map $v^* \mapsto v^*(v)$ defines a bounded linear functional on V^*, and its norm equals $\|v\|$.

Definition 5.3.14. Let V be a vector space. The *weak*-topology* on the dual space V^* is the coarsest topology that makes all evaluation maps

$$v^* \mapsto v^*(v)$$

continuous for every $v \in V$.

The following concepts and results will be needed later in Section 5.5.2.

Proposition 5.3.15. *[89, Proposition 2.3.10] Let V and W be bounded vector spaces and let $f : V \to W$ be a bounded linear map. Then its dual $f^* : V^* \to W^*$, defined by*

$$f^*(\varphi) = \varphi \circ f,$$

is also a bounded linear map.

Theorem 5.3.16. *[90, Theorem 9.15] Let $\mathcal{H}$ be a finite-dimensional Hilbert space, and let φ be a linear functional on $\mathcal{H}^*$. Then there exists a unique vector $v_\varphi \in \mathcal{H}$ such that*

$$\varphi(w) = \langle w, v_\varphi \rangle \quad \textit{for all } w \in \mathcal{H}.$$

We call v_φ the Riesz vector *for φ and denote it by v_φ.*

Using the Riesz representation theorem, we can define a map $\phi^* : V^* \to V$ by setting $\phi^*(\varphi) = v_\varphi$, where v_φ is the Riesz vector corresponding to $\varphi \in V^*$. Since the Riesz representation is unique, ϕ^* is well-defined [90].

Proposition 5.3.17. *A $\phi^* : \mathcal{B}(\mathcal{H})^* \to \mathcal{B}(\mathcal{K})$, is defined as $\phi^*(\varphi) = \sum_{i,j} \varphi(|j\rangle\langle i|) \cdot |i\rangle\langle j|$.*

Alternativa 1

Proof. □

Consider a linear map $\phi : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$. Since it is well known that we can identify $\mathcal{B}(\mathcal{H})$ with the matrix algebra $\mathcal{M}_n$, where $n = \dim(\mathcal{H})$, the map ϕ can be equivalently represented as $\phi : \mathcal{M}_n \to \mathcal{M}_m$, where $m = \dim(\mathcal{K})$.

Consequently, by the Riesz representation theorem, and the definitions of trace and inner product for matrices we have:

$$\varphi(|j\rangle\langle i|) = \langle |j\rangle\langle i|, v_\varphi \rangle = \mathrm{Tr}(|i\rangle\langle j| \cdot v_\varphi) = (v_\varphi)_{i,j},$$

where $(v_\varphi)_{i,j}$ denotes the entry in the i-th row and j-th column of v_φ.

As a result, we obtain

$$\phi^*(\varphi) = \sum_{i,j} \varphi(|j\rangle\langle i|) \cdot |i\rangle\langle j| = \sum_{i,j} (v_\varphi)_{i,j} \cdot |i\rangle\langle j| = v_\varphi.$$

Alternativa 2: Mudar a assinatura de ϕ^* para as matrizes e assim temos

By the Riesz representation theorem, and the definitions of trace and inner product for matrices we have:

$$\varphi(|j\rangle\langle i|) = \langle |j\rangle\langle i|, v_\varphi \rangle = \mathrm{Tr}(|i\rangle\langle j| \cdot v_\varphi) = (v_\varphi)_{i,j},$$

where $(v_\varphi)_{i,j}$ denotes the entry in the i-th row and j-th column of v_φ.

As a result, we obtain

$$\phi^*(\varphi) = \sum_{i,j} \varphi(|j\rangle\langle i|) \cdot |i\rangle\langle j| = \sum_{i,j} (v_\varphi)_{i,j} \cdot |i\rangle\langle j| = v_\varphi.$$

Definition 5.3.18. A function $f : (V, d) \to (W, d')$ between two metric spaces is said to be *uniformly continuous* if for every $\varepsilon > 0$ there exists some $\delta > 0$ (depending only on ε) such that

$$d(x, y) < \delta \quad \Rightarrow \quad d'(f(x), f(y)) < \varepsilon$$

for all $x, y \in X$. Any uniformly continuous function is continuous. An important property of uniformly continuous functions is that they map Cauchy sequences into Cauchy sequences [88].

Em principio estas 2 coisas vão sair

Definition 5.3.19. A function $f : (V, d_V) \to (W, d_W)$ between metric spaces is said to be *Lipschitz continuous* if there exists a real number $c \geq 0$ such that for every $v_1, v_2 \in V$,

$$d_W(f(v_1), f(v_2)) \leq c \cdot d_V(v_1, v_2).$$

The number c is called a *Lipschitz constant* for f. Clearly, every Lipschitz continuous function is uniformly continuous [88].

Lemma 5.3.20. *[47, Lemma 7.3.19] Let A be a dense subset of a topological space V, and let $f : A \to W$ be a continuous map into a metric space W. Then f has at most one continuous extension $g : V \to W$,* i.e., *one such that f and g coincide on A. This continuous extension g exists if (V, d_V) is a metric space, (W, d_W) is a complete metric space, and f is uniformly continuous from (A, d_V) to (W, d_W).*

5.4 W^*-Algebras

While quantum theory is traditionally formulated in terms of Hilbert spaces, there is also a more abstract and general formulation using operator algebras. This perspective traces back to Heisenberg's work on the spectral lines of the hydrogen atom in 1925, where he realized that observable quantities in quantum systems, such as the position of an electron in a hydrogen atom, are better represented by infinite arrays of complex numbers [91]. Born and Jordan subsequently recognized that these arrays should follow the rules of matrix multiplication [92]. To address the mathematical challenges posed by infinite matrices, Von Neumann formalized these ideas using operators on Hilbert spaces, more concretely, *Von Neumann* algebras [93]. This gave rise to the study of operator algebras, which are now applied in various domains in quantum theory, including quantum statistical mechanics [94], quantum field theory [95, 96], and quantum information theory [97]. We refer to the abstract characterization of von Neumann algebras as W^*-algebras, although the terms are often used interchangeably in the literature.

While C^*-algebras can also model quantum computing, W^*-algebras may be more suitable for this purpose. For instance, whereas C^*-algebras correspond to noncommutative geometry [98], W^*-algebras can be viewed as noncommutative analogues of measure theory or probability, aligning with the probabilistic nature of quantum [99, 100]. Moreover, there is previous work in this setting: in [22] it is shown that Selinger's category Q corresponds (up to categorical equivalence) to the finite-dimensional subcategory of $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$. Since it is impossible to introduce W^*-algebras without first covering C^*-algebras, this section presents the key concepts and results of both, laying the groundwork for Section 5.5.2.

5.4.1 C^*-Algebras

Uppercase letters in math script, $\mathscr{A}, \mathscr{B}, \mathscr{C}, \ldots$, will typically denote C^*-algebras.

Definition 5.4.1. A C^**-algebra* is a complex vector space $\mathscr{A}$ endowed with:

1. a binary operation, called *multiplication* (and denoted as such), which is associative and linear in both coordinates;

2. an element 1, called the *unit*, such that $1 \cdot a = a = a \cdot 1$ for all $a \in \mathscr{A}$;

3. a unary operation $(\cdot)^*$, called *involution*, such that for all $a, b \in \mathscr{A}$ and $\lambda \in \mathbb{C}$,

$$(a^*)^* = a, \quad (ab)^* = b^*a^*, \quad (\lambda a)^* = \overline{\lambda}a^*, \quad \text{and} \quad (a+b)^* = a^* + b^*;$$

4. a complete norm $\|\cdot\|$ such that $\|ab\| \leq \|a\| \cdot \|b\|$ for all $a, b \in \mathscr{A}$, and

$$\|a^*a\| = \|a\|^2.$$

   This last equality is called the C^**-identity*.

Note that while we have previously used $(\cdot)^*$ to denote elements of dual spaces, in the definition above and in the definition of involutive maps, this symbol will denote the involution operation.

Remark 5.4.2. In the literature, a C^*-algebra is typically not required to have a unit. When it does, it is called a *unital* C^**-algebra*.

Maps between C^*-Algebras

We consider only linear maps, hence the term “map” will always mean a linear map.

Definition 5.4.3. Let $\mathscr{A}$ be a C^*-algebra. An element x of $\mathscr{A}$ is positive if there exists an $y \in \mathscr{A}$ such that $x = y^*y$. We denote the set of positive elements of $\mathscr{A}$ by $\mathscr{A}_+$.

Definition 5.4.4. A linear map $\Phi : \mathscr{A} \to \mathscr{B}$ between C^*-algebras is called

1. *multiplicative* if $\Phi(ab) = \Phi(a)\Phi(b)$ for all $a, b \in \mathscr{A}$;
2. *involution preserving* if $\Phi(a^*) = \Phi(a)^*$ for all $a \in \mathscr{A}$;
3. *unital* if $\Phi(1) = 1$;
4. *subunital* if $1 - \Phi(1)$ is positive;
5. *positive* if $\Phi(a)$ is positive for every positive $a \in \mathscr{A}$.

A multiplicative involutive linear map is called a $*$-*homomorphism* and a unital $*$-*homomorphism* is also known as a **miu**-map. A bijective $*$-homomorphism is called a $*$-isomorphism.

Proposition 5.4.5. *[101, Theorem 1.5.7] Every $*$-homomorphism $\Phi : \mathscr{A} \to \mathscr{B}$ between C^*-algebras is* short*. Moreover, Φ is* isometric *if and only if it is injective.*

Proposition 5.4.6. *[22, Proposition 2.4] Let $\Phi : \mathscr{A} \to \mathscr{B}$ be a positive map between C^*-algebras. Then Φ is subunital if and only if it is short.*

Representations of C^*-Algebras

Definition 5.4.7. A *representation* of a C^*-algebra $\mathscr{A}$ is a pair $(\mathcal{H}, \pi)$, where $\mathcal{H}$ is a Hilbert space and $\pi : \mathscr{A} \to \mathcal{B}(\mathcal{H})$ is a miu-map. The representation is said to be *faithful* if π is injective.

Theorem 5.4.8. *[76, Theorem 9.18.] Every C^*-algebra admits a faithful representation.*

Matrices over C^*-Algebras

Definition 5.4.9. Let $\mathscr{A}$ be a C^*-algebra. For $n \in \mathbb{N}$, let $\mathcal{M}_n(\mathscr{A})$ denote the set of $n \times n$ matrices with entries in $\mathscr{A}$. Then $\mathcal{M}_n(\mathscr{A})$ is equipped with the following operations:

- *Addition and scalar multiplication* are defined pointwise:

$$(a_{ij}) + (b_{ij}) := (a_{ij} + b_{ij}), \qquad \alpha(a_{ij}) := (\alpha a_{ij})$$

- *Multiplication* is matrix multiplication:

$$(a_{ij})(b_{ij}) := \left[\sum_k a_{ik} b_{kj}\right]$$

- *Involution* is given by the conjugate transpose:

$$(a_{ij})^* := (a^*_{ji}).$$

Proposition 5.4.10. *[102, p.16-17] Let $\mathscr{A}$ be a C^*-algebra. Then $\mathcal{M}_n(\mathscr{A})$ is a C^*-algebra, too.*

In the context of the proposition above, the norm is determined via the identification $\mathcal{M}_n(\mathcal{B}(\mathcal{H})) = \mathcal{B}(\mathcal{H}^{\oplus n})$ [102, Exercises 1.1 and 1.2]. That is, considering a faithful representation $(\mathcal{H}, \pi)$ of $\mathcal{M}_n(\mathscr{A})$, and given $(A_{ij}) \in \mathcal{M}_n(\mathscr{A})$, we have

$$\|(A_{ij})\| = \|\pi(A_{ij})\| = \|\pi(A_{ij})\|_{\mathsf{op}} = \sup\{\|\pi(A_{ij})(v)\| \mid \|v\|_2 = 1, v \in \mathcal{B}(\mathcal{H}^{\oplus n})\}.$$

Note that, by Proposition 5.4.5, π is an isometry.

These matrices are important because they are used to define complete positivity in this setting.

Falar cb?

Definition 5.4.11. Let $\Phi : \mathscr{A} \to \mathscr{B}$ be a linear map between C^*-algebras. For each $n \in \mathbb{N}$, Φ induces a linear map

$$\mathcal{M}_n(\Phi) : \mathcal{M}_n(\mathscr{A}) \to \mathcal{M}_n(\mathscr{B}), \quad \mathcal{M}_n(\Phi)\big[x_{ij}\big] := \big[f(x_{ij})\big].$$

The map Φ is said to be *n-positive* if $\mathcal{M}_n(\Phi)$ is positive, and *completely positive* if $\mathcal{M}_n(\Phi)$ is positive for all $n \in \mathbb{N}$.

We have previously a definition of complete positivity in the setting of bounded/trace-class operators over finite dimensional Hilbert spaces (Definition 5.2.24). Given $\mathcal{B}(\mathcal{H})$ is an example of a C^*-algebra, we will check that the definitions are equivalent.

Proposition 5.4.12. *The definitions of completely positivity presented (Definition 5.2.24 and Definition 5.4.11) are equivalent for maps*

$$\Phi : \mathcal{B}(\mathcal{H} \otimes \mathcal{K}_1) \to \mathcal{B}(\mathcal{H} \otimes \mathcal{K}_2),$$

where $\mathcal{H}, \mathcal{K}_1, \mathcal{K}_2$ are finite-dimensional.

Proof. We begin by observing that the composition of positive maps is positive, and that the map which swaps tensor factors is itself positive [70]. Therefore, Definition 5.2.24 is equivalent to the one obtained by replacing $\Phi \otimes \mathrm{id}_{\mathcal{B}(\mathcal{H})}$ with $\mathrm{id}_{\mathcal{B}(\mathcal{H})} \otimes \Phi$.

We now proceed to demonstrate the equivalence between this definition and Definition 5.4.11. Here we note the isometric isomorphisms $\mathcal{M}_n \otimes \mathcal{M}_m \cong \mathcal{M}_{nm}$, $\mathcal{B}(\mathcal{H}) \cong \mathcal{M}_n$ [70] with $\dim(\mathcal{H}) = n$ and $\mathcal{M}_n(\mathcal{B}(\mathcal{H})) \cong \mathcal{M}_n \otimes \mathcal{B}(\mathcal{H})$ [103, Corollary 8.1.3]. It is straightforward that the first two isomorphisms are positive. Now, regarding the third isomorphism, consider that $[a_{i,j}] \in \mathcal{M}_n(\mathcal{B}(\mathcal{H}))$ is positive, *i.e.* $[a_{i,j}] = [b_{i,j}]^*[b_{i,j}]$. Note that

$$[b_{i,j}]^*[b_{i,j}] = [b_{i,j}^*][b_{i,j}] = \left[\left(\sum_k b_{k,i}^* b_{k,j}\right)_{i,j}\right],$$

i.e., the (i,j)-th entry of the resulting matrix is the sum $\sum_k b_{k,i}^* b_{k,j}$. The isomorphism $i : \mathcal{M}_n(\mathcal{B}(\mathcal{H})) \to \mathcal{M}_n \otimes \mathcal{B}(\mathcal{H})$ is defined as $i([a_{i,j}]) = \sum_{i,j} |i\rangle\langle j| \otimes a_{ij}$, where $\{|i\rangle\langle j|\}_{i,j=1}^n$ denotes any orthonormal basis for $\mathcal{M}_n$. As a result, $b := i([b_{i,j}]) = \sum_{i,j} |i\rangle\langle j| \otimes b_{ij}$ and $b^* := i([b_{i,j}^*]) = \sum_{i,j} |i\rangle\langle j| \otimes b_{ij}^*$. Next, we calculate:

$$\begin{aligned}
b^*b &= \left(\sum_{i,j} |j\rangle\langle i| \otimes b_{ij}^*\right)\left(\sum_{i,j} |i\rangle\langle j| \otimes b_{ij}\right) \\
&= \sum_{i,k,l,j} |i\rangle\langle k|\,|l\rangle\langle j| \otimes b_{k,i}^* b_{l,j} \\
&= \sum_{i,k,l,j} |i\rangle\langle j| \otimes \left(\sum_k b_{k,i}^* b_{k,j}\right) && (|i\rangle\langle k|\,|l\rangle\langle j| = \delta_{kl}\,|i\rangle\langle j|) \\
&= i([b_{i,j}]^*[b_{i,j}]),
\end{aligned}$$

thereby demonstrating that i is positive.

The equivalence of definitions 5.2.24 and 5.4.11, follows from the fact that the two diagrams below commute and the composition of positive maps is positive. For the first we assume that $\mathrm{id}_{\mathcal{T}(\mathcal{H})} \otimes \Phi$ is positive for all finite dimensional $\mathcal{H}$, and for the second that $\mathcal{M}_n(\Phi)$ is positive for all $m \in \mathbb{N}$.

$$\begin{tikzcd}
\mathcal{B}(\mathcal{H}\otimes\mathcal{K}_1) \arrow[r, "\mathrm{id}_{\mathcal{T}(\mathcal{H})}\otimes\Phi"] & \mathcal{B}(\mathcal{H}\otimes\mathcal{K}_2) \arrow[d, "\cong"] \\
\mathcal{M}_{nm} \arrow[u, "\cong"'] & \mathcal{M}_{no} \arrow[d, "\cong"] \\
\mathcal{M}_n\otimes\mathcal{M}_m \arrow[u, "\cong"'] & \mathcal{M}_n\otimes\mathcal{M}_o \arrow[d, "A\otimes B\mapsto[(a_{i,j}B)_{i,j}]"] \\
\mathcal{M}_n(\mathcal{M}_m) \arrow[u, "{[a_{i,j}]\mapsto\sum_{i,j}|i\rangle\langle j|\otimes a_{ij}}"'] \arrow[r, "\mathcal{M}_n(\Phi)"'] & \mathcal{M}_n(\mathcal{M}_o)
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{B}(\mathcal{H}\otimes\mathcal{K}_1) \arrow[r, "\mathrm{id}_{\mathcal{T}(\mathcal{H})}\otimes\Phi"] \arrow[d, "\cong"] & \mathcal{B}(\mathcal{H}\otimes\mathcal{K}_2) \\
\mathcal{M}_{nm} \arrow[d, "\cong"] & \mathcal{M}_{no} \arrow[u, "\cong"'] \\
\mathcal{M}_n\otimes\mathcal{M}_m \arrow[d, "A\otimes B\mapsto[(a_{i,j}B)_{i,j}]"] & \mathcal{M}_n\otimes\mathcal{M}_o \arrow[u, "\cong"'] \\
\mathcal{M}_n(\mathcal{M}_m) \arrow[r, "\mathcal{M}_o(\Phi)"'] & \mathcal{M}_n(\mathcal{M}_o) \arrow[u, "{[a_{i,j}]\mapsto\sum_{i,j}|i\rangle\langle j|\otimes a_{ij}}"']
\end{tikzcd}$$

□

The following result we be useful later on.

Proposition 5.4.13 (Proposition 2.3)**.** *[22] Let $\Phi : \mathscr{A} \to \mathscr{B}$ be a $*$-homomorphism between C^*-algebras. Then Φ is completely positive.*

Direct sums of C^*-Algebras

Definition 5.4.14. One can form products (in the categorical sense) of C^*-algebras as follows. Let $\mathscr{A}_i$ be a C^*-algebra for each i in some index set I. The *direct sum* of the family $(\mathscr{A}_i)_{i\in I}$ is the C^*-algebra denoted by $\bigoplus_{i\in I} A_i$, consisting of elements

$$a \in \prod_{i\in I} \mathscr{A}_i \quad \text{such that} \quad \sup_{i\in I} \|a(i)\| < \infty,$$

with operations defined coordinatewise and norm given by

$$\|a\| = \sup_{i \in I} \|a(i)\|.$$

Tensor products of C^*-Algebras

Definition 5.4.15. Given C^*-algebras $\mathscr{A}_1$ and $\mathscr{A}_2$, the *injective* C^**norm* on $\mathscr{A}_1 \odot \mathscr{A}_2$, the algebraic tensor product, is defined by

$$\|a\|_{\min} = \sup \left\{ \|(\pi_1 \odot \pi_2)(a)\| \right\}, \quad a \in \mathscr{A}_1 \odot \mathscr{A}_2,$$

where π_1 and π_2 run over all representations of $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively. The subscript $\min$ will be ommited unless ambiguity arises. The completion $A_1 \widehat{\otimes} A_2$ is called the *injective* C^*-*tensor product* of A_1 and A_2. The injective C^*-norm (respectively, C^*-tensor product) is also referred to as the *spatial* (or *spatial*) C^*-norm (respectively, C^*-tensor product).

Proposition 5.4.16. *[75, Proposition 1.22.7] Let $\mathscr{A}$ and $\mathscr{B}$ be C^*-algebras, and let α be a C^*-norm on the algebraic tensor product $\mathscr{A} \odot \mathscr{B}$ such that*

$$\|a \otimes b\|_\alpha \leq k\|a\| \cdot \|b\| \quad \textit{for all } a \in A,\ b \in B,$$

where k is a fixed positive constant. Then,

$$\| \cdot \|_{\min} \leq \| \cdot \|_\alpha.$$

We note that although [75] uses a diferent definition from [76] they are shown no be equivalent in [76, Theorem 4.9].

5.4.2 W^*-Algebras

The letters $\mathscr{M}, \mathscr{N}, \mathscr{T}$ will typically denote W^*-algebras.

Basics of W^*-Algebras

Definition 5.4.17. A W^*-algebra is a C^*-algebra $\mathscr{M}$ that admits a predual, i.e., a Banach space V together with an isometric isomorphism $V^* \cong \mathscr{M}$. It turns out that the predual of a W^*-algebra $\mathscr{M}$ is unique up to isometric isomorphism [75, Corollary 1.13.3].

Remark 5.4.18. In this work, W^*-algebras are unital by definition (since we assume C^*-algebras to be unital). However, W^*-algebras are always unital: if a C^*-algebra (not necessarily unital) admits a predual, then it must have a unit [75, Chapter 1.7].

Definition 5.4.19. The weak* topology on $\mathscr{M}$ induced by the predual is referred to as the *ultraweak topology*. A linear map between W^*-algebras is said to be *normal* if it is ultraweakly continuous. We denote the set of normal functionals on $\mathscr{M}$ by $\mathscr{M}_*$; it is standard that $\mathscr{M}_*$ is a predual of $\mathscr{M}$.

One of the most important examples of a W^*-algebra is $\mathcal{B}(\mathcal{H})$, the space of all bounded linear operators on a Hilbert space $\mathcal{H}$. The following proposition clarifies why $\mathcal{B}(\mathcal{H})$ qualifies as a W^*-algebra.

Proposition 5.4.20. *[74, Theorem 19.2] Let $\mathcal{H}$ be a Hilbert space. Let $\mathcal{B}(\mathcal{H})$ denote the space of bounded operators on $\mathcal{H}$, and let $\mathcal{T}(\mathcal{H})$ denote the space of trace-class operators on $\mathcal{H}$. The dual of $\mathcal{T}(\mathcal{H})$ is isometrically isomorphic to $\mathcal{B}(\mathcal{H})$ via the map*

$$\Phi : \mathcal{B}(\mathcal{H}) \to \mathcal{T}(\mathcal{H})^*, \quad \Phi(T)(-) = \mathrm{tr}(T(-)), \quad \textit{for all } A \in \mathcal{T}(\mathcal{H}).$$

Example 5.4.21. The following are examples of W^*-algebras.

1. For a Hilbert space $\mathcal{H}$, $\mathcal{B}(\mathcal{H})$ is a W^*-algebra (see the Proposition immediately above).
2. $\mathcal{M}_n \cong \mathcal{B}(\mathbb{C}^n)$ is a W^*-algebra. Its predual is itself $\mathcal{M}_n \cong \mathcal{T}(\mathbb{C}^n)$ equipped with the trace norm.

Remark 5.4.22. While W^*-algebras and von Neumann algebras are often used interchangeably (*e.g.*, in [77]), the latter typically denotes concrete ultraweakly closed C^*-subalgebras of $\mathcal{B}(\mathcal{H})$.

Definition 5.4.23. We denote the category of W^*-algebras and normal completely positive subunital maps by $\mathsf{W}^*_{\mathsf{CPSU}}$.

Direct Sums of W^*-Algebras

Direct sums of W^*-algebras are defined as in Definition 5.4.14.

Proposition 5.4.24. *[77, Exercise 47 IV] Let $(\mathscr{M}_i)_i$ be a family of W^*-algebras. Then the direct sum $\bigoplus_i \mathscr{M}_i$ (see Definition 5.4.14) is itself a W^*-algebra, and the canonical projections*

$$\pi_j : \bigoplus_i A_i \to A_j, \quad \textit{given by } \pi_j(a) = a(j),$$

are normal. Moreover, this makes $\bigoplus_i \mathscr{M}_i$ the categorical product of $\mathscr{M}_i$ in the category $\mathsf{W}^*_{\mathsf{CPSU}}$.

Tensor products of W^*-Algebras

Here we adopt the definition of the spatial tensor product of W^*-algebras from [77], rather than the more common approach based on ultraweak completion of the spatial tensor product of C^*-algebras (Definition 5.4.15) as in [75, 76]. The approach in [77] is more abstract, not resorting to representations on Hilbert spaces. Nevertheless, the author proves that the standard one is a particular realization of his definition [77, Theorem 111 VII].

Definition 5.4.25. A bilinear map $\beta : \mathscr{M} \times \mathscr{N} \to \mathscr{T}$ between W^*-algebras is said to be:

1. *unital* if $\beta(1, 1) = 1$,
2. *multiplicative* if $\beta(m_1m_2, n_1n_2) = \beta(m_1, n_1)\,\beta(m_2, n_2)$ for all $m_1, m_2 \in \mathscr{M}$, $n_1, n_2 \in \mathscr{N}$,
3. *involution preserving* if $\beta(m, n)^* = \beta(m^*, n^*)$ for all $n \in \mathscr{M}$, $m \in \mathscr{N}$.

From now on, we will refer to a bilinear map that is multiplicative, involution preserving, and unital as a *miu-bilinear map*.

Definition 5.4.26. A miu-bilinear map $\gamma : \mathscr{M} \times \mathscr{N} \to \mathscr{T}$ between W^*-algebras is called a *tensor product of $\mathscr{M}$ and $\mathscr{N}$* when it satisfies the following three conditions:

1. The range of γ generates $\mathscr{T}$, meaning that the linear span of the image of γ is ultraweakly dense in $\mathscr{T}$. This implies that for all $f \in \mathscr{M}_*$ and $g \in \mathscr{N}_*$, there exists *at most one* $h \in \mathscr{T}_*$ such that

$$h(\gamma(n, m)) = f(n)g(m) \quad \text{for all } n \in \mathscr{M},\ m \in \mathscr{N}.$$

   We call such an h the *product functional* for f and g, and denote it by $\gamma(f, g)$ (when it exists).

2. For all normal positive functionals $\sigma : \mathscr{M} \to \mathbb{C}$ and $\tau : \mathscr{N} \to \mathbb{C}$, the product functional $\gamma(\sigma, \tau) : \mathscr{T} \to \mathbb{C}$ exists and is positive.
3. The product functionals $\gamma(\sigma, \tau)$ of normal positive functionals σ and τ form a *faithful collection* of normal positive functionals on $\mathscr{T}$ (*i.e.*, $t \in \mathscr{T}_+$ is zero iff $\gamma(\sigma, \tau)(t) = 0$ for all such functionals).

   We will usually designate this bilinear map by $\overline{\otimes}$.

Definition 5.4.27. A *basic functional* is a map $\omega : \mathscr{M} \odot \mathscr{N} \to \mathbb{C}$ with

$$\omega \equiv (\varphi_1 \odot \varphi_2)(t^*(\cdot)t)$$

for some normal positive maps $\varphi_1 : \mathscr{M} \to \mathbb{C}$, $\varphi_2 : \mathscr{N} \to \mathbb{C}$, and $t \in \mathscr{M} \odot \mathscr{N}$. A *simple functional* is a finite sum of basic functionals.

Definition 5.4.28. The tensor product norm on $\mathscr{M} \odot \mathscr{N}$ is the norm given by

$$\|t\|_{w^*} = \sup\{w(t^*t)^{\frac{1}{2}} \mid \omega(1) \leq 1\},$$

where ω ranges over all basic functionals.

Once $\mathscr{M} \odot \mathscr{N}$ is equipped with the tensor product norm, we may consider bounded linear functionals on $\mathscr{M} \odot \mathscr{N}$, along with the corresponding operator norm. The basic and simple functionals are bounded, as noted in [77, Definition 112 II (3)].

Definition 5.4.29. The *ultraweak tensor product topology* is the least topology on $\mathscr{M} \odot \mathscr{N}$ that makes all operator norm limits of simple functionals continuous.

Next, we recall the algebraic tensor product from Definition 4.1.8 and introduce the notation $\beta_\odot$ for the unique linear map $V \odot W \to R$ induced by the universal property of the tensor product.
A bilinear map $\beta\colon \mathscr{M} \times \mathscr{N} \to \mathscr{T}$ between W^*-algebras is:

1. *bounded* when the unique extension $\beta_\odot\colon \mathscr{M} \odot \mathscr{N} \to \mathscr{T}$ is bounded;
2. *normal* when $\beta_\odot$ is continuous with respect to the ultraweak tensor product topology on $\mathscr{M} \odot \mathscr{N}$ and the ultraweak topology on $\mathscr{T}$.

The following theorem establishes a universal property analogous to Definition 4.1.8, but for the W^*-tensor product rather than the algebraic case. Later, in Section 5.5.2, we will make use of this result for proving that $\mathrm{W}^*_{\mathrm{CPSU}}$ is a first-order model.

Theorem 5.4.30. *[77, Theorem 112 XI] A tensor product $\gamma\colon \mathscr{M} \times \mathscr{N} \to \mathscr{T}$ of W^*-algebras $\mathscr{M}$ and $\mathscr{N}$ satisfies the following universal property: for every normal bounded bilinear map $\beta\colon \mathscr{M} \times \mathscr{N} \to \mathscr{O}$ into a von Neumann algebra $\mathscr{O}$, there exists a unique ultraweakly continuous map $\beta_\gamma\colon \mathscr{T} \to \mathscr{O}$ such that $\beta_\gamma \circ \gamma = \beta$. Moreover, $\|\beta_\gamma\|_{op} = \|\beta_\odot\|_{op}$, where $\beta_\odot\colon \mathscr{M} \odot \mathscr{N} \to \mathscr{O}$.*

Proposition 5.4.31. *[[77](), Exercise 114 II] The tensor product of W^*-algebras $\mathscr{M}$ and $\mathscr{N}$ is unique in the sense that when $\gamma\colon \mathscr{M}\times\mathscr{N}\to\mathscr{T}$ and $\gamma'\colon \mathscr{M}\times\mathscr{N}\to\mathscr{T}'$ are tensor products of $\mathscr{M}$ and $\mathscr{N}$, then there is a unique normal miu-isomorphism $\varphi\colon \mathscr{T}\to\mathscr{T}'$ with $\varphi(\gamma(a,b)) = \gamma'(a,b)$ for all $a\in\mathscr{M}$ and $b\in\mathscr{N}$. In other words, the tensor product of W^*-algebras $\mathscr{M}$ and $\mathscr{N}$ is unique up to unique normal miu-isomorphism. Note that φ is a $*$-isomorphism and therefore an isometry.*

Since the tensor product is unique up to unique normal miu-isomorphism, we may fix a choice and denote it by $\overline{\otimes}\colon \mathscr{M}\times\mathscr{N}\to\mathscr{M}\,\overline{\otimes}\,\mathscr{N}$.
The results that follow will be useful for demonstrating that $\mathsf{W}^*_{\mathsf{CPSU}}$ is a first-order model in Section 5.5.2.

Proposition 5.4.32. *[77, Proposition 115 II] Given normal completely positive maps $f:\mathscr{M}\to\mathscr{T}$ and $g:\mathscr{N}\to\mathcal{S}$ between W^*-algebras, there exists a unique normal completely positive map*

$$f\overline{\otimes}g : \mathscr{M}\,\overline{\otimes}\,\mathscr{N}\to\mathscr{T}\,\overline{\otimes}\,\mathscr{S}$$

such that

$$(f\overline{\otimes}g)(m\otimes n) = f(m)\otimes g(n) \quad \text{for all } m\in\mathscr{M},\ n\in\mathscr{N}.$$

Moreover $f\overline{\otimes}g$ is (sub)unital if f and g are (sub)unital.

Proposition 5.4.33. *[77, Proof 115 III] Let $\mathscr{M}$ and $\mathscr{N}$ be W^*-algebras. Given normal completely positive maps $f:\mathscr{M}\to\mathscr{T}$ and $g:\mathscr{N}\to\mathscr{T}$. We may take $f\,\overline{\otimes}\,g = \beta_{\overline{\otimes}}$ as in the theorem Theorem 5.4.30. Moreover, $\|\beta_{\odot}(s)\|_{w^*}\leq\|f\|_{op}\|g\|_{op}\|s\|_{w^*}$, given an element $s\in\mathscr{M}\overline{\otimes}\mathscr{N}$.*

Proposition 5.4.34. *[77, Exercise 116 III] Let $\mathscr{M}$ and $\mathscr{N}$ be W^*-algebras, $\|m\,\overline{\otimes}\,n\|_{w^*} = \|m\|\,\|n\|$ for all $m\in\mathscr{M}$ and $n\in\mathscr{N}$.*

Proposition 5.4.35. *[77, Corollary 119 IV] There is a unique normal $*$-isomorphism*

$$\alpha_{\mathscr{M},\mathscr{N},\mathscr{T}} : \mathscr{M}\otimes(\mathscr{N}\otimes\mathscr{T})\longrightarrow(\mathscr{M}\otimes\mathscr{N})\otimes\mathscr{T},$$

called an associator*, with*

$$\alpha_{\mathscr{M},\mathscr{N},\mathscr{T}}(m\otimes(m\otimes o)) = (m\otimes m)\otimes o$$

for all $m\in\mathscr{M}$, $n\in\mathscr{N}$, $o\in\mathscr{T}$.

Proposition 5.4.36. *[77, Exercise 119 IVc] Let $\mathscr{M}$ and $\mathscr{N}$ be W^*-algebras. There exists a unique normal $*$-isomorphism*

$$\mathrm{sw}_{\mathscr{M},\mathscr{N}} : \mathscr{M}\overline{\otimes}\mathscr{N} \to \mathscr{N}\overline{\otimes}\mathscr{M},$$

called the braiding isomorphism*, satisfying*

$$\mathrm{sw}_{\mathscr{M},\mathscr{N}}(m \otimes n) = n \otimes m \quad \text{for all } n \in \mathscr{M}, m \in \mathscr{N}.$$

Distributivity

Proposition 5.4.37. *[77, Proposition 117 III] Given W^*-algebras $\mathscr{M}$ and $(\mathscr{N}_i)_{i\in I}$, we have a natural isomorphism*

$$\mathscr{M} \,\overline{\otimes} \bigoplus_{i\in I} \mathscr{N}_i \cong \bigoplus_{i\in I} \mathscr{M} \,\overline{\otimes}\, \mathscr{N}_i.$$

That is, the spatial tensor product distributes over (possibly infinite) direct sums.

Theorem 5.4.38. *[77, Theorem 119 V] Endowed with the tensor product, the category $\mathsf{W}^*_{\mathsf{CPSU}}$ is a symmetric monoidal category, with $\mathbb{C}$ as the unit object.*

Theorem 5.4.39. *The category $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ is a distributive symmetric monoidal category.*

Proof. It follows directly from Propositions 5.4.24 and 5.4.37 and Theorem 5.4.38 □

The following result will be useful for demonstrating that $\mathsf{W}^*_{\mathsf{CPSU}}$ is a first-order model in Section 5.5.2.

Theorem 5.4.40. *[22, Theorem 3.2] Let $\mathscr{A}, \mathscr{B}, \mathscr{C}$ be W^*-algebras. Then the canonical map*

$$\langle \mathrm{id}\overline{\otimes}\pi_1, \mathrm{id}\overline{\otimes}\pi_1\rangle : \mathscr{A}\overline{\otimes}(\mathscr{B}\times\mathscr{C}) \to (\mathscr{A}\overline{\otimes}\mathscr{B}) \times (\mathscr{A}\overline{\otimes}\mathscr{C})$$

is a unital $*$-isomorphism, and therefore, isometric.

5.5 Categories for (first-order) quantum computation

We will now explore different potential metric models for quantum computation. A perhaps surprising point is that the categories that "naturally arise" in quantum computation are first-order, and therefore we will work in this setting. In other words, we will now work with categories that do not need to be closed. Note, however, that this does not preclude the interpretation of λ-calculus. In fact, one of our contributions is to provide the necessary ingredients to embed these categories into closed ones, which are indeed models of metric λ-calculus with conditionals. We do not detail how such embeddings work, for they involve advanced categorical machinery which falls out of this dissertation's scope [104]. Alternatively, we can also consider a "first-order λ-calculus" in which the type $\mathbb{A} \multimap \mathbb{B}$ is not allowed.

We divide this section into two parts, each corresponding to a different formulation of quantum theory. In the first part, we consider Schrödinger's picture, where quantum programs are interpreted as maps between quantum states (*i.e.*, density operators). Here, we study Selinger's category Q [21]. In the second part, we adopt Heisenberg's picture, in which programs are modeled as maps between observables (*i.e.*, self-adjoint operators). In this setting, we explore Cho's category $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$, the opposite of the category $\mathsf{W}^*_{\mathsf{CPSU}}$, whose objects are W^*-algebras and morphisms are completely positive subunital maps between them. We highlight the following connection between the two categories: the full subcategory of $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ consisting of finite-dimensional W^* algebras is equivalent to Q [22].

5.5.1 Schrödinger's picture

We begin by presenting the category CPTP of completely positive trace-preserving maps, which was shown in [1] to form a symmetric monoidal Met-category. In this section $\mathcal{M}_n$ will denote the vector space of complex $n \times n$ matrices.

Definition 5.5.1. The category CPTP is the category whose objects are natural numbers $n \geq 1$ and whose morphisms $n \to m$ are quantum channels $\mathcal{M}_n \to \mathcal{M}_m$.

A natural candidate for interpreting quantum programs with coproducts is the category CPTP$^+$, obtained via the coproduct cocompletion of CPTP. However, for CPTP$^+$ to serve as a suitable model for quantum computation, its morphisms should be able to express the measurement operation, *i.e.*, an operation mapping a density matrix $\rho \in \mathcal{M}_2$ to a classical bit $b \in \mathbb{C} \oplus \mathbb{C}$. Unfortunately, this is not the case. If such a measurement were to exist, it would correspond to a morphism $\Phi \colon 2 \to 1 \oplus 1$, where $\oplus$ is the coproduct, allowing us to reason about it within our metric equational system. Simultaneously, we should be able to "decompose" Φ into two projection maps $\pi_1 \colon 2 \to 1$ and $\pi_2 \colon 2 \to 1$. Now, recall that in CPTP$^+$, a morphism consists of a pair $(f, (\phi_i)_{i \in I})$, where: $f \colon I \to J$ is a function between index sets, and $(\phi_i)_{i \in I}$ is a family of CPTP-morphisms $\phi_i \colon A_i \to B_{f(i)}$. Because f is a function, the category does not support morphisms of the form $(f, \phi_i \colon A \to C_i)_{i \in I}$ when I has more than one element. As a result, we cannot express measurements in CPTP$^+$. We could consider introducing measurements via a "product completion," but a similar problem would arise for the injections. This suggests that $+$ should be a biproduct. Nonetheless, as we will see, not all morphisms of the biproduct are necessary: the coproduct structure together with the projections suffices.

Definition 5.5.2. The category Q is defined as

- An object is a signature $\sigma = n_1, \ldots, n_s$. We denote these signatures by the Greek letters σ, τ and μ.
- Given signatures $\sigma = n_1, \ldots, n_s$ and $\tau = m_1, \ldots, m_t$, a morphism $\Phi \in \sigma \to \tau$ is a matrix
$$\begin{pmatrix} \Phi_{11} & \ldots & \Phi_{s1}, \\ \vdots & \ddots & \vdots \\ \Phi_{1t} & \ldots & \Phi_{ts} \end{pmatrix}$$
of arrows $\Phi_{ij} : \mathcal{M}_{n_i} \to \mathcal{M}_{m_j}$ in CP which is trace-nonincreasing, *i.e.*, the following condition holds:
$$\sum_j \sum_i \mathsf{Tr}\left(\Phi_{ij}(A_i)\right) \leq \sum_i \mathsf{Tr}\left(A_i\right)$$
for all positive $A_i \in \mathcal{M}_{n_i}$.

Remark 5.5.3. Q corresponds to the finite biproduct completion of CP (which extends CPTP to include all completely positive maps), further restricted to trace-nonincreasing morphisms.

Every signature σ is associated with a complex vector space $\mathcal{M}_\sigma := \mathcal{M}_{n_1} \oplus \cdots \oplus \mathcal{M}_{n_s}$ This space consists of matrix vectors

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_n \end{pmatrix}$$

where the signature σ specifies both the number of matrices, s, and their respective dimensions, $n_i \times n_i$. The elements of $\mathcal{M}_\sigma$ are represented uppercase letters such as A, B, etc. We also establish that $\mathbb{C}^\sigma := \mathbb{C}^{n_1} \oplus \cdots \oplus \mathbb{C}^{n_s}$.

Note that the definition of trace induces a generalized notion of trace applicable to elements of $\mathcal{M}_\sigma$. Specifically, for $A = \begin{pmatrix} A_1 & \ldots & A_n \end{pmatrix}^T$, we have $\mathsf{Tr}(A) = \sum_{i=1}^n \mathsf{Tr}(A_i)$.

Definition 5.5.4. [*Coproduct*] Concatenation $\sigma \oplus \sigma'$ of signatures σ and σ' yields coproducts in Q. The co-pairing map $[\Phi, \Psi] : \sigma \oplus \sigma' \to \tau$ is defined as $[\Phi, \Psi](A, B) = \Phi(A) + \Psi(B)$, where addition uses the fact that the codomain is always a direct sum of vector spaces.

Remark 5.5.5. The category Q does not have finite products. To see this, observe that the diagonal morphism

$$\langle \mathrm{id}, \mathrm{id} \rangle : \tau \to \tau \oplus \tau$$

is not trace-nonincreasing and hence is not a valid morphism. However, Q does contain the two projection morphisms

$$\pi_1 : \sigma \oplus \sigma \to \sigma' \quad \text{and} \quad \pi_2 : \sigma \oplus \sigma \to \sigma'.$$

Definition 5.5.6. [*Tensor Product*] For signatures $\sigma = n_1, \ldots, n_s$ and $\tau = m_1, \ldots, m_t$, the tensor product of σ and τ is defined as $\sigma \otimes \tau = n_1 m_1, \ldots, n_1 m_t, \ldots, n_s m_1, ..., n_s m_t$. The morphism part of the tensor product follows the definition in the category of vector spaces. If $\Phi : \sigma \to \tau$ and $\Psi : \sigma' \to \tau'$, then their tensor product $\Psi \otimes \Phi : \sigma \otimes \sigma' \to \tau \otimes \tau'$ is the Kronecker product of their matrices representation, *i.e.*,

$$\Phi \otimes \Psi = \begin{pmatrix} \Phi_{11} \otimes \Psi_{11} & \ldots & \Phi_{11} \otimes \Psi_{s'1} & \ldots & \Phi_{s1} \otimes \Psi_{11} & \ldots & \Phi_{s1} \otimes \Psi_{s'1} \\ \vdots & & & & & & \vdots \\ \Phi_{1t} \otimes \Psi_{1t'} & \ldots & \Phi_{1t} \otimes \Psi_{s't'} & \ldots & \Phi_{st} \otimes \Psi_{1t'} & \ldots & \Phi_{st} \otimes \Psi_{s't'} \end{pmatrix}$$

Moreover, dist is an identity map:

$$(\sigma \oplus \sigma') \otimes \tau = (\sigma \otimes \tau) \oplus (\sigma' \otimes \tau)$$

The category Q is a distributive symmetric monoidal category with binary coproducts. However, this category is not closed [105].

Nota para mim mesma ir definir $\bigoplus_i A_i$ na secção de espaços de Hilbert

First, given signatures $\sigma : n_1, \ldots, n_s$ and $\tau : m_1, \ldots, m_t$, note the following inclusion:

$$\begin{aligned} \mathcal{M}_{n_1} \oplus \cdots \oplus \mathcal{M}_{n_s} &\overset{\iota_\sigma}{\hookrightarrow} \mathcal{T}(\mathbb{C}_{n_1} \oplus \cdots \oplus \mathbb{C}_{n_s}) \\ (A_1, \ldots, A_s) &\mapsto \bigoplus_i A_i \end{aligned}$$

Then, we define $\iota_{(\sigma\otimes\tau)'} : \mathcal{M}_{\sigma\otimes\tau} \to \mathcal{T}(\mathbb{C}_\sigma \otimes \mathbb{C}_\tau)$ as $\iota_{(\sigma\otimes\tau)'} = (\mathrm{dist}')^{-1} \cdot \iota_{(\sigma\otimes\tau)}$, where

$$(\mathrm{dist}')^{-1}(a)$$

Voltar aqui depois.

With these inclusions established, we can now define a

Proposition 5.5.7. *For all* $\Phi, \Phi' \in \mathsf{Q}(\sigma, \mu)$ *and* $\Psi, \Psi' \in \mathsf{Q}(\tau, \mu)$*, it holds that*

$$\|[\Phi - \Phi', \Psi - \Psi']\|_{\diamondsuit} \leq \sup\{\Phi - \Phi', \Psi - \Psi'\}.$$

Proof. Attending to the definition of the diamond norm, the fact that dist is an identity and Lemma 4.2.2 we compute:

$$\begin{aligned}
&\|[\Phi, \Psi]\|_{\diamondsuit\,\mathsf{gen}} \\
&= \left\|[\Phi, \Psi] \otimes \mathrm{id}_{\mathcal{M}_{\sigma\oplus\tau}}\right\|_1 \\
&= \|[\Phi \otimes \mathrm{id}_{\mathcal{M}_\sigma}, \Psi \otimes \mathrm{id}_{\mathcal{M}_\tau}]\|_1 && (\mathrm{dist} = \mathrm{id}) \\
&\leq \sup\{\|\Phi \otimes \mathrm{id}_{\mathcal{M}_\sigma}\|_1, \|\Psi \otimes \mathrm{id}_{\mathcal{M}_\tau}\|_1\} && (\text{Lemma 4.2.2}) \\
&= \sup\left\{\|\Phi\|_{\diamondsuit}, \|\Psi\|_{\diamondsuit}\right\}
\end{aligned}$$

The inequality in Proposition 5.5.7 follows from the linearity of the co-pairing map. □

Corollary 5.5.8. *Let* $\sigma : n_1, \ldots, n_s$ *and* $\tau : m_1, \ldots, m_t$ *be signatures. Let* $\Phi : \sigma \to \tau$ *be a completely positive trace-nonincreasing super-operator.*

Proof. Given that Φ is a completely positive trace-nonincreasing super-operator, if follows that $\Phi \otimes \mathrm{id}_{\mathcal{M}_\sigma}$ is a positive trace-nonincreasing super-operator. Let $\Psi = \Phi \otimes \mathrm{id}$, it holds that,

$$\begin{aligned}
&\|\Phi\|_{\diamondsuit} = \|\Psi\|_1 \\
&= \max\left\{\mathsf{Tr}\left(\Psi(uu^\dagger)\right) \mid \|u\|_2 = 1\right\} && (\text{Theorem 5.2.30}) \\
&= \max\left\{\sum_i \sum_j \mathsf{Tr}\left(\Psi_{ij}(u_i u_i^\dagger)\right) \,\middle|\, \left\|\left(u_1, \ldots, u_s^2\right)^T\right\|_2 = 1\right\} \\
&\leq \max\left\{\sum_i \mathsf{Tr}\left(u_i u_i^\dagger\right) \,\middle|\, \sqrt{\sum_i \|u_i\|_2^2} = 1\right\} && (\Psi \text{ is trace-nonincreasing}) \\
&= 1
\end{aligned}$$

□

Proposition 5.5.9. *The category* Q *a symmetric monoidal* Met*-category.*

Proof. Here we follow the same reasoning as [1, Proof of Proposition 4.1].

First, we establish that Q is Met-enriched. By unpacking the relevant definitions, this reduces to proving the following: for all Q-morphisms $\Phi, \Phi' : \sigma \to \tau$ and $\Psi, \Psi' : \tau \to \mu$ the inequation $\|\Phi - \Phi'\|_\diamond + \|\Psi - \Psi'\|_\diamond \geq \|\Psi\Phi - \Psi'\Phi'\|_\diamond$ holds. We proceed as follows:

$$\begin{aligned}
&\|\Phi - \Phi'\|_\diamond + \|\Psi - \Psi'\|_\diamond \\
&\geq \|(\Phi - \Phi')\Psi\|_\diamond + \|\Phi'(\Psi - \Psi')\|_\diamond && (\text{Proposition 5.2.31, and Corollary 5.5.8}) \\
&= \|\Phi\Psi - \Phi'\Psi\|_\diamond + \|\Phi'\Psi - \Phi'\Psi'\|_\diamond \\
&\geq \|\Phi\Psi - \Phi'\Psi + \Phi'\Psi - \Phi'\Psi'\|_\diamond && \{\text{Triangle inequality}\} \\
&= \|\Psi\Phi - \Psi'\Phi'\|_\diamond .
\end{aligned}$$

Next, to prove that $\|\Phi - \Phi'\|_\diamond + \|\Psi - \Psi'\|_\diamond \geq \|\Psi \otimes \Phi - \Psi' \otimes \Phi'\|_\diamond$, we calculate

$$\begin{aligned}
&\|\Psi - \Psi'\|_\diamond + \|\Phi - \Phi'\|_\diamond \\
&\geq \|\mathrm{id} \otimes (\Psi - \Psi')\|_\diamond + \|\mathrm{id} \otimes (\Phi - \Phi')\|_\diamond && (\text{Theorem 5.2.32 and Corollary 5.5.8 }) \\
&= \|\mathrm{id} \otimes \Psi - \mathrm{id} \otimes \Psi'\|_\diamond + \|\mathrm{id} \otimes \Phi - \mathrm{id} \otimes \Phi'\|_\diamond \\
&\geq \|(\mathrm{id} \otimes \Psi) \cdot (\Phi \otimes \mathrm{id}) - (\mathrm{id} \otimes \Psi') \cdot (\Phi' \otimes \mathrm{id})\|_\diamond && (\text{Q is aMet-category}) \\
&= \|\Psi \otimes \Phi - \Psi' \otimes \Phi'\|_\diamond
\end{aligned}$$

□

Theorem 5.5.10. Q *is a distributive symmetric monoidal* Met*-category.*

Proof. It follows directly from Proposition 5.5.7 and Proposition 5.5.9. □

5.5.2 Heisenberg's picture

In the previous quantum model, we considered the Schrödinger picture—that is, morphisms between quantum states (*i.e.*, density operators). In [22], the author presents a model in the Heisenberg picture, where maps are between observables (*i.e.*, self-adjoint operators), which can be seen as an infinite-dimensional extension of Selinger's model. This model is given by the category $(\mathrm{W}^*_{\mathrm{CPSU}})^{\mathrm{op}}$, the opposite of the category $\mathrm{W}^*_{\mathrm{CPSU}}$ whose objects are W*-algebras and morphisms are completely positive subunital maps between them. It is shown that Selinger's category Q is equivalent (up to categorical equivalence) to the finite-dimensional subcategory of $(\mathrm{W}^*_{\mathrm{CPSU}})^{\mathrm{op}}$,$(\mathrm{FdW}^*_{\mathrm{CPSU}})^{\mathrm{op}}$ [22]. Now, we will prove that $(\mathrm{W}^*_{\mathrm{CPSU}})^{\mathrm{op}}$ is a model of our lambda calculus.

Q **and** $(\mathsf{FdW}^*_{\mathsf{CPSU}})^{\mathsf{op}}$

Vai passar para o Trabalho futuro

In [22], the author established an equivalence of categories $\mathsf{Q} \simeq {\mathsf{FdW}^*_{\mathsf{CPSU}}}^{\mathrm{op}}$. Using this equivalence, we can induce an alternative norm on Q by assigning to each super-operator $\Phi \in \mathsf{Q}$ the norm of its corresponding map $\Phi^* \in {\mathsf{FdW}^*_{\mathsf{CPSU}}}^{\mathrm{op}}$.

Now we describe how to obtain Φ^* from Φ. Recall that for a Hilbert space $\mathcal{H}$, $\mathcal{B}(\mathcal{H})$, *i.e.*, the set of bounded operators on $\mathcal{H}$, is a W^*-algebra with the predual $\mathcal{T}(\mathcal{H})$. A quantum channel (resp. quantum operator) $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ defines a linear mapping $\Phi^* : \mathcal{B}(\mathcal{K}) \to \mathcal{B}(\mathcal{H})$ completely positive and unital (resp. subunital) [22, Proposition 5.1,]. These maps are related in the following way:

$$\mathrm{Tr}\left[\Phi(T)\cdot S\right] = \mathrm{Tr}\left[T \cdot \Phi^*(S)\right]$$

which holds for all trace-class operators $T \in \mathcal{T}(\mathcal{H})$ and all bounded operators $S \in \mathcal{B}(\mathcal{K})$. Another, perhaps more useful, way to describe the relationship between Φ and Φ^* in the finite-dimensional setting—where all trace-class operators are bounded and we have the identification $\mathcal{B}(\mathcal{H}) = \mathcal{T}(\mathcal{H})$—is given by the diagram below:

$$\begin{tikzcd}
\mathcal{T}(\mathcal{H}) \arrow[d, "\Phi"] & \mathcal{T}(\mathcal{H})^* \arrow[rr, "\varphi \mapsto \sum_{ij} \varphi(|j\rangle\langle i|)\cdot |i\rangle\langle j|"] & & \mathcal{B}(\mathcal{H}) \\
\mathcal{T}(\mathcal{K}) & \mathcal{T}(\mathcal{K})^* \arrow[u, "\varphi \mapsto \varphi\cdot\Phi"'] & & \mathcal{B}(\mathcal{K}) \arrow[ll, "A \mapsto (B \mapsto \mathrm{Tr}(AB))"] \arrow[u, dotted, "\Phi^*"']
\end{tikzcd}$$

In other words, the map $\phi^* : \mathcal{B}(\mathcal{K}) \to \mathcal{B}(\mathcal{H})$ can be defined as

$$\phi^*(A) = \sum_{i,j} \mathrm{tr}\left(A \cdot \phi(|j\rangle\langle i|)\right) \cdot |i\rangle\langle j|\,,$$

where i and j range over an orthonormal basis of $\mathcal{H}$.

$(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ **forms a first-oder model**

We start with some considerations on the choice of norm for morphisms in $\mathsf{W}^*_{\mathsf{CPSU}}$. The norm on morphisms between C^*-algebras faces an issue analogous to the trace norm in the context of quantum operations: there exists a positive unital isometry $\Phi\colon \mathscr{A} \to \mathscr{A}$ such that

$$\|\Phi \odot \mathrm{id}_{\mathscr{A}} \colon \mathscr{A} \odot \mathscr{A} \to \mathscr{A} \odot \mathscr{A}\|_{\mathsf{op}} = \infty,$$

i.e. the map $\Phi \odot \mathrm{id}_{\mathscr{A}}$ is unbounded under the usual operator norm [106, Prop. 3.5.2]. Due to these limitations, the *completely bounded norm* becomes the natural choice for studying maps between C^*-algebras [102, 103].

Definition 5.5.11. Given a map $\Phi\colon \mathscr{A} \to \mathscr{B}$, the completely bounded norm is defined as

$$\|\Phi\|_{\mathsf{cb}} = \sup_n \|\mathcal{M}_n(\Phi)\|_{\mathsf{op}} \,,$$

and can alternatively be expressed as

$$\|\Phi\|_{\mathsf{cb}} = \sup_{\mathscr{C}} \left\|\mathrm{id}_{\mathscr{C}} \check{\otimes} \Phi\right\|_{\mathsf{op}} = \sup_n \left\|\mathrm{id}_{\mathcal{M}_n} \check{\otimes} \Phi\right\|_{\mathsf{op}} \,,$$

where the supremum ranges over all C^*-algebras $\mathscr{C}$ [107, Introduction, p. 4]. A map Φ is said to be *completely bounded* if $\|\Phi\|_{\mathsf{cb}}$ is finite.

In the context of W^*-algebras, since we are concerned with the tensor product of W^*-algebras, $\overline{\otimes}$, we introduce a similar norm —which, to the best of our knowledge, has not been previously studied in the literature.

Definition 5.5.12. Let $\Phi : \mathscr{N}_1 \to \mathscr{N}_2$ be a normal map between W^*-algebras. The W^* *completely bounded norm* is defined as follows,

$$\|\Phi\|_{\mathsf{cb}w^*} = \sup_{\mathscr{M}} \|\mathrm{id}_{\mathscr{M}} \overline{\otimes} \Phi\|_{\mathsf{op}} \,,$$

where the supremum ranges over all W^*-algebras $\mathscr{M}$. Φ is said to be W^**-completely bounded* if $\|\Phi\|_{\mathsf{cb}w^*}$ is finite.

Eu sei que a motivação não é a melhor mas, arranjar exemplos é deficil

Our motivation for the use of this norm, beyond the fact that it satisfies the requirements for making $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ a model, has to do with the fact that it significantly simplifies calculations, since we are not handling two different tensor products, $\check{\otimes}$ and $\overline{\otimes}$, simultaneously. Moreover, if it turns out that the completely bounded norm satisfies $\|\mathrm{id}\overline{\otimes}\Phi\|_{\mathsf{cb}}$ for a normal map Φ between W^*-algebras, then the remaining results required for the completely bounded norm to give rise to a model of the metric lambda calculus follow from the inequality $\|\Phi\|_{\mathsf{cb}} \leq \|\Phi\|_{\mathsf{cb}w^*}$ (as shown in Proposition 5.5.14). We now establish a few results of the W^* completely bounded norm, so we can infer that $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ is a (first-oder) model of our metric lambda calculus.

Proposition 5.5.13. *Given completely bounded normal maps $\Phi : \mathscr{M}_1 \to \mathscr{N}_1$ and $\Psi : \mathscr{M}_2 \to \mathscr{N}_2$ between W^* algebras, it holds that:*

$$\left\|\Phi\check{\otimes}\Psi\right\|_{op} \leq \left\|\Phi\overline{\otimes}\Psi\right\|_{op}.$$

Proof. Given Propositions 5.4.16 and 5.4.34, we have that

$$\|m\|_{min} \leq \|m\|_{w^*} \qquad \text{for all } m \in \mathscr{M}_1\check{\otimes}\mathscr{M}_2.$$

Thus, in order to prove that the above proposition holds, it suffices to prove that for all $m \in \mathscr{M}_1\check{\otimes}\mathscr{M}_2$, $\Phi \check{\otimes} \Psi(m) = \Phi \overline{\otimes} \Psi(m)$.

Moreover, given that for any completely bounded maps Φ and Ψ between C^*-algebras, $\left\|\Phi\check{\otimes}\Psi\right\|_{\text{cb}} = \|\Phi\|_{\text{cb}} \|\Psi\|_{\text{cb}}$ ([102, Theorem 12.3]), and $\|\Phi\|_{\text{cb}} \geq \|\Phi\|_{\text{op}}$, it follows that $\Phi \odot \Psi$ is Lipschitz continuous and therefore uniformly continuous.

$$\begin{tikzcd}
V \times W \arrow[r, "T"] \arrow[d, "\odot"] & Z \\
V \odot W \arrow[ur, dashed, "T'"] \arrow[d, hook, "\iota"] & \\
V\check{\otimes}W \arrow[uur, dashed, "\check{T}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
V \times W \arrow[r, "T"] \arrow[d, "\odot"] & Z \\
V \odot W \arrow[d, hook, "\iota"] & \\
V\check{\otimes}W \arrow[uur, dashed, "\check{T}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
Z & & \\
V \odot W \arrow[u, dashed, "T'"] \arrow[d, hook, "\iota_1"] \arrow[drr, hook, "\iota_2"] & & \\
V\check{\otimes}W \arrow[uu, dashed, bend left=60, "\check{T}"] \arrow[rr, hook, "\iota_3"] & & V\overline{\otimes}W \arrow[uull, dashed, "\overline{T}"']
\end{tikzcd}$$

In the third diagram, we reason as follows: from the first diagram above, we obtain $\check{T} \cdot \iota_1 = T = \overline{T} \cdot \iota_2$, then, given that $\iota_2 = \iota_3 \cdot \iota_1$, we have $\check{T} \cdot \iota_1 = \overline{T} \cdot \iota_3 \cdot \iota_1$, from where if follows that $\overline{T} \circ \iota_3 = \check{T}$.

We note that since [75] and [76] employ the standard completion, we may assume it satisfies the usual properties.

□

Ou Corolário ?

Proposition 5.5.14. *Given a completely bounded normal map $\Phi : \mathscr{N}_1 \to \mathscr{N}_1$, between W^* algebras, it holds that:*

$$\|\Phi\|_{cb} \leq \|\Phi\|_{cbw^*}.$$

Proof. This follows from Proposition 5.5.13, the definition of both norms (Definition 5.5.11, Definition 5.5.12), the definition of *supremum* as the least upper bound, and the fact that $\mathcal{M}_n$ is a W^*-algebra. We reason as follows,

$$\begin{aligned} \|\Phi\|_{\mathsf{cb}} &= \sup_n \left\|\mathrm{id}_{\mathcal{M}_n} \check{\otimes} \Phi\right\|_{\mathsf{op}} \\ &\leq \sup_n \|\mathrm{id}_{\mathcal{M}_n} \overline{\otimes} \Phi\|_{\mathsf{op}} && (\text{Proposition 5.5.13}) \\ &\leq \sup_{\mathscr{M}} \|\mathrm{id}_{\mathscr{M}} \overline{\otimes} \Phi\|_{\mathsf{op}} = \|\Phi\|_{\mathsf{cb}w^*} . \end{aligned}$$

□

Proposition 5.5.15. *Given normal completely positive maps Φ and Ψ between W^*-algebras, it holds that*

$$\|\phi \,\bar{\otimes}\, \psi\|_{op} \leq \|\Phi\| \, \|\Psi\| .$$

Proof. It follows directly from Theorem 5.4.30 and Proposition 5.4.33. □

Corollary 5.5.16. *Given a normal completely positive subunital map Φ between W^*-algebras, it holds that*

$$\|\Phi\|_{cbw^*} = \|\Phi\|_{op} .$$

Proof. It follows from the proposition above and the definition of W^* completely bounded norm (Definition 5.5.12) that $\|\Phi\|_{\mathsf{cb}w^*} \leq \|\Phi\|_{\mathsf{op}}$. The inverse inequality follows from Proposition 5.5.14 and the fact that $\|\phi\| = \|\phi\|_{\mathsf{cb}}$ [108, Exercise 11.5 (iii)]. □

Proposition 5.5.17. *The W^*-completely bounded norm is submultiplicative with respect to composition for completely bounded normal maps between W^*-algebras. That is, given W^*-completely bounded normal maps Φ and Ψ between W^*-algebras, we have:*

$$\|\Phi \cdot \Psi\|_{cbw^*} \leq \|\Phi\|_{cbw^*} \, \|\Psi\|_{cbw^*} .$$

Proof. It follows directly from the submultiplicativity of the operator norm (Lemma 4.1.4), the definition of *supremum* as the least upper bound, the definition of W^* completely bounded norm,and the following property: for any two subsets A and B of only positive elements of an ordered field $\mathcal{F}$, let $AB = \{a \cdot b \mid a \in A, y \in B\}$, it holds that $\sup A \cdot \sup B = \sup AB$ [109, Chapter 2, Section 8-9]. Given the submultiplicativity of the operator norm, we have that each

$$a \in \{\|\mathrm{id}_{\mathscr{M}} \overline{\otimes}\, \Phi \cdot \Psi\|_{\mathsf{op}} \mid \mathscr{M} \text{ is a } W^* \text{ algebra}\}$$

has an upper bound

$$b \in \{\|\mathrm{id}_{\mathscr{M}_1} \overline{\otimes} \Phi\|_{\mathsf{op}} \quad \|\mathrm{id}_{\mathscr{M}_2} \overline{\otimes} \Psi\|_{\mathsf{op}} \quad \mid \mathscr{M}, \mathscr{M}_2 \text{ are } W^* \text{ algebras}\}.$$

So, it follows from the definition of *supremum* that,

$$\sup_{\mathscr{M}}\{\|\mathrm{id}_{\mathscr{M}} \overline{\otimes} \Phi \cdot \Psi\|_{\mathsf{op}}\} \leq \sup_{\mathscr{M}_1, \mathscr{M}_2} \{\|\mathrm{id}_{\mathscr{M}_1} \overline{\otimes} \Phi\|_{\mathsf{op}} \quad \|\mathrm{id}_{\mathscr{M}_2} \overline{\otimes} \Psi\|_{\mathsf{op}} \quad \}$$

Consequently, applying the property stated at the beginning of the proof, it holds that,

$$\|\Phi \cdot \Psi\|_{\mathsf{cb}w^*} \leq \|\Phi\|_{\mathsf{cb}w^*} \; \|\Psi\|_{\mathsf{cb}w^*} \, .$$

□

Proposition 5.5.18. *Given a W^* completely bounded normal map Φ between W^*-algebras, it holds that*

$$\|\Phi \bar{\otimes} \mathrm{id}\|_{cbw^*} \leq \|\Phi\|_{cbw^*} \quad \textit{and} \quad \|\mathrm{id} \bar{\otimes} \Phi\|_{cbw^*} \leq \|\Phi\|_{cbw^*} \, .$$

Proof. Given we just proved that the W^* completely bounded norm is submultiplicative with respect to composition for W^* completely bounded normal maps, it holds that

$$\|\Phi \bar{\otimes} \mathrm{id}\|_{\mathsf{cb}w^*} = \|\mathrm{sw} \cdot (\mathrm{id} \bar{\otimes} \Phi)\|_{\mathsf{cb}w^*} \leq \|\mathrm{sw}\|_{\mathsf{cb}w^*} \; \|\mathrm{id} \bar{\otimes} \Phi\|_{\mathsf{cb}w^*}$$

By Proposition 5.4.36 and Proposition 5.4.13, it follows that sw is a normal $*$-isomorphism, and, therefore, a completely positive normal map and an isometry (with respect to the operator norm). As a result, by Corollary 5.5.16 and Proposition 5.4.5, we obtain $\|\mathrm{sw}\|_{\mathsf{cb}w^*} = \|\mathrm{sw}\|_{\mathsf{op}} = 1$. Consequently,

$$\|\Phi \bar{\otimes} \mathrm{id}\|_{\mathsf{cb}w^*} \leq \|\mathrm{id} \bar{\otimes} \Phi\|_{\mathsf{cb}w^*}$$

At last, we need to prove that

$$\|\Phi \bar{\otimes} \mathrm{id}\|_{\mathsf{cb}w^*} \leq \|\Phi\|_{\mathsf{cb}w^*} \, ,$$

which follows direcly from the definition of the norm (Definition 5.5.12) and the fact that the associator, α, is an isometry with respect to the operator norm (Proposition 5.4.35).

□

Proposition 5.5.19. *Given W^* completely bounded normal maps $\Phi : \mathscr{M} \to \mathscr{N}$ and $\Psi : \mathscr{M} \to \mathscr{T}$ between W^*-algebras, it holds that*

$$\|\langle \Phi, \Psi \rangle\|_{cbw^*} \leq \max\{\|\Phi\|_{cbw^*} \, , \|\Psi\|_{cbw^*}\}.$$

Eu aqui só estou relutante em adotar a sua reformulação pelo seguinte motivo: nas provas anteriores

Proof. By Theorem 5.4.30, Theorem 5.4.40, and Definition 5.4.14 we have

$$
\begin{aligned}
&\|\langle\Phi,\Psi\rangle\|_{\mathsf{cb}w^*} = \|\mathrm{id}\,\overline{\otimes}\,\langle\Phi,\Psi\rangle\| \\
&= \|\mathrm{id}\odot\langle\Phi,\Psi\rangle\| && \text{(Thm. 5.4.30)} \\
&= \sup_{s_i}\left\{\left\|\mathrm{id}\odot\langle\Phi,\Psi\rangle\left(\sum_i s_i\otimes(m_i\otimes n_i)\right)\right\| \;\middle|\; \left\|\sum_i s_i\otimes(m_i,n_i)\right\| = 1\right\} \\
&= \sup_{s_i}\left\{\left\|\sum_i (s_i\otimes\Phi(m_i), s_i\otimes\Phi(n_i))\right\| \;\middle|\; \left\|\sum_i (s_i\otimes m_i, s_i\otimes n_i)\right\| = 1\right\} && \text{(Thm. 5.4.40)} \\
&= \sup_{\mathscr{S}}\left\{\max\{\|\mathrm{id}_{\mathscr{S}}\odot\Phi\|, \|\mathrm{id}_{\mathscr{S}}\odot\Psi\|\}\right\} && \text{(Def. 5.4.14)} \\
&= \max\{\sup_{\mathscr{S}}\mathrm{id}\,\overline{\otimes}\,\Phi, \sup_{\mathscr{S}}\mathrm{id}\,\overline{\otimes}\,\Psi\} && \text{(Thm. 5.4.30)} \\
&\leq \max\{\|\mathrm{id}\,\overline{\otimes}\,\Phi\|_{\mathsf{cb}w^*}, \|\mathrm{id}\,\overline{\otimes}\,\Psi\|_{\mathsf{cb}w^*}\}
\end{aligned}
$$

□

Theorem 5.5.20. $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ *is a distributive symmetric monoidal* Met*-category.*

Proof. Firstly, note that the copairing in $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ corresponds to the pairing in $\mathsf{W}^*_{\mathsf{CPSU}}$. As a result, by proof of Proposition 5.5.9 and the definition of symmetric monoidal Met-category, we need to prove that for any normal completely subunital maps Φ, Φ', Ψ, Ψ' between W^*-algebras:

1. $\|(\Phi-\Phi')\Psi\|_{\mathsf{cb}w^*} \leq \|\Phi-\Phi'\|_{\mathsf{cb}w^*}$ and $\|\Phi'(\Psi-\Psi')\|_{\mathsf{cb}w^*} \leq \|\Psi-\Psi'\|_{\mathsf{cb}w^*}$. Given these operadors are normal completely positive subunital and the W^* completely bounded norm is submultiplicative with respect to composition for such maps, by Corollary 5.5.16 and Proposition 5.4.6 the inequalities hold.

2. $\|\Phi\,\bar{\otimes}\,\mathrm{id}\|_{\mathsf{cb}w^*} \leq \|\Phi\|_{\mathsf{cb}w^*}$ and $\|\mathrm{id}\,\bar{\otimes}\,\Phi\|_{\mathsf{cb}w^*} \leq \|\Phi\|_{\mathsf{cb}w^*}$. It follows directly from Proposition 5.5.18.

3. $\|\langle\Phi-\Phi', \Psi-\Psi'\rangle\|_{\mathsf{cb}w^*} \leq \max\{\|\Phi-\Phi'\|_{\mathsf{cb}w^*}, \|\Psi-\Psi'\|_{\mathsf{cb}w^*}\}$. It follows directly from Proposition 5.5.19.

□

5.6 Examples

We now illustrate the use of (first-order) λ-calculus with conditionals for describing quantum programs. To this effect, we first consider a type qbit of qubits, the basic unit of information in quantum computation. We then regard $\mathbb{I} \oplus \mathbb{I}$ to be the type of bits. Next we propound the following basic quantum operations: the conversion of a bit into a qubit, $q : \mathbb{I} \oplus \mathbb{I} \to \texttt{qbit}$, the measurement of a qubit, *meas* $: \texttt{qbit} \to \mathbb{I} \oplus \mathbb{I}$, and pre-determined sets of operations on n-qubits, *U* $: \texttt{qbit}, \dots, \texttt{qbit} \to \texttt{qbit}^{\otimes n}$ and *CPTP* $: \texttt{qbit}, \dots, \texttt{qbit} \to \texttt{qbit}^{\otimes n}$. The former includes unitary operations, as the Hadamard gate $H : \texttt{qbit} \to \texttt{qbit}$, the not-gate $X : \texttt{qbit} \to \texttt{qbit}$, and the cnot-gate $CNOT : \texttt{qbit}, \texttt{qbit} \to \texttt{qbit}^{\otimes 2}$, and the latterr set included operations such as the dephasing with probability p, $D_p : \texttt{qbit}, \texttt{qbit} \to \texttt{qbit}^{\otimes 2}$. We consider as well a pre-determined set of quantum states $|\psi\rangle : \mathbb{I} \to \texttt{qbit}$.

Q forms a model of the metric λ-theory for quantum computation via the following interpretation: $\llbracket \mathbb{I} \rrbracket = \mathbb{C} \ni 1$, $\llbracket \texttt{qbit} \rrbracket = \mathbb{C}^2$, $\llbracket q \rrbracket ((a,b)) = \left(\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right)$, for $\psi \in \{0,1\}$ we define $\llbracket |\psi\rangle \rrbracket (1) = |\psi\rangle \langle\psi|$, $\llbracket \textit{meas} \rrbracket (\rho) = (\mathsf{Tr}(M_0 \rho M_0^\dagger), \mathsf{Tr}(M_1 \rho M_1^\dagger))$, for unitary operations U we define $\llbracket U \rrbracket = U \rho U^\dagger$. For completely positive trace-preserving operators *CPTP*, defined as *CPTP*$(\rho) = \sum_i K_i \rho K_i^\dagger$, we define $\llbracket \textit{CPTP} \rrbracket =$ *CPTP*(ρ).

Let us now apply this machinery to two well-known problems in quantum computation and quantum information.

5.6.1 Quantum state discrimination

Example 5.6.1 (Coin-Toss)**.** In the quantum setting, tossing a “fair” coin can be described as preparing a qubit in a superposition of two states, $|0\rangle$ and $|1\rangle$, representing ‘heads’ and ‘tails’, each with an equal probability of 0.5 and then measuring it. This is achieved by simply applying a Hadamard gate to the initial state $|0\rangle$, followed by a measurement. More generally, tossing a coin (whether “fair” or “unfair”) can be described as preparing a qubit in a superposition of $|0\rangle$ and $|1\rangle$, with probabilities p and $1-p$, respectively, and then measuring it. Considering $p = \cos(\theta/2)^2$ and the quantum gate $R_{y,\theta} : \texttt{qbit} \to \texttt{qbit}$, representing a single-qubit rotation by an angle θ around the y-axis, this process is described by the following λ-term:

$$\textbf{CoinToss} = - \triangleright \textit{meas}(R_{y,\theta}(|0\rangle)) : \mathbb{I} \oplus \mathbb{I}$$

When running a quantum program on a real quantum computer, the quantum circuits are mapped to the hardware's native quantum gates during compilation. For instance consider 2020 IBM's native quantum gate set U_1, U_2, U_3, CX where

$$\begin{aligned}
U_1(\lambda) &= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix} \\
U_2(\phi, \lambda) &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{pmatrix} \\
U_3(\theta, \phi, \lambda) &= \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{pmatrix}
\end{aligned}$$

Here, $R_{y,\theta}$, can be expressed as $U_3(\theta, 0, 0)$. We now examine how the coin toss outcome is affected when the U_3 gate is faulty, particularly when its parameter θ is perturbed by an error ϵ. In this case, the implemented gate becomes $U_3(\theta + \epsilon, \phi, \lambda)$, *i.e.*, $R_{y,\theta+\epsilon}$. First, we compute the action of the unitary operator $U_3(\theta, \phi, \lambda)$ on an arbitrary quantum state $|\psi\rangle$.

$$\begin{aligned}
U_3(\theta, \phi, \lambda)|\psi\rangle &= U_3(\theta, \phi, \lambda)\big(\cos(\alpha/2)|0\rangle + e^{i\beta}\sin(\alpha/2)|1\rangle\big) \\
&= \big(\cos(\alpha/2)\cos(\theta/2) - e^{i(\lambda+\beta)}\sin(\alpha/2)\sin(\theta/2)\big)|0\rangle \\
&+ \big(e^{i\phi}\cos(\alpha/2)\sin(\theta/2) + e^{i(\beta+\lambda+\phi)}\sin(\alpha/2)\cos(\theta/2)\big)|1\rangle
\end{aligned}$$

Designating $U_3(\theta, \phi, \lambda)|\psi\rangle = a|0\rangle + b|1\rangle$, one has

$$\begin{aligned}
aa^* &= |\cos(\alpha/2)\cos(\theta/2) - e^{i(\lambda+\beta)}\sin(\alpha/2)\sin(\theta/2)|^2 \\
&= \cos^2(\alpha/2)\cos^2(\theta/2) - 2\cos(\beta+\lambda)\cos(\alpha/2)\cos(\theta/2)\sin(\alpha/2)\sin(\theta/2) \\
&\quad + \sin(\alpha/2)^2\sin^2(\theta/2) \\
&= \cos^2(\alpha/2)\cos^2(\theta/2) + \sin^2(\alpha/2)\sin^2(\theta/2) - 1/2\cos(\beta+\lambda)\sin(\alpha)\sin(\theta) \\
&= \cos^2((\theta+\alpha)/2) + (1-\cos(\beta+\lambda))\sin(\alpha)\sin(\theta) - 1 \\
a^*b &= \big(\cos(\alpha/2)\cos(\theta/2) - e^{-i(\lambda+\beta)}\sin(\alpha/2)\sin(\theta/2)\big)\big(e^{i\phi}\cos(\alpha/2)\sin(\theta/2) \\
&\quad + e^{i(\beta+\lambda+\phi)}\sin(\alpha/2)\cos(\theta/2)\big) \\
&= (1/2)\big(e^{i\phi}\cos^2(\alpha/2)\sin(\theta) + e^{i(\beta+\lambda+\phi)}\sin(\alpha)\cos^2(\theta/2) - e^{-i(\beta+\lambda-\phi)}\sin(\alpha)\sin^2(\theta/2) \\
&\quad - e^{i\phi}\sin^2(\alpha/2)\cos(\theta/2)\big) \\
&= (1/2)\left(e^{i\phi}\cos(\alpha)\sin(\theta) + \sin(\alpha)\left(e^{i(\beta+\lambda+\phi)}\cos^2(\theta/2) - e^{-i(\beta+\lambda-\phi)}\sin^2(\theta/2)\right)\right)
\end{aligned}$$

Then, we calculate the vector Bloch of $U_3(\theta,\phi,\lambda)\left|\psi\right\rangle$,

$$\begin{aligned}
x &= 2\mathsf{Im}\,(a^*b) = \cos(\phi)\cos(\alpha)\sin(\theta) + \sin(\alpha)\big(\cos\left(\beta+\lambda+\phi\right)\cos^2(\theta/2) \\
&\quad - \cos\left(\beta+\lambda-\phi\right)\sin^2(\theta/2)\big) \\
y &= 2\mathsf{Re}\,(a^*b) = \sin(\phi)\cos(\alpha)\sin(\theta) + \sin(\alpha)\big(\sin\left(\beta+\lambda+\phi\right)\cos^2(\theta/2) \\
&\quad + \sin\left(\beta+\lambda-\phi\right)\sin^2(\theta/2)\big) \\
z &= 2aa^* - 1 = 2\cos^2((\theta+\alpha)/2) + (1-\cos(\beta+\lambda))\sin(\alpha)\sin(\theta) - 1
\end{aligned}$$

As a result, we have,

$$\begin{aligned}
&\left\| U_3(\theta,0,0)\left|\psi\right\rangle\left\langle\psi\right| U_3(\theta,0,0)^\dagger - U_3(\theta+\epsilon,0,0)\left|\psi\right\rangle\left\langle\psi\right| U_3(\theta+\epsilon,0,0)^\dagger \right\|_1 \\
&= \|(\cos(\alpha)(\sin(\theta)-\sin(\theta+\epsilon)) + \sin(\alpha)\cos\beta\left(\cos(\theta)-\cos(\theta+\epsilon)\right), 0, \\
&\quad 2(\cos^2((\theta+\alpha)/2) - \cos^2((\theta+\epsilon+\alpha)/2)) + \cos(\beta)\sin(\alpha)(\sin(\theta)-\sin(\theta+\epsilon)))\|_2 \\
&\leq \|(\cos(\alpha)\epsilon + \sin(\alpha)\cos(\beta)\epsilon, 0, 2\epsilon + \sin(\alpha)\cos(\beta)\epsilon)\|_2 \\
&= \sqrt{\epsilon^2(\cos^2(\alpha) + 2\sin^2(\alpha)\cos^2(\beta) + \sin(2\alpha)\cos(\beta) + 2\sin(\alpha)\cos(\beta) + 4)} \\
&\leq \epsilon\sqrt{(1+\sin^2(\alpha) + \sin(2\alpha) + 2\sin(\alpha) + 4)} \\
&\leq \epsilon\sqrt{(1+4+4)} = 3\epsilon
\end{aligned}$$

The first inequality arises from both functions $\cos$ and $\sin$ being Lipschitz continuous. Attending to Theorem 5.2.33, it follows that

$$\|R_{y,\theta} - R_{y,\theta+\epsilon}\|_\diamond \leq 3\epsilon$$

Using our metric deductive system, we can easily conclude that **CoinToss** $=_{3\epsilon}$ **CoinToss**$^\epsilon$, where **CoinToss**$^\epsilon$ is the the judgement that results from replacing $R_{y,\theta}$ by $U^{R_{y,\theta+\epsilon}}$.

Example 5.6.2 (Quantum state discrimination)**.** Quantum state discrimination is a pivotal challenge in quantum theory of communications [70, 110] and quantum cryptography [111]. While orthogonal states can be perfectly distinguished, the same does not apply to nonorthogonal states. In fact, even when the set of possible nonorthogonal states is known, determining the optimal discrimination strategy is considered a nontrivial problem.

The problem of quantum state discrimination can be naturally introduced through its connection with quantum communication. Consider two parties, Alice and Bob, who want to communicate with each other using a quantum channel. Alice chooses a state from a known set $\{|\psi_i\rangle\}$, each occurring with a known probability p_i and sends it to Bob through the channel. Bob, who knows both the set of possible states and their associated probabilities, performs a suitable measurement to determine which state Alice sent. This scenario defines the quantum state discrimination problem: how to optimally distinguish between a known set of quantum states, each prepared with a known prior probability p_i.

When distinguishing between two pure states, the optimal measurement known as the *Helstrom measurement* is given by a projective measurement [110]. When operating within the computational basis, a projective measurement can be understood as the application of a unitary operator followed by a subsequent measurement in the computational basis. Thus the optimal measurement can be interpreted as a unitary transformation applied to the quantum state, followed by a measurement in the computational basis.

We will now show how to describe this discrimination task in λ-calculus. Consider two pure states $|\psi_0\rangle$ and $|\psi_1\rangle$, prepared *a priori* with probabilities p_0 and $p_1 = 1 - p_0$, respectively. Consider as well an operation $U : \texttt{qbit} \to \texttt{qbit}$ which corresponds to the basis-change associated with the optimal measurement. The relevant λ-terms are then:

$$\textbf{StatePreparation} = b : \mathbb{I} \oplus \mathbb{I} \triangleright \text{case } b \, \{\text{inl}_{\mathbb{B}}(x) \Rightarrow |\psi_0\rangle \, ; \text{inr}_{\mathbb{A}}(y) \Rightarrow |\psi_1\rangle\} : \texttt{qbit}$$

$$\textbf{HMeasure} = x : \texttt{qbit} \triangleright \textit{meas}(\textit{U}(x)) : \mathbb{I} \oplus \mathbb{I}$$

$$\textbf{Discrimination} = - \triangleright \textbf{HMeasure}[\textbf{StatePreparation}[\textbf{CoinToss}(*)/b]/x] : \mathbb{I} \oplus \mathbb{I}$$

An arbitrary single qubit unitary $U \in \mathbb{C}^{2\times 2}$ may be written

$$U = e^{i\alpha} R_{z,\beta} R_{y,\gamma} R_{z,\delta},$$

for appropriate choices of angles α, β, γ and δ.

As in the previous example, we assume the hardware's native gate set consists of $\{U_1, U_2, U_3,$ CNOT$\}$, and the quantum circuit is compiled into these gates. As previously noted, the $R_y(\theta)$ gate can be implemented as $U_3(\theta, 0, 0)$. Similarly, the $R_z(\lambda)$ gate is equivalent to $U_1(\lambda)$ up to a global phase factor $e^{-i\lambda/2}$, consequently, it can be directly implemented using this gate. We will also consider that the gates U_1 and U_3 are affected by errors ϵ_1 and ϵ_2, respectively. More precisely, we will consider erroneous implementations of this gates $U_1(\lambda+\epsilon_1)$ and $U_3(\theta+\epsilon_2, \phi, \lambda)$. From the previous example, we know that the error in the R_y gate is bounded by $3\epsilon_2$. Consider the single-qubit state

$$|\psi\rangle = \cos\left(\frac{\alpha}{2}\right)|0\rangle + e^{i\beta}\sin\left(\frac{\alpha}{2}\right)|1\rangle\,.$$

Applying the $U_1(\lambda)$ gate yields

$$U_1(\lambda)\left|\psi\right\rangle = \cos\left(\frac{\alpha}{2}\right)\left|0\right\rangle + e^{i(\beta+\lambda)}\sin\left(\frac{\alpha}{2}\right)\left|1\right\rangle .$$

The corresponding Bloch vector is then

$$(\cos(\beta+\lambda)\sin\alpha,\ \sin(\beta+\lambda)\sin\alpha,\ \cos\alpha)\,.$$

Consequently, applying the same reasoning as in the previous example, it follows that

$$\begin{aligned}
&\left\|U_1(\lambda)\left|\psi\right\rangle\left\langle\psi\right|U_1(\lambda)^\dagger - U_1(\lambda+\epsilon)\left|\psi\right\rangle\left\langle\psi\right|U_1(\lambda+\epsilon)^\dagger\right\|_1\\
&= \left\|\left(\sin(\alpha)\left(\cos(\beta+\lambda)-\cos(\beta+\lambda+\epsilon)\right),\sin(\alpha)\left(\sin(\beta+\lambda)-\sin(\beta+\lambda+\epsilon)\right),0\right)\right\|_2\\
&\leq \left\|(\cos(\beta+\lambda)-\cos(\beta+\lambda+\epsilon),\sin(\beta+\lambda)-\sin(\beta+\lambda+\epsilon),0)\right\|_2\\
&\leq \left\|(\epsilon,\epsilon,0)\right\|_2\\
&= \sqrt{2}\epsilon
\end{aligned}$$

Attending to Theorem 5.2.33, it follows that

$$\left\|U_1(\lambda)-U_1(\lambda+\epsilon)\right\|_\diamond \leq \sqrt{2}\epsilon$$

As a result, considering the λ-term **HMeasure** and the erroneous implementation of U described above, denoted $U^{\epsilon_1,\epsilon_2}$, using our deductive metric system, we have $U =_{\sqrt{2}\epsilon_1+3\epsilon_2} U^{\epsilon_1,\epsilon_2}$. Therefore, we deduce **HMeasure** $=_{\sqrt{2}\epsilon_1+3\epsilon_2}$ **HMeasure**$^{\epsilon_1,\epsilon_2}$, where **HMeasure**$^{\epsilon_1,\epsilon_2}$ is the the judgement that results from replacing U by $U^{\epsilon_1,\epsilon_2}$. Moreover, considering the erroneous implementation of the R_y gate also afecting the **CoinToss** term, as discussed in the previous example, we deduce that

$$\textbf{Discrimination} =_{\sqrt{2}\epsilon_1+6\epsilon_2} \textbf{Discrimination}^{\epsilon_1,\epsilon_2},$$

where **Discrimination**$^{\epsilon_1,\epsilon_2}$ denotes the judgement that results from replacing **HMeasure** by **HMeasure**$^{\epsilon_1,\epsilon_2}$ and **CoinToss** by **CoinToss**$^{\epsilon_2}$.

5.6.2 Quantum teleportation protocol

Example 5.6.3 (Quantum teleportation)**.** [112] introduced the concept of quantum teleportation, a protocol that allows the transfer of unknown quantum states between distant parties. The quantum teleportation protocol is a fundamental building block of quantum communication, quantum computation, and quantum networks, its applications ranging from secure quantum communication to distributed quantum computing [113–115].

Conceptually it can be described as follows: Alice and Bob share an entangled pair of qubits, specifically in a Bell state. Alice keeps the first qubit and Bob the second. Moreover, Alice has a qubit in an unknown state $|\psi\rangle$ that she wants to send to Bob. Alice entangles her qubit and the first qubit in the Bell state, and then measures both. The result of this measurement is two classical bits that Alice then sends to Bob though a classical channel. Based on the measurement results, Bob applies a correction to his qubit so it matches the initial state $|\psi\rangle$. The circuit corresponding to the implementation of quantum teleportation is depicted in Figure 7.

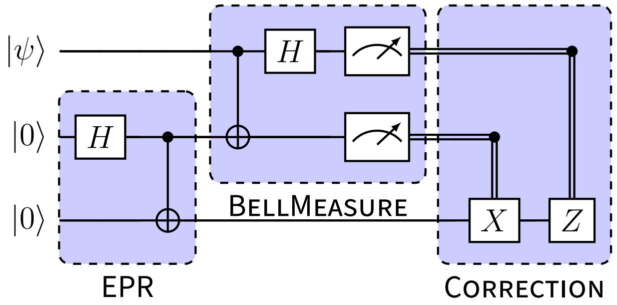


Figure 7: Quantum Teleportation Protocol

We first describe each of the rectangles filled in blue separately, and using standard quantum gate operations, namely $H : \texttt{qbit} \to \texttt{qbit}$, $X : \texttt{qbit} \to \texttt{qbit}$, $Z : \texttt{qbit} \to \texttt{qbit}$, and $\mathit{CNOT} : \texttt{qbit}, \texttt{qbit} \to \texttt{qbit} \otimes \texttt{qbit}$:

$$\textbf{EPR} = \mathit{CNOT}(H\,|0\rangle\,, |0\rangle) : \texttt{qbit} \otimes \texttt{qbit}$$

$$\textbf{BellMeasure} = q_1 : \texttt{qbit}, q_2 : \texttt{qbit} \triangleright \text{pm}\ \mathit{CNOT}(q_1, q_2)\ \text{to}\ x \otimes y.$$

$$\mathit{meas}(H(x)) \otimes \mathit{meas}(y) : (\mathbb{I} \oplus \mathbb{I}) \otimes (\mathbb{I} \oplus \mathbb{I})$$

$$\textbf{Correction} = q : \texttt{qbit}, x : \mathbb{I} \oplus \mathbb{I}, y : \mathbb{I} \oplus \mathbb{I} \triangleright$$

$$\text{case}\ x \left\{ \begin{array}{l} \mathrm{inl}(x_0) \Rightarrow x_0\ \text{to}\ * .\text{case}\ y \left\{ \begin{array}{l} \mathrm{inl}(y_0) \Rightarrow y_0\ \text{to}\ * . q; \\ \mathrm{inr}(y_1) \Rightarrow y_1\ \text{to}\ * . X \end{array} \right\} ; \\ \mathrm{inr}(x_1) \Rightarrow x_1\ \text{to}\ * .\text{case}\ y \left\{ \begin{array}{l} \mathrm{inl}(y_0) \Rightarrow y_0\ \text{to}\ * . Z(q); \\ \mathrm{inr}(y_1) \Rightarrow y_1\ \text{to}\ * . Z(X(q)) \end{array} \right\} \end{array} \right\} : \texttt{qbit}$$

Designating the qubit to be teleported as qb_0, one then describes the teleportation procedure in λ-calculus as follows:

$$\begin{aligned}\textbf{QTP} = qb_0 : \texttt{qbit} \triangleright \text{pm } &\textbf{EPR} \text{ to } qb_1 \otimes qb_2.\\ &\text{pm } \textbf{BellMeasure}\,[qb_0/q_1, qb_1/q_2] \text{ to } c_0 \otimes c_1.\\ &\textbf{Correction}\,[qb_2/q, c_0/x, c_1/y] : \texttt{qbit}\end{aligned}$$

Following the approach of previous examples, we analyze erroneous implementations of the gates U_1 and U_3 within the hardware's native gate set. Additionally, we consider the action of both dephasing and amplitude damping channels. Furthermore, we account for an adversarial agent that applies a bit-flip operation immediately prior to measurement with probability $p = 0.5$.

Here, we consider imperfect implementations of the gates U_1 and U_3, given by $U_1(\lambda)$ and $U_3(\theta, \phi + \epsilon_2, \lambda + \epsilon_3)$, respectively. Recall from the previous example that we established the bound $\|U_1(\lambda) - U_1(\lambda + \epsilon)\|_\diamond \leq \sqrt{2}\epsilon$. The Hadamard gate, H, is the composition $U_3(\pi/2, 0, 0)\cdot U_1(\pi)$. We calculate,

$$\begin{aligned}&\left\|U_3(\pi/2,0,0)\,|\psi\rangle\,\langle\psi|\,U_3(\pi/2,0,0)^\dagger - U_3(\pi/2,\epsilon_2,\epsilon_3)\,|\psi\rangle\,\langle\psi|\,U_3(\pi/2,\epsilon_2,\epsilon_3)^\dagger\right\|_1\\ &= \|(\cos(\alpha)(1-\cos\epsilon_2) + \sin(\alpha)(1/2(\cos(\beta+\epsilon_2+\epsilon_3) - \cos(\beta-\epsilon_2+\epsilon_3))),\\ &\qquad \cos(\alpha)(1-\sin\epsilon_2) + (1/2)\sin(\alpha)(\sin(\beta) - \sin(\beta+\epsilon_2+\epsilon_3)\\ &\qquad + \sin(\beta) - \sin(\beta-\epsilon_2+\epsilon_3)), (\cos(\beta+\epsilon_3) - \cos(\beta))\sin(\alpha))\|_2\\ &\leq ||(\cos(\alpha)\epsilon_2 + (1/2)\sin(\alpha)(\epsilon_2+\epsilon_3), \cos(\alpha)\epsilon_2 + (1/2)\sin(\alpha)(\epsilon_2+\epsilon_3+|\epsilon_3-\epsilon_2|),\\ &\qquad \sin(\alpha)\epsilon_3)||_2\\ &< \|(\epsilon_2 + (1/2)(\epsilon_2+\epsilon_3), \epsilon_2 + (1/2)(\epsilon_2+\epsilon_3+|\epsilon_3-\epsilon_2|), \epsilon_3)\|_2\\ &\leq 3\epsilon_2 + 2\epsilon_3 + |\epsilon_2 - \epsilon_3|\end{aligned}$$

Attending to Theorem 5.2.33, it follows that

$$\|U_3(\pi/2,0,0) - U_3(\pi/2,\epsilon_2,\epsilon_3)\|_\diamond \leq 3\epsilon_2 + 2\epsilon_3 + |\epsilon_2 - \epsilon_3|$$

As a result, denoting the imperfect implementation of the Hadamard gate as $H^{\epsilon_1,\epsilon_2,\epsilon_3}$, we have

$$H =_{\sqrt{2}\epsilon_1 + 3\epsilon_2 + 2\epsilon_3 + |\epsilon_2 - \epsilon_3|} H^{\epsilon_1,\epsilon_2,\epsilon_3}. \tag{5.6}$$

The gate X can be implemented as $U_3(\pi, \psi, 0)$. We compute,

$$
\begin{aligned}
&\left\| U_3(\pi, 0, \pi) \left|\psi\right\rangle \left\langle\psi\right| U_3(\pi, 0, \pi)^\dagger - U_3(\pi, \epsilon_2, \pi + \epsilon_3) \left|\psi\right\rangle \left\langle\psi\right| U_3(\pi, \epsilon_2, \pi + \epsilon_3)^\dagger \right\|_1 \\
&= \|(\sin(\alpha)(\cos(\beta) + \cos(\beta + \epsilon_3 - \epsilon_2)), \sin(\alpha)(\sin(\beta + \epsilon_3 - \epsilon_2) - \sin(\beta)), 0)\|_2 \\
&\leq \|(|\epsilon_3 - \epsilon_2|, |\epsilon_3 - \epsilon_2|, 0)\|_2 \\
&= \sqrt{2}|\epsilon_3 - \epsilon_2|
\end{aligned}
$$

Considering Theorem 5.2.33, it holds that

$$\|U_3(\pi, 0, \pi) - U_3(\pi, \epsilon_2, \pi + \epsilon_3)\|_\diamond \leq \sqrt{2}|\epsilon_3 - \epsilon_2|$$

As a result, denoting the erroneous implementation of the X gate as $X^{\epsilon_2, \epsilon_3}$, we have

$$X =_{\sqrt{2}|\epsilon_3 - \epsilon_2|} X^{\epsilon_2, \epsilon_3}. \tag{5.7}$$

Finally, the gate Z corresponds to $U_1(\pi)$, therefore, denoting the erroneous implementation of the X gate as X^{ϵ_1}, we postulate the following axiom

$$Z =_{\sqrt{2}\epsilon_1} Z^{\epsilon_1}. \tag{5.8}$$

Designating the **Correction** block with the imperfect implementations of X and Z by **Correction**$^{\epsilon_1, \epsilon_2, \epsilon_3}$, in light of the axioms in equations (5.7) and (5.7) and our metric deductive system we have that

$$\textbf{Correction} =_{\sqrt{2}(\epsilon_1 + |\epsilon_3 - \epsilon_2|)} \textbf{Correction}^{\epsilon_1, \epsilon_2, \epsilon_3}. \tag{5.9}$$

Dephasing channel

Realistic quantum systems are never isolated, but are immersed in the surrounding environment and interact continuously with it. 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:

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

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

$$\rho' = D_p(\rho) = D_0 \rho D_0^\dagger + D_1 \rho D_1^\dagger = |\alpha|^2 \left|0\right\rangle \left\langle 0\right| + (1-p)\alpha\overline{\beta} \left|0\right\rangle \left\langle 1\right| + (1-p)\overline{\alpha}\beta \left|1\right\rangle \left\langle 0\right| + |\beta|^2 \left|1\right\rangle \left\langle 1\right|$$

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 $(1-p)$.

In this scenario (and in subsequent ones), we will add identity gates to the ideal program to simplify the calculations. Thus, attending to the definition of trace norm for matrices and Equation 5.1, we have:

$$
\begin{aligned}
&\|\mathrm{id}(\rho) - D_p(\rho)\|_1 \\
&\left\|\alpha\overline{\beta}\,|0\rangle\langle 1| + \overline{\alpha}\beta\,|1\rangle\langle 0| - (1-p)\alpha\overline{\beta}\,|0\rangle\langle 1| - (1-p)\overline{\alpha}\beta\,|1\rangle\langle 0|\right\|_1 \\
&= p \cdot \left\|\alpha\overline{\beta}\,|0\rangle\langle 1| + \overline{\alpha}\beta\,|1\rangle\langle 0|\right\|_1 \\
&= p \cdot \mathrm{Tr}\left(\sqrt{\left(\alpha\overline{\beta}\,|0\rangle\langle 1| + \overline{\alpha}\beta\,|1\rangle\langle 0|\right)^2}\right) \\
&= p \cdot \mathrm{Tr}\left(\sqrt{|\alpha|^2|\beta|^2(|0\rangle\langle 0| + |1\rangle\langle 1|)}\right) \\
&= 2 \cdot p \cdot |\alpha||\beta| \\
&\leq p
\end{aligned}
$$

The last step arises from the fact that the expression is maximized when $|\alpha| = |\beta| = 1/\sqrt{2}$. Considering Theorem 5.2.34, it holds that

$$\|\mathrm{id} - D_p\|_\diamond \leq \sqrt{2p}$$

Consequently, we can postulate the following axiom:

$$\mathrm{id} =_{\sqrt{2p}} D_p. \tag{5.10}$$

If a dephasing channel acts on the first qubit of the EPR state, we are interested in reasoning about the following judgements:

$$
\begin{aligned}
&\textbf{EPR} = (\mathrm{id} \otimes \mathrm{id})(\mathit{CNOT}(H\,|0\rangle\,, |0\rangle)) : \mathtt{qbit} \otimes \mathtt{qbit} \\
&\textbf{EPR}^{\epsilon_1,\epsilon_2,\epsilon_3,p} = (D_p \otimes \mathrm{id})(\mathit{CNOT}(H^{\epsilon_1,\epsilon_2,\epsilon_3}\,|0\rangle\,, |0\rangle)) : \mathtt{qbit} \otimes \mathtt{qbit}
\end{aligned}
$$

Given axioms in equations (5.6) and (5.10), using our metric deductive system, we infer that

$$\textbf{EPR} =_{\sqrt{2\epsilon_1+3\epsilon_2+2\epsilon_3+|\epsilon_2-\epsilon_3|}+\sqrt{2p}} \textbf{EPR}^{\epsilon_1,\epsilon_2,\epsilon_3,p} \tag{5.11}$$

Amplitude Dephasing channel

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 [116, 117].
The amplitude damping channel with probability γ is described by the Kraus operators:

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

Applying these Kraus operators an arbitray density operator $\rho = |\alpha|^2\,|0\rangle\,|0\rangle + \alpha\overline{\beta}\,|0\rangle\,|1\rangle + \overline{\alpha}\beta\,|1\rangle\,|0\rangle + |\beta|^2\,|1\rangle\,|1\rangle$, we obtain the state ρ' as follows:

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

Once again, we will add identity gates to the ideal program to simplify the calculations, as a result it is necessary to compute the trace norm of the diference between the identity applied to the density operator $\rho = |\psi\rangle\langle\psi|$ and the amplitude damping channel applied to ρ. First, attending to the definition of trace norm for matrices and Equation 5.1, we calculate,

$$\begin{aligned}&\|\mathrm{id}(\rho) - A_\gamma(\rho)\|_1\\ &= \Big\||\alpha|^2\,|0\rangle\,|0\rangle + \alpha\overline{\beta}\,|0\rangle\,|1\rangle + \overline{\alpha}\beta\,|1\rangle\,|0\rangle + |\beta|^2\,|1\rangle\,|1\rangle - \big((|\alpha|^2 + \gamma|\beta|^2)\,|0\rangle\langle 0|\\ &\quad + \sqrt{1-\gamma}\alpha\overline{\beta}\,|0\rangle\langle 1| + \sqrt{1-\gamma}\overline{\alpha}\beta\,|1\rangle\langle 0| + (1-\gamma)|\beta|^2\,|1\rangle\langle 1|\big)\Big\|_1\\ &= \Big\|\gamma|\beta|^2\,|0\rangle\langle 0| + (1-\sqrt{1-\gamma})(\alpha\overline{\beta}\,|0\rangle\langle 1| + \overline{\alpha}\beta\,|1\rangle\langle 0|) - \gamma|\beta|^2\,|1\rangle\langle 1|\Big\|_1\\ &= \text{Tr}\left(\sqrt{\left(\gamma|\beta|^2\,|0\rangle\langle 0| + (1-\sqrt{1-\gamma})(\alpha\overline{\beta}\,|0\rangle\langle 1| + \overline{\alpha}\beta\,|1\rangle\langle 0|) - \gamma|\beta|^2\,|1\rangle\langle 1|\right)^2}\right)\\ &= \text{Tr}\left(\sqrt{\left((1-\sqrt{1-\gamma})^2|\alpha|^2|\beta|^2 + \gamma^2|\beta|^4\right)(|0\rangle\langle 0| + |1\rangle\langle 1|)}\right)\\ &= 2\cdot\sqrt{(1-\sqrt{1-\gamma})^2|\alpha|^2|\beta|^2 + \gamma^2|\beta|^4}\\ &\le 2\gamma\end{aligned}$$

This final step follows because the expression attains its maximum when $|\alpha| = 0$ and $|\beta| = 0$. Attending to Theorem 5.2.34, it holds that

$$\|\mathrm{id} - A_\gamma\|_\diamond \le 2\sqrt{\gamma}$$

As a result, we can postulate the following axiom:

$$\mathrm{id} =_{2\sqrt{\gamma}} A_\gamma. \tag{5.12}$$

When an amplitude damping channel acts on the final qubit following the Correction block, we define two new lambda terms consisting of the ideal operation **Id** and its erroneous counterpart **Id**$^\gamma$.

$$\textbf{Id} = qb : \texttt{qbit} \triangleright \mathrm{id}(qb) \tag{5.13}$$

$$\textbf{Id}^\gamma = A_\gamma(q) : \texttt{qbit} \triangleright A_\gamma(qb) \tag{5.14}$$

Consequently the ideal version of teleportation protocol is now defined as follows

$$\begin{aligned} \textbf{QTP} = qb_0 : \texttt{qbit} \triangleright\ & \text{pm } \textbf{EPR} \text{ to } qb_1 \otimes qb_2. \\ & \text{pm } \textbf{BellMeasure}\,[qb_0/q_1, qb_1/q_2] \text{ to } c_0 \otimes c_1. \\ & \textbf{Id}\,[\textbf{Correction}/qb]\,[qb_2/q, c_0/x, c_1/y] : \texttt{qbit} \end{aligned}$$

Considering the axiom in equation (5.12) and our metric deductive system, it holds that

$$\textbf{Id} =_{2\sqrt{\gamma}} \textbf{Id}^\gamma$$

Malicious attack

Finally, 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:

$$\begin{aligned} &T : \ \texttt{qbit} \to \texttt{qbit} \\ &T = q : \texttt{qbit} \triangleright \text{pm}\, CU(R_{x,\frac{\pi}{2}}(|0\rangle), q) \text{ to } newq \otimes qb.\, \textit{disc}(newq) \end{aligned}$$

Here, CU denotes the controlled operation that applies U to the second qubit when the first qubit is in the state $|1\rangle\langle 1|$, and leaves it unchanged when the first qubit is in the state $|0\rangle\langle 0|$. The operator $R_{x,\frac{\pi}{2}}$ represents a rotation by $\frac{\pi}{2}$ around the x-axis of the Bloch sphere.
This operation is depicted in Figure 8.

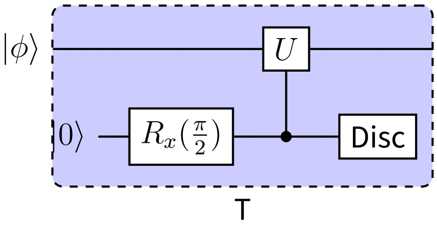


Figure 8: T operation

First, let us verify the result of applying operation T to a quantum state $\rho = |\psi\rangle\langle\psi|$:

$$
\begin{aligned}
& |\psi\rangle\langle\psi| \\
\xmapsto{\mathrm{id}\otimes\llbracket|0\rangle\rrbracket} \quad & |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0| \\
\xmapsto{\mathrm{id}\otimes\left\llbracket R_{x,\frac{\pi}{2}}\right\rrbracket} \quad & |\psi\rangle\langle\psi| \otimes \frac{1}{2}\left(|0\rangle\langle 0| - i\,|0\rangle\langle 1| + i\,|1\rangle\langle 0| + |1\rangle\langle 1|\right) \\
& = \frac{1}{2}\left(|\psi\rangle\langle\psi|\,|0\rangle\langle 0| - i\,|\psi\rangle\langle\psi|\,|0\rangle\langle 1| + i\,|\psi\rangle\langle\psi|\,|1\rangle\langle 0| + |\psi\rangle\langle\psi|\,|1\rangle\langle 1|\right) \\
\xmapsto{\llbracket\mathsf{CU}\rrbracket} \quad & \frac{1}{2}\left(|\psi\rangle\langle\psi|\,|0\rangle\langle 0| - i\,|\psi\rangle\langle\psi|\,|0\rangle\langle 1|\,U^{\dagger} + iU\,|\psi\rangle\langle\psi|\,|1\rangle\langle 0| + U\,|\psi\rangle\langle\psi|\,|1\rangle\langle 1|\,U^{\dagger}\right) \\
\xmapsto{\mathrm{id}\otimes\mathsf{Tr}} \quad & \frac{1}{2}\left(|\psi\rangle\langle\psi| + U\,|\psi\rangle\langle\psi|\,U^{\dagger}\right)
\end{aligned}
$$

Considering X as U, we compute

$$
\begin{aligned}
& \|\mathrm{id}(\rho) - T(\rho)\|_1 \\
& = \Big\|(1/2)\Big((|\alpha|^2 - |\beta|^2)\,|0\rangle\langle 0| + (\alpha\overline{\beta} - \overline{\alpha}\beta)\,|0\rangle\langle 1| + (\overline{\alpha}\beta - \alpha\overline{\beta})\,|0\rangle\langle 1| \\
& \quad + (|\beta|^2 - |\alpha|^2)\,|1\rangle\langle 1|\Big)\Big\|_1 \\
& = (1/2)\mathsf{Tr}\Big(\sqrt{\big((|\alpha|^2 - |\beta|^2)\,|0\rangle\langle 0| + (\alpha\overline{\beta} - \overline{\alpha}\beta)\,|0\rangle\langle 1| + (\overline{\alpha}\beta - \alpha\overline{\beta})\,|0\rangle\langle 1| + (|\beta|^2 - |\alpha|^2)\,|1\rangle\langle 1|\big)^2}\Big) \\
& = (1/2)\mathsf{Tr}\left(\sqrt{(((|\alpha|^2 - |\beta|^2)^2 + (\alpha\overline{\beta} - \overline{\alpha}\beta)(\overline{\alpha}\beta - \alpha\overline{\beta}))(|0\rangle\langle 0| + |1\rangle\langle 1|))}\right) \\
& = (1/2)\mathsf{Tr}\left(\sqrt{(((|\alpha|^2 - |\beta|^2)^2 + 2|\alpha|^2|\beta|^2 - (\overline{\alpha})^2\beta^2 - (\overline{\beta})^2\alpha^2)(|0\rangle\langle 0| + |1\rangle\langle 1|))}\right) \\
& = (1/2)\mathsf{Tr}\left(\sqrt{((|\alpha|^4 + |\beta|^4 - 2\mathsf{Re}(\alpha\beta))(|0\rangle\langle 0| + |1\rangle\langle 1|))}\right) \\
& = \sqrt{|\alpha|^4 + |\beta|^4 - 2\mathsf{Re}(\alpha\beta)} \\
& \leq 1
\end{aligned}
$$

This last step holds because the expression is maximized when the imaginary part of α or β is 1.

Considering Theorem 5.2.34, it holds that

$$\|\mathrm{id} - T\|_{\diamond} \leq \sqrt{2}$$

Consequently, we postulate the following axiom:

$$\mathrm{id} =_{\sqrt{2}} T. \tag{5.15}$$

In this case we reason about the following λ-terms:

$$\begin{aligned}
\textbf{BellMeasure} &= q_1 : \texttt{qbit}, q_2 : \texttt{qbit} \triangleright \text{pm } \textit{CNOT}(q_1, q_2) \text{ to } x \otimes y. \\
&\qquad\qquad \textit{meas}(\textit{H}(x)) \otimes \textit{meas}(y) : (\mathbb{I} \oplus \mathbb{I}) \otimes (\mathbb{I} \oplus \mathbb{I}) \\
\textbf{BellMeasure}^{\epsilon_1,\epsilon_2,\epsilon_3,T} &= q_1 : \texttt{qbit}, q_2 : \texttt{qbit} \triangleright \text{pm } \textit{CNOT}(q_1, q_2) \text{ to } x \otimes y. \\
&\qquad \textit{meas}(T(\textit{H}^{\epsilon_1,\epsilon_2,\epsilon_3}(x))) \otimes \textit{meas}(T(y)) : (\mathbb{I} \oplus \mathbb{I}) \otimes (\mathbb{I} \oplus \mathbb{I})
\end{aligned}$$

Attending to the axioms in equations (5.6) and (5.15), via our metric deductive system, we infer that

$$\textbf{BellMeasure} =_{2\sqrt{2}+\sqrt{2}\epsilon_1+3\epsilon_2+2\epsilon_3+|\epsilon_2-\epsilon_3|} \textbf{BellMeasure}^{\epsilon_1,\epsilon_2,\epsilon_3,T} \tag{5.16}$$

Lastly, designating the judgment **QTP** with the erroneous implementations of **EPR**, **BellMeasure**, **Correction**, **Id**, by $\textbf{QTP}^{\epsilon_1,\epsilon_2,\epsilon_3,T,p,\gamma}$, given equations (5.11), (5.16), (5.8), and (5.13), using our deductive metric system, it follows that

$$\textbf{QTP} =_{3\sqrt{2}\epsilon_1+6\epsilon_2+4\epsilon_3+(2+\sqrt{2})|\epsilon_2-\epsilon_3|+2\sqrt{2}+\sqrt{2p}+2\sqrt{\gamma}} \textbf{QTP}^{\epsilon_1,\epsilon_2,\epsilon_3,T,p,\gamma}$$

Chapter 6

Future work

This work focuses specifically on the metric quantale. A natural direction for future research would be to generalize the metric equations and the associated results of soundness and completeness to other quantales, such as the Boolean, ultrametric, and Gödel quantales.

In the case of the Boolean quantale, the relevant equations are labelled by $\{0, 1\}$. The judgement $\Gamma \triangleright v =_1 w : \mathbb{A}$ can be treated as an inequation $\Gamma \leq v =_1 w : \mathbb{A}$, whereas $\Gamma \triangleright v =_0 w : \mathbb{A}$ corresponds to a trivial equation—that is, one that always holds. In this context, it would be interesting to explore, for instance, affine Boolean λ-theories in the setting of real-time computation, particularly in scenarios where the exact timing difference between two programs is irrelevant—what matters is simply whether one program finishes before the other [1]. For the ultrametric quantale, one could investigate ultrametric λ-theories [1] within computational paradigms such as the guarded λ-calculus [118] and functional reactive programming [119]. Finally, the Gödel quantale, which underlies fuzzy logic [120], gives rise to what we refer to as *fuzzy inequations*.

Another possible direction stems from the fact that the quantum categories discussed in section 5.5 are not closed. In [1], the authors used general results from category theory to address a similar issue in the category CPTP. A natural next step would be to extend such a construction for Q and $(\mathrm{W}^*_{\mathrm{CPSU}})^{\mathrm{op}}$.

In this work, our focus is limited to showing that the metric induced by this norm makes $(\mathrm{W}^*_{\mathrm{CPSU}})^{\mathrm{op}}$ into a first-order model. In future work, we aim to explore more deeply the relationship between this norm and the completely bounded norm, as well as to establish additional results that simplify distance computations between morphisms. That would, for instance, allows us to reason about quantum walks on a line [121].

[122] extends [1] by introducing a sound and complete $\mathcal{V}$- equation system —which includes the metric quantale— for aλ-calculus with graded modal types, allowing multiple uses of

the same resource. Since this work does not consider quantum computation, a natural next step would be to explore categorical models suited for this setting. Such an extension would enable us to reason about approximate equivalence in various scenarios, such as discriminating between two known states given n copies of an unknown state [123], or estimating an unknown parameter across n copies of quantum channels in quantum metrology [124, 125].

Bibliography

[1] Fredrik Dahlqvist and Renato Neves. The syntactic side of autonomous categories enriched over generalised metric spaces. *Logical Methods in Computer Science*, Volume 19, Issue 4, 2023.

[2] Fernando Ferreira. O problema da decisão e a máquina universal de turing.

[3] Richard Zach. Chapter 71 - Kurt Gödel, paper on the incompleteness theorems (1931). In I. Grattan-Guinness, Roger Cooke, Leo Corry, Pierre Crépel, and Niccolo Guicciardini, editors, *Landmark Writings in Western Mathematics 1640-1940*, pages 917–925. Elsevier Science, 2005.

[4] Kurt Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37(1):349–360, 1930-12-01.

[5] Torkel Franzén. *Gödel's Theorem*. A. K. Peters.

[6] D. Hilbert and W. Ackermann. *Grundzüge Der Theoretischen Logik*. Springer, 1928.

[7] Alonzo Church. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58(2):345–363, 1936.

[8] Simon Peyton Jones. *The Implementation of Functional Programming Languages*. Prentice Hall, 1987.

[9] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Cambridge University Press, 1989.

[10] Joachim Lambek. From lambda-calculus to cartesian closed categories. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, pages 375–402, 1980.

[11] J. Lambek and P. J. Scott. *Introduction to Higher-Order Categorical Logic*. Cambridge University Press, 1988.

[12] Glynn Winskel. *The Formal Semantics of Programming Languages: An Introduction*. MIT Press, 1993.

[13] Fredrik Dahlqvist and Renato Neves. An Internal Language for Categories Enriched over Generalised Metric Spaces. In Florin Manea and Alex Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*, volume 216 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[14] 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.

[15] 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.

[16] Matteo Mio, Ralph Sarkis, and Valeria Vignudelli. Universal quantitative algebra for fuzzy relations and generalised metric spaces. *Log. Methods Comput. Sci.*, 20(4), 2024.

[17] Jan Jurka, Stefan Milius, and Henning Urbat. Algebraic reasoning over relational structures. *CoRR*, abs/2401.08445, 2024.

[18] Ugo Dal Lago, Furio Honsell, Marina Lenisa, and Paolo Pistone. On quantitative algebraic higher-order theories. In Amy P. Felty, editor, *7th International Conference on Formal Structures for Computation and Deduction, FSCD 2022, August 2-5, 2022, Haifa, Israel*, volume 228 of *LIPIcs*, pages 4:1–4:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[19] Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva, editors. *Foundations of Probabilistic Programming*. Cambridge University Press, 2020.

[20] 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.

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

[22] Kenta Cho. Semantics for a Quantum Programming Language by Operator Algebras. *New Generation Computing*, 34(1):25–68, 2016.

[23] Ian Mackie, Leopoldo Román, and Samson Abramsky. An internal language for autonomous categories. *Applied Categorical Structures*, 1(3):311–343, 1993.

[24] Roy L. Crole. *Categories for Types*. Cambridge University Press, 1994.

[25] Peter Selinger. Lecture notes on the lambda calculus, 2013.

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

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

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

[29] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *Continuous Lattices and Domains*. Encyclopedia of Mathematics and Its Applications. Cambridge University Press, 2003.

[30] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2 edition, 2002.

[31] Isar Stubbe. An introduction to quantaloid-enriched categories. 256:95–116, 2014.

[32] Karel De Leeuw, Edward F Moore, Claude E Shannon, and Norman Shapiro. Computability by probabilistic machines. *Automata studies*, 34:183–198, 1956.

[33] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, 2014.

[34] Sebastian Thrun et al. Robotic mapping: A survey. 2002.

[35] Christopher Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[36] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[37] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[38] Noah D. Goodman, Vikash K. Mansinghka, Daniel Roy, Keith Bonawitz, and Joshua B. Tenenbaum. Church: a language for generative models. UAI'08, page 220–229, Arlington, Virginia, USA, 2008. AUAI Press.

[39] David Tolpin, Jan-Willem van de Meent, and Frank Wood. Probabilistic programming in anglican. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part III 15*, pages 308–311. Springer, 2015.

[40] Raphaëlle Crubillé and Ugo Dal Lago. Metric reasoning about λ-terms: The affine case. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 633–644, 2015.

[41] Raphaëlle Crubillé and Ugo Dal Lago. Metric Reasoning About $$\lambda $$-Terms: The General Case. In Hongseok Yang, editor, *Programming Languages and Systems*, pages 341–367. Springer, 2017.

[42] Walter Rudin. *Functional Analysis*. McGraw-Hill, 1991.

[43] *Infinite Dimensional Analysis*. Springer-Verlag.

[44] Raymond A. Ryan. *Introduction to Tensor Products of Banach Spaces*. Springer Science & Business Media, 2013.

[45] Fredrik Dahlqvist, Alexandra Silva, and Dexter Kozen. Semantics of Probabilistic Programming: A Gentle Introduction. In Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva, editors, *Foundations of Probabilistic Programming*, pages 1–42. Cambridge University Press, 2020.

[46] Vladimir I. Bogachev. *Measure Theory*. Springer, 2007.

[47] Jean Goubault-Larrecq. *Non-Hausdorff Topology and Domain Theory: Selected Topics in Point-Set Topology*. New Mathematical Monographs. Cambridge University Press, 2013.

[48] Krishna B. Athreya and Soumendra N. Lahiri. *Measure Theory and Probability Theory*. Springer Science & Business Media, 2006.

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

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

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

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

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

[54] 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.

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

[56] 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.

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

[58] 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.

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

[60] 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.

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

[62] Peter Selinger and Benoît Valiron. On a fully abstract model for a quantum linear functional language. *Electronic Notes in Theoretical Computer Science*, 210:123–137, 2008.

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

[64] Michele Pagani, Peter Selinger, and Benoît Valiron. Applying quantitative semantics to higher-order quantum computing. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '14, page 647–658, New York, NY, USA, 2014. Association for Computing Machinery.

[65] Kenta Cho and Abraham Westerbaan. Von Neumann Algebras form a Model for the Quantum Lambda Calculus, 2016.

[66] Peter Selinger and Benoît Valiron. A linear-non-linear model for a computational call-by-value lambda calculus. In *International Conference on Foundations of Software Science and Computational Structures*, pages 81–96. Springer, 2008.

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

[68] 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.

[69] 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.

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

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

[72] Teiko Heinosaari and Mário Ziman. *The Mathematical Language of Quantum Theory: From Uncertainty to Entanglement*. Cambridge University Press, 2011.

[73] John B. Conway. *A Course in Functional Analysis*, volume 96 of *Graduate Texts in Mathematics*. Springer, 2007.

[74] John B. Conway. *A Course in Operator Theory*. American Mathematical Society, 2000.

[75] Shôichirô Sakai. *C*-Algebras and W*-Algebras*, volume 60 of *Classics in Mathematics*. Springer, 1998.

[76] Masamichi Takesaki, editor. *Theory of Operator Algebras I*. Springer, 1979.

[77] Abraham A. Westerbaan. The Category of Von Neumann Algebras, 2019.

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

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

[80] John S Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195, 1964.

[81] Alain Aspect, Jean Dalibard, and Gérard Roger. Experimental test of bell's inequalities using time-varying analyzers. *Physical review letters*, 49(25):1804, 1982.

[82] Wolfgang Tittel, Jürgen Brendel, Bernard Gisin, Thomas Herzog, Hugo Zbinden, and Nicolas Gisin. Experimental demonstration of quantum correlations over more than 10 km. *Physical Review A*, 57(5):3229, 1998.

[83] Jian-Wei Pan, Dik Bouwmeester, Harald Weinfurter, and Anton Zeilinger. Experimental entanglement swapping: entangling photons that never interacted. *Physical review letters*, 80(18):3891, 1998.

[84] Juan Yin, Yuan Cao, Yu-Huai Li, Sheng-Kai Liao, Liang Zhang, Ji-Gang Ren, Wen-Qi Cai, Wei-Yue Liu, Bo Li, Hui Dai, et al. Satellite-based entanglement distribution over 1200 kilometers. *Science*, 356(6343):1140–1144, 2017.

[85] Yu Shi and Edo Waks. Error metric for non-trace-preserving quantum operations. *Phys. Rev. A*, 108:032609, Sep 2023.

[86] Daniel A Lidar. Lecture notes on the theory of open quantum systems. *arXiv preprint arXiv:1902.00967*, 2019.

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

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

[89] Gert K. Pedersen. *Analysis Now*, volume 118 of *Graduate Texts in Mathematics*. Springer, 1989.

[90] Steven Roman. *Advanced Linear Algebra*, volume 135 of *Graduate Texts in Mathematics*. Springer, 1992.

[91] W. Heisenberg. Über quantentheoretische umdeutung kinematischer und mechanischer beziehungen. 33(1):879–893, 1925.

[92] M. Born and P. Jordan. Zur quantenmechanik. 34(1):858–888, 1925.

[93] J von Neumann. Wahrscheinlichkeitstheoretischer aufbau der quantenmechanik. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1927:245–272, 1927.

[94] Ola Bratteli and Derek W. Robinson. *Operator Algebras and Quantum Statistical Mechanics 1*. Springer, 1987.

[95] Huzihiro Araki. *Mathematical Theory of Quantum Fields*. Oxford University Press, 1999.

[96] Rudolf Haag and Daniel Kastler. An Algebraic Approach to Quantum Field Theory. *Journal of Mathematical Physics*, 5(7):848–861, 1964.

[97] Michael Keyl. Fundamentals of quantum information theory. *Physics Reports*, 369(5):431–548, 2002.

[98] Alain Connes. *Noncommutative Geometry*. Academic Press, 1995.

[99] Jan Hamhalter. *Quantum Measure Theory*. Springer Science & Business Media, 2003.

[100] Hans Maassen. Quantum probability and quantum information theory. *Lecture Notes in Physics*, 808:65–108, 07 2010.

[101] Gert Kjaergård Pedersen. *C*-Algebras and Their Automorphism Groups*. Academic Press, 1979.

[102] Vern Paulsen. *Completely Bounded Maps and Operator Algebras*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2003.

[103] Edward G. Effros and Zhong-Jin Ruan. *Operator Spaces*. Clarendon Press, 2000.

[104] Francis Borceux. *Handbook of Categorical Algebra: Volume 2: Categories and Structures*, volume 2 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 1994.

[105] Peter Selinger. Towards a semantics for higher-order quantum computation. *Proceedings of the 2nd International Workshop on Quantum Programming Languages*, pages 127–143, 01 2004.

[106] Nathanial Patrick Brown and Narutaka Ozawa. *C*-Algebras and Finite-dimensional Approximations*. American Mathematical Soc., 2008.

[107] Gilles Pisier. *Tensor Products of C*-Algebras and Operator Spaces: The Connes–Kirchberg Problem*. London Mathematical Society Student Texts. Cambridge University Press, 2020.

[108] Gilles Pisier. *Introduction to Operator Space Theory*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2003.

[109] Elias Zakon. *Mathematical Analysis I*. The Saylor Foundation, 2011.

[110] Stephen Barnett. *Quantum Information*. Oxford University Press, Inc., USA, 2009.

[111] Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden. Quantum cryptography. *Rev. Mod. Phys.*, 74:145–195, Mar 2002.

[112] 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.

[113] 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.

[114] 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.

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

[116] 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.

[117] Jing Wang, Li Jiang, Han Zhang, Hanzhuang Zhang, and Liquan Zhang. Fidelity of structured amplitude-damping channels. *Physica Scripta*, 83(4):045008, mar 2011.

[118] Lars Birkedal, Jan Schwinghammer, and Kristian Støvring. A metric model of lambda calculus with guarded recursion. Informal workshop proceedings, Aug 2010.

[119] Neelakantan R. Krishnaswami and Nick Benton. Ultrametric semantics of reactive programs. In *Proceedings of the 2011 IEEE 26th Annual Symposium on Logic in Computer Science*, LICS '11, page 257–266, USA, 2011. IEEE Computer Society.

[120] K. Denecke, M. Erné, and S. L. Wismath, editors. *Galois Connections and Applications*. Springer Netherlands, 2004.

[121] Salvador Elías Venegas-Andraca. Quantum walks: A comprehensive review. *Quantum Information Processing*, 11(5):1015–1106, 2012.

[122] Fredrik Dahlqvist and Renato Neves. A Complete V-Equational System for Graded lambda-Calculus. *Electronic Notes in Theoretical Informatics and Computer Science*, Volume 3 - Proceedings of MFPS XXXIX, 2023.

[123] A. Acín, E. Bagan, M. Baig, Ll. Masanes, and R. Muñoz Tapia. Multiple-copy two-state discrimination with individual measurements. *Phys. Rev. A*, 71:032338, Mar 2005.

[124] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum metrology. *Phys. Rev. Lett.*, 96:010401, Jan 2006.

[125] Sisi Zhou. Limits of noisy quantum metrology with restricted quantum controls. *Phys. Rev. Lett.*, 133:170801, Oct 2024.

[126] Samuel Eilenberg and Saunders MacLane. General Theory of Natural Equivalences. *Transactions of the American Mathematical Society*, 58(2):231–294, 1945.

[127] Noson S. Yanofsky. Monoidal Category Theory: Unifying Concepts in Mathematics, Physics, and Computing (Lecture Slides), 2024.

[128] Steve Awodey and Steve Awodey. *Category Theory*. Oxford Logic Guides. Oxford University Press, second edition, second edition edition, 2010.

Part III
Appendices

Appendix A
Mathematical backgound

A.1 Equivalence Relations and Quotients in Sets

Definition A.1.1. A relation $\sim$ on a set S is an *equivalence relation* if it is

- reflexive: for all $x \in S, x \sim x$,
- symmetric: for all $x, y \in S$, if $x \sim y$ then $y \sim x$, and
- transitive: for all $x, y, z \in S$, if $x \sim y$ and $y \sim z$, then $x \sim z$.

Definition A.1.2. Given an equivalence relation on a set S, we can describe the so-called *equivalence classes*. If $s \in S$, then the *equivalence class* of s is the set of all elements related to it:

$$[s] = \{r \in S \mid r \sim s\}.$$

That is, $[s]$ is the set of all elements that are considered "the same" as s under the relation $\sim$. For a given set S and an equivalence relation $\sim$ on S, we define the *quotient set*, denoted $S/\ \sim$, whose elements are all the equivalence classes of elements in S. Observe that the quotient mapping

$$q : S \longrightarrow S/{\sim},$$

which takes an element $s \in S$ to its equivalence class $[s]$, has the property that a map $f : X \to Y$ extends along q,

$$\begin{tikzcd}
S \arrow[r, "q"] \arrow[dr, "f"'] & S/{\sim} \arrow[d, dotted] \\
& P
\end{tikzcd}$$

just in case f respects the equivalence relation, in the sense that $s \sim p$ implies $f(s) = f(p)$. For instance, consider a set of cars S. We can define an equivalence relation on S by grouping cars according to their colour. This results in subsets such as the set of blue cars, the set of red cars, the set of green cars, and so on — these subsets are the *equivalence classes*. Moreover, the collection of all such equivalence classes forms a new set, called the *quotient set*.

A.2 Category theory

Category theory originated as an effort to connect and unify two distinct areas of mathematics. The goal was to study and classify specific geometric structures—such as topological spaces, manifolds, and bundles—by associating them with corresponding algebraic structures like groups, rings, and abelian groups. It became clear that a language was needed to connect geometric and algebraic objects—one not explicitly tailored to geometry or algebra. Only a language of such generality could allow meaningful discussion across both fields. This is the birth of category theory. Described as "a language about nothing, and therefore about everything" category theory provides a highly general way of discussing mathematical concepts. It was invented by Samuel Eilenberg and Saunders MacLane [126]. They organized various mathematical structures into categories called geometric and algebraic. To connect these categories, they defined functors, which map objects and morphisms from one category to another, much like functions do. They further introduced natural transformations, which provide a way to compare functors, translating the results of one functor into those of anothers. [127]

As previously mentioned, in light os its deep connection with lambda calculus and its capacity to encompass multiple "perspectives", thereby broadening the applicability of results, we adopt a categorical interpretation.

A.2.1 Categories

Definition A.2.1. A *category* C consists of

- a collection of objects $A, B, C, \ldots$, denoted |C| or Obj(C);
- for every two pairs of objects A and B, a collection of morphisms $f, g, \ldots$, usually denoted C(A, B), $\mathrm{Hom}_{\mathsf{C}}(A, B)$, or $\mathrm{Hom}(A, B)$ if there is no ambiguity.

The collection for morphisms has the following structure:

- Each morphism has a specified domain(A) and codomain(B) and the notation $f : A \to B$ indicates that f is a morphism from object A to object B.
- Every object A has an identity morphism $\mathrm{id}_A : A \to A$.
- For any pair of morphisms $f : A \to B$ and $g : B \to C$, there exists a composite morphism $g \circ f : X \to Z$. We will also write $g \circ f$ as $f \cdot g$ or simply fg.

The composition is required to satisfy the two following laws: if $f : A \to B, g : B \to C$, and $h : C \to D$ are morphisms, then

- $f \circ \mathrm{id}_A = f = \mathrm{id}_B \circ f$;

- $(f \circ g) \circ h = f \circ (g \circ h)$.

Example A.2.2. Set is the category whose objects are sets and whose morphisms are functions between them. Given a function $f : A \to B$, it assigns to each element $a \in A$ a unique element $f(a) \in B$. For any two functions $f : A \to B$ and $g : B \to C$, their composition is defined by

$$(g \circ f)(a) = g(f(a)) \quad \text{for all } a \in A.$$

This composition is *associative*. That is, for any further function $h : C \to D$, we have

$$(h \circ g) \circ f = h \circ (g \circ f),$$

since for every $a \in A$,

$$((h \circ g) \circ f)(a) = h(g(f(a))) = (h \circ (g \circ f))(a).$$

Moreover, for every set A, there exists an *identity function*

$$\mathrm{id}_A : A \to A, \quad \text{defined by } \mathrm{id}_A(a) = a,$$

which satisfies the unit laws for composition:

$$f \circ \mathrm{id}_A = f \quad \text{and} \quad \mathrm{id}_B \circ f = f$$

for any function $f : A \to B$.
Therefore, Set, with sets as objects and functions as morphisms, satisfies the axioms of a category.

Another common type of example consists of categories of sets equipped with additional structure, along with functions that preserve that structure.

Definition A.2.3. A *partially ordered set* or *partial order* is a set A equipped with a binary relation $\leq_A$ satisfying the following properties for all $a, b, c \in A$:

- Reflexivity: $a \leq_A a$;
- Transitivity: If $a \leq_A b$ and $b \leq_A c$, then $a \leq_A c$;
- Antisymmetry: If $a \leq_A b$ and $b \leq_A a$, then $a = b$.

Example A.2.4. The set of real numbers $\mathbb{R}$, equipped with the usual ordering $\leq$, forms a poset. Moreover, it is *linearly ordered* (or *totally ordered*), since for any $x, y \in \mathbb{R}$, either $x \leq y$ or $y \leq x$ holds.

Example A.2.5. Each partially ordered set naturally defines a category. Let $(P, \leq)$ be a poset. We define a category B(P, $\leq$), often denoted simply by B(P) or even P, where the objects are the elements of P, and there is a unique morphism $p \to q$ if and only if $p \leq q$. The reflexivity of the order $\leq$ ensures the existence of identity morphisms, while transitivity guarantees that morphisms compose appropriately. Moreover, since there is at most one morphism between any two objects, composition is trivially associative.

Definition A.2.6. Given two partial orders $(A, \leq_A)$ and $(B, \leq_B)$, a function $m : A \to B$ is called a *monotone map* (or *order-preserving map*) if for all $a, a' \in A$,

$$a \leq_A a' \quad \Rightarrow \quad m(a) \leq_B m(a').$$

Example A.2.7. PO is the category of all partial orders and all monotone maps. First, for any poset A, the identity function $\mathrm{id}_A : A \to A$ is monotone. Indeed, for all $a \in A$,

$$a \leq_A a \quad \Rightarrow \quad \mathrm{id}_A(a) \leq_A \mathrm{id}_A(a).$$

Next, given monotone maps $f : A \to B$ and $g : B \to C$, their composition $g \circ f : A \to C$ is also monotone. For all $a, a' \in A$, if $a \leq_A a'$, then

$$f(a) \leq_B f(a') \quad \text{and} \quad g(f(a)) \leq_C g(f(a')),$$

so it follows that

$$(g \circ f)(a) \leq_C (g \circ f)(a').$$

Example A.2.8. CVect is the category of finite complex vector spaces and linear mappings.

Definition A.2.9. A morphism $f : A \to B$ in a category C be a category is called an *isomorphism* if there exists a morphism $f^{-1} : B \to A$ such that

$$f^{-1} \circ f = \mathrm{id}_A \quad \text{and} \quad f \circ g = \mathrm{id}_B.$$

In this case, f^{-1} is called the *inverse* of f, and it is unique. If such an isomorphism exists, we say that A and B are *isomorphic*, written $A \cong B$.

One of the central ideas in category theory is *duality*. Simply put, for a given definition of a structure, there is often a corresponding dual concept obtained by reversing the directions of all the morphisms.

Definition A.2.10. Let C be a category. The *opposite category*, denoted C$^{\mathrm{op}}$, is defined as follows:

- The objects of C$^{\mathrm{op}}$ are the same as those of C.
- For any pair of objects A, B, the hom-set in C$^{\mathrm{op}}$ is defined by

  $$\mathit{Hom}_{\mathsf{C}^{\mathrm{op}}}(A, B) = \mathit{Hom}_{\mathsf{C}}(B, A),$$

  that is, each morphism $f : A \to B$ in C$^{\mathrm{op}}$ corresponds to a morphism $f : B \to A$ in C.
- Composition in C$^{\mathrm{op}}$ is defined using the composition in C, but in reverse order. That is, if

  $$A \xrightarrow{f} B \xrightarrow{g} C$$

  are morphisms in C$^{\mathrm{op}}$, corresponding to morphisms

  $$C \xrightarrow{g} B \xrightarrow{f} A$$

  in C, then the composition in C$^{\mathrm{op}}$ is defined by

  $$g \circ f := f \circ_{\mathsf{C}} g.$$

Thus, C$^{\mathrm{op}}$ reverses the direction of morphisms and composition while retaining the same collection of objects.

Definition A.2.11. A subcategory D of a category C is a category such that

- All the objects of D are objects of C;
- For any objects A and B in D, we have $\mathrm{Hom}_{\mathsf{D}}(A, B) \subseteq \mathrm{Hom}_{\mathsf{C}}(A, B)$.
- The indentities in D are those of C and the composition in D is the respective restriction relative to C.

Example A.2.12. The category FinSet, whose objects are finite sets and whose morphisms are functions between them, forms a subcategory of the category Set.

Definition A.2.13. A category is called *small* if both its collection of objects and its collection of morphisms form sets. A category is called *locally small* if, for every pair of objects, the corresponding hom-set is a set.

A.2.2 Products and coproducts

A category frequently possesses a more intricate structure than a mere collection of objects and their morphisms. The existence of particular relationships among certain objects and morphisms can give some objects important properties.
A diagram is said to commute if, for every pair of objects A and B in the diagram, all directed paths from A to B yield equal morphisms.

Definition A.2.14. An object 0 in a category C is called an *initial object* if for every object $A \in$ C, there exists a unique morphism $f : 0 \to A$.

Definition A.2.15. An object 1 in a category C is called a *terminal object* if for every object $A \in$ C, there exists a unique morphism $f : A \to 1$.

Example A.2.16. In the category Set, the empty set $\emptyset$ is an initial object, since for any set S, there exists a unique function$f : \emptyset \to S$. This function is unique because there are no elements in $\emptyset$ to map to.
Any singleton set, such as $\{*\}$ or $\{a\}$, is a terminal object in this category. For any set S, there exists a unique function $f : S \to \{*\}$, which maps every element of S to the sole element of the singleton set

Example A.2.17. Let $(P, \leq)$ be a partial order and P be its associated category. Here, the initial object is the *bottom element*—an element that is less than or equal to every other element in P. The terminal object in P is the *top element*—an element that is greater than or equal to every other element in P.

Definition A.2.18. Consider a category C. We say that it has (binary) products if for any objects A and B in C there also exists an object $A \times B$ in C with morphisms $\pi_A : A \times B \to A$ and $\pi_B : A \times B \to B$ that satisfy a certain universal property: specifically for every two morphisms $f : C \to A$ and $g : C \to B$ there exists a *unique* morphism $\langle f, g\rangle : C \to A \times B$ called *pairing* that makes the diagram below commute.

$$\begin{tikzcd}
 & C \arrow[ld, "f"'] \arrow[d, dotted, "\langle f, g\rangle"] \arrow[rd, "g"] & \\
A & A \times B \arrow[l, "\pi_A"] \arrow[r, "\pi_B"'] & B
\end{tikzcd}$$

Definition A.2.19. Let $A \times B$ be a product of objects A and B, and let $A' \times B'$ be a product of objects A' and B' in a category C. Suppose we are given morphisms $f : A \to A'$ and $g : B \to B'$. Then there exists a unique morphism

$$f \times g : a \times b \to a' \times b'$$

such that the following diagram commutes.

$$\begin{tikzcd}
A \arrow[d, "f"'] & A \times B \arrow[l, "\pi_A"] \arrow[r, "\pi_B"'] \arrow[d, dotted, "f \times g"] & B \arrow[d, "g"] \\
A' & A' \times B' \arrow[l, "\pi'_A"] \arrow[r, "\pi'_B"'] & B'
\end{tikzcd}$$

This induced morphism $f \times g$ is called the *product of the morphisms* f and g, and it is given explicitly by

$$f \times g = \langle f \circ \pi_A,\, g \circ \pi_B\rangle.$$

Theorem A.2.20. *Let $A \times B$ be the product of objects A and B in a category* C*. For any object C and morphisms $f : C \to A$, $g : C \to B$, $h : D \to C$ are morphisms, it holds that:*

$$\langle f \circ h,\, g \circ h\rangle = \langle f, g\rangle \circ h.$$

Proof. The universal property of the product induces a unique morphism $\langle f, g\rangle : C \to A \times B$ such that $\pi_A \circ \langle f, g\rangle = f \quad$ and $\quad \pi_B \circ \langle f, g\rangle = g$. Now, let $h : D \to C$ be another morphism. Then the compositions $f \circ h : D \to A$ and $g \circ h : D \to B$ also induce a unique morphism $\langle f \circ h,\, g \circ h\rangle : D \to A \times B$ by the universal property of the product. As a result, the following diagram commutes by the universal property of the product.

$$\begin{tikzcd}
& D \arrow[d, "h"] \arrow[dddl, "f \circ h"'] \arrow[dddr, "g \circ h"] \arrow[dd, dotted, bend right, "\langle f \circ h, g \circ h \rangle"'] & \\
& C \arrow[ddl, "f"'] \arrow[ddr, "g"] \arrow[d, dotted, "\langle f, g \rangle"] & \\
& A \times B \arrow[dl, "\pi_A"] \arrow[dr, "\pi_B"'] & \\
A & & B
\end{tikzcd}$$

□

Example A.2.21. In the category Set, the product of two sets A and B is given by their Cartesian product, denoted as

$$A \times B = \{(a, b) \mid a \in A,\ b \in B\}.$$

The projection maps are defined by

$$\pi_A(a, b) = a \quad \text{and} \quad \pi_B(a, b) = b.$$

Given a set C and morphisms $f : C \to A$ and $g : C \to B$, their pairing is the map

$$\langle f, g\rangle(c) = (f(c), g(c)).$$

Example A.2.22. Let $(P, \leq)$ be a partial order and P be its associated category. Consider a product of elements $p \times q \in P$. Then, by definition, there must exist projections satisfying

$$p \times q \leq p \quad \text{and} \quad p \times q \leq q.$$

Furthermore, for any element $x \in P$, if

$$x \leq p \quad \text{and} \quad x \leq q,$$

then it follows that

$$x \leq p \times q.$$

This operation $p \times q$ corresponds to what is commonly known as the *greatest lower bound* or *meet*, and is typically denoted by $p \wedge q$.

Example A.2.23. In the category CVect, the product of two vector spaces V and W corresponds to their direct sum, denoted by $V \oplus W$. The projection maps are the linear maps

$$\pi_V : V \oplus W \to V, \quad \pi_V(v, w) = v,$$

$$\pi_W : V \oplus W \to W, \quad \pi_W(v, w) = w.$$

Given any vector space U and linear maps $f : U \to V$ and $g : U \to W$, the unique map $\langle f, g\rangle : U \to V \oplus W$ is defined by

$$\langle f, g\rangle(u) = (f(u), g(u)).$$

The *coproduct* is the dual of the *product*—it is obtained by reversing all the morphisms in the definition of a product. Consequently, a product in a category C corresponds to a coproduct in the opposite category C$^{\mathrm{op}}$. More explicitly,

Definition A.2.24. Consider a category C. We say that it has (binary) coproducts if for any objects A and B in C there also exists an object $A \oplus B$ in C with morphisms $\mathrm{inl} : A \to A \oplus B$ and $\mathrm{inr} : B \to A \oplus B$ that satisfy a certain universal property: specifically for every two morphisms $f : A \to C$ and $g : B \to C$ there exists a *unique* morphism $[f, g] : A \oplus B \to C$ known as *co-pairing* that makes the diagram below commute.

$$\begin{tikzcd}
 & C & \\
A \arrow[ur, "f"] \arrow[r, "\mathrm{inl}"'] & A \oplus B \arrow[u, dotted, "{[f,g]}"'] & B \arrow[ul, "g"'] \arrow[l, "\mathrm{inr}"]
\end{tikzcd}$$

Definition A.2.25. Let $A \oplus B$ be a coproduct of objects A and B, and let $A' \oplus B'$ be a coproduct of objects A' and A' in a category C. Suppose we are given morphisms $f : A \to A'$ and $g : B \to B'$. Then there exists a unique morphism

$$f \oplus g : A \oplus B \to A' \oplus B'$$

such that the following diagram commutes.

$$\begin{tikzcd}
A \arrow[r, "\mathrm{inl}"'] \arrow[d, "f"'] & A \oplus B \arrow[d, dotted, "f \oplus g"] & B \arrow[l, "\mathrm{inr}"] \arrow[d, "g"] \\
A' \arrow[r, "\mathrm{inl}"'] & A' \oplus B' & B' \arrow[l, "\mathrm{inr}"]
\end{tikzcd}$$

This induced morphism $f \oplus g$ is called the *coproduct of the morphisms* f and g, and it is given explicitly by

$$f \oplus g = [\mathrm{inl} \circ f, \, \mathrm{inr} \circ g].$$

Theorem A.2.26. *Let $A \oplus B$ be the product of objects A and B in a category* C*. For any object C and morphisms $f : A \to C$ and $g : B \to C$ are morphisms, it holds that:*

$$[h \circ f, \, h \circ g] = h \circ [f, g].$$

Proof. This result is a direct consequence of the duality with products. □

Proposition A.2.27. *[128, Proposition 3.12] Coproducts are unique up to isomorphism. Explicitly, this can be formulated as follows: let $(C, \mathrm{inl} : A \to C, \mathrm{inr} : B \to C)$ and $(C', \mathrm{inl}' : A \to C', \mathrm{inr}' : B \to C')$ be two coproducts of objects A and B in a category. Then there exists a unique isomorphism $\varphi : C \to C'$ such that*

$$\varphi \cdot \mathrm{inl} = \mathrm{inl}' \quad \textit{and} \quad \varphi \cdot \mathrm{inr} = \mathrm{inr}'.$$

Example A.2.28. In the category Set, the coproduct $A \oplus B$ of two sets is their disjoint union, which can be constructed as

$$A \oplus B = \{(a,1) \mid a \in A\} \cup \{(b,2) \mid b \in B\}.$$

The canonical coproduct injections are defined by

$$\mathrm{inl}(a) = (a,1), \quad \mathrm{inr}(b) = (b,2).$$

Given any set C and functions $f : A \to C$ and $g : B \to C$, the copairing $[f,g] : A \oplus B \to C$ is defined by

$$[f,g](x,\delta) = \begin{cases} f(x) & \text{if } \delta = 1, \\ g(x) & \text{if } \delta = 2. \end{cases}$$

Example A.2.29. Let $(P, \leq)$ be a partial order and P be its associated category. Consider a coproduct of elements $p \oplus q \in P$. Then, by definition, there must exist injections satisfying

$$p \leq p \oplus q \quad \text{and} \quad q \leq p \oplus q.$$

Furthermore, for any element $z \in P$, if

$$p \leq z \quad \text{and} \quad q \leq z,$$

then it follows that

$$p \oplus q \leq z.$$

This operation $p + q$ corresponds to what is commonly known as the *least upper bound* or *join*, and is typically denoted by $p \vee q$.

Example A.2.30. In CVect the coproduct coincides with the product. In such cases, this structure is called a *biproduct*. In CVect the injection maps are the linear maps

$$\mathrm{inl} : V \to V \oplus W, \quad \mathrm{inl}(v) = (v, 0),$$

$$\mathrm{inr} : W \to V \oplus W, \quad \mathrm{inr}(w) = (0, w).$$

Given any vector space U and linear maps $f : V \to U$ and $g : W \to U$, the unique map $[f,g] : V \oplus W \to U$ is defined by

$$[f,g](v,w) = f(v) + g(w).$$

Up until how we have only discussed binary products/coproducts. However, we can also define *ternary products*$A_1 \times A_2 \times A_3$ with an analogous universal property. That is, there exist three projection morphisms $\pi_i : A_1 \times A_2 \times A_3 \to A_i$ for $i = 1, 2, 3$, and for any object B and morphisms $f_i : B \to A_i$, there exists a unique morphism $\langle f_1, f_2, f_3 \rangle : B \to A_1 \times A_2 \times A_3$ such that $\pi_i \cdot \langle f_1, f_2, f_3 \rangle = f_i$ for each $i = 1, 2, 3$. Such a condition can be formulated for any number of factors and if a category has binary products, then it has all finite products, *i.e.*any finite number $n \geq 1$ of factors. Any object A is the unary product of A with itself one time. Observe also that a terminal object is a "nullary" product, that is, a product of no objects: given no objects, there exists an object $\mathbf{1}$ with no projectors, and for any other object A, there exists a unique arrow $! : A \to \mathbf{1}$ making no additional diagrams commute. One can also define the product of a family of objects $(C_i)_{i \in I}$ indexed by a set I, as an object $\prod_{i \in I} C_i$ together with a family of projection morphisms

$$\pi_i : \prod_{j \in I} C_j \to C_i \quad \text{for each } i \in I,$$

such that for every object A and every family of morphisms $(f_i : A \to C_i)_{i \in I}$, there exists a unique morphism $u : A \to \prod_{i \in I} C_i$ such that $\pi_i \cdot u = f_i \quad$ for all $i \in I$.
Reversing all arrows in the definitions above yields the notion of *finite coproducts* and the *coproduct* of a family of objects $(C_i)_{i \in I}$.

Definition A.2.31. A category C is said to have *all small products* if every set of objects in C has a product.

A.2.3 Functors

Although categories are already interesting on their own, the real strength of category theory lies in understanding how categories relate to one another. Just as functions express relationships between sets, functors play a similar role for categories. A functor maps each object in one category to an object in another category, and it does the same for morphisms, preserving the structure of composition.

Definition A.2.32. Let C and D be two categories. A *functor* $F : \mathsf{C} \to \mathsf{D}$ consists of a mapping that assigns to each object A in C an object FA in D, and to each morphism $f \in \mathrm{Hom}_{\mathsf{C}}(A, B)$ a morphism $Ff \in \mathrm{Hom}_{\mathsf{D}}(FA, FB)$, in such a way that the following two conditions are satisfied for all objects A, B, C in C and all morphisms $f \in \mathrm{Hom}_{\mathsf{C}}(A, B)$ and $g \in \mathrm{Hom}_{\mathsf{C}}(B, C)$:

$$F(\mathrm{id}_A) = \mathrm{id}_{FA}, \qquad F(g \circ f) = F(g) \circ F(f).$$

A functor $F : \mathsf{C} \to \mathsf{D}$ is said to be *full* if, for all objects A and B in C, the induced map

$$F_{A,B} : \mathrm{Hom}_{\mathsf{C}}(A, B) \longrightarrow \mathrm{Hom}_{\mathsf{D}}(FA, FB), \quad f \mapsto Ff,$$

is surjective. The functor is called *faithful* if each $F_{A,B}$ is injective, and *fully faithful* if each $F_{A,B}$ is bijective. A *full embedding* is a functor that is fully faithful and, in addition, injective on objects.

Example A.2.33. Let C be a category. Then there exists an *identity functor* $\mathrm{id}_{\mathsf{C}} : \mathsf{C} \to \mathsf{C}$, which is defined on objects by $\mathrm{id}_{\mathsf{C}}(A) = A$ for every object A in C, and analogously on morphisms, that is, $\mathrm{id}_{\mathsf{C}}(f) = f$ for every morphism f in C.

Example A.2.34. Consider the natural numbers $\mathbb{N}$ as a partial order category. There is a functor $(-) + 5 : \mathbb{N} \to \mathbb{N}$ that maps each object $m \in \mathbb{N}$ to $m + 5$. This defines a functor because it preserves morphisms: if $m \leq m'$, then $m + 5 \leq m' + 5$. Moreover, the identity morphisms are trivially preserved.

Example A.2.35. Consider the set of real numbers $\mathbb{R}$ and the set of integers $\mathbb{Z}$, each regarded as a partial order category. In this context, there exists a functor $\mathrm{Floor} : \mathbb{R} \to \mathbb{Z}$ that assigns to each real number $r \in \mathbb{R}$ the greatest integer less than or equal to r, denoted $\lfloor r \rfloor$. For instance, $\lfloor 6.2 \rfloor = 6$ and $\lfloor -1.66 \rfloor = -2$.
Similarly, there exists a *ceiling functor* $\mathrm{Ceil} : \mathbb{R} \to \mathbb{Z}$ that maps each real number r to the least integer greater than or equal to r, denoted $\lceil r \rceil$.

Example A.2.36. Let P and P′ be partial order categories. Any functor $F \colon$ P $\to$ P′ corresponds precisely to a monotone function between the underlying posets.

Definition A.2.37. Given categories C, D, and E, a *bifunctor* $F :$ C $\times$ D $\to$ E is simply a functor from the product category C $\times$ D to E. In particular, F is a rule that assigns:

- to every objects $A \in$ C and $B \in$ D, an object $F(A, B) \in$ E;
- to every morphisms $f : A \to A'$ in C and $g : B \to B'$ in D, a morphism $F(f, g) : F(A, B) \to F(A', B') \in$ E.

These assignments must satisfy the following two requirements:

- *Respect for composition*: For morphisms $f : A \to A'$, $f' : A' \to A''$ in C and $g : B \to B'$, $g' : B' \to B''$ in D, it should hold that

  $$F(f' \circ f,\, g' \circ g) = F(f', g') \circ F(f, g),$$

  where the $\circ$ on the right-hand side is composition in E.

- *Respect for identities*: For all objects $A \in$ C and $B \in$ D, it should hold that

  $$F(\mathrm{id}_A, \mathrm{id}_B) = \mathrm{id}_{F(A,B)},$$

  where id_A and id_B are the identity morphisms in C and D, respectively, and $\mathrm{id}_{F(A,B)}$ is the identity morphism in E.

Many times, rather than writing the name of the bifunctor before the input, like $F(A, B)$, we write the bifunctor in infix notation, for example, $a \,\Box\, b$. When we use this notation, the condition

$$F(f' \circ f,\, g' \circ g) = F(f', g') \circ F(f, g)$$

becomes

$$(f' \circ f) \,\Box\, (g' \circ g) = (f' \,\Box\, g') \circ (f \,\Box\, g).$$

A.2.4 Natural Tranformations

If category theory is about morphisms, then morphisms between functors should also be a natural concept. These are called *natural transformations*, and provide a way of relating two functors that have the same domain and codomain. Intuitively, if we consider two functors $F, G : \mathsf{C} \to \mathsf{D}$ as different ways of assigning images of the category C into the category D, then a natural transformation $\eta : F \Rightarrow G$ is a coherent way of transforming the image of F into the image of G.

Definition A.2.38. Let C and D be categories, and let $F, G : \mathsf{C} \to \mathsf{D}$ be functors. A *natural transformation* $\eta : F \Rightarrow G$ is a family of morphisms in D,

$$(\eta_A : FA \to GA)_{A \in \mathrm{Ob}(\mathsf{C})},$$

indexed by the objects of C, such that for every morphism $f : A \to A'$ in C, the following diagram commutes.

$$\begin{tikzcd}
FA \arrow[r, "\eta_A"] \arrow[d, "Ff"'] & GA \arrow[d, "Gf"] \\
FA' \arrow[r, "\eta_{A'}"'] & GA'
\end{tikzcd}$$

Given a natural transformation $\eta : F \Rightarrow G$, the morphism $\eta_A : F(A) \to G(A)$ in D is called the *component* of η at A. A natural transformation $\eta : F \Rightarrow G$ is represented diagrammatically as

$$\begin{tikzcd}
\mathsf{C} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Downarrow_{\eta}"] & \mathsf{D}
\end{tikzcd}$$

Example A.2.39. For every functor $F : \mathsf{C} \to \mathsf{D}$, there exists a natural transformation

$$\iota_F : F \Rightarrow F$$

known as *identity natural transformation*, such that for each object $A \in \mathsf{C}$, each component of ι_F is the identity morphism:

$$(\iota_F)_A = \mathrm{id}_{F(A)} : F(A) \to F(A).$$

Example A.2.40. The *list functor*

$$\mathrm{List} : \mathsf{Set} \to \mathsf{Set}$$

assigns to each set S the set of all finite sequences (or lists) of its elements.
For instance, if $S = \{a, b, c\}$, then

$$\mathrm{List}(S) = \{\varepsilon, a, b, c, aa, ab, ac, ba, \ldots, abc, cba, \ldots\},$$

where ε denotes the empty list.
Given a function $f : S \to T$, where $T = \{1, 2\}$, the functor maps it to $\mathrm{List}(f) : \mathrm{List}(S) \to \mathrm{List}(T)$, which applies f to each element of a list. For example, if

$$f(a) = 2, \quad f(b) = 1, \quad f(c) = 2,$$

then $\mathrm{List}(f)(aabccba) = 2212212$.
There exists a natural transformation

$$\mathrm{Reverse} : \mathrm{List} \Rightarrow \mathrm{List},$$

whose component at a set S, $\mathrm{Reverse}_S$, maps each list to its reversal. For example:

$$\mathrm{Reverse}_S(accbab) = babcca.$$

Definition A.2.41. A natural transformation $\eta : F \Rightarrow G$ between functors $F, G : \mathsf{C} \to \mathsf{D}$ is called a *natural isomorphism* if, for every object $A \in \mathsf{C}$, $\eta_A : F(A) \to G(A)$ is an isomorphism in D.

A.2.5 Equivalence of Categories

In category theory, the concept of isomorphism between categories can be quite strict. A more forgiving notion is an *equivalence of categories*.

If $F : \mathsf{C} \to \mathsf{D}$ is an isomorphism of categories, then for every object $B \in \mathsf{D}$, there exists a *unique* object $A \in \mathsf{C}$ such that $F(A) = B$. This expresses the idea that C and D are structurally identical. An *equivalence of categories* relaxes this requirement. For every object $B \in \mathsf{D}$, there exists an object $A \in \mathsf{C}$ such that $F(A)$ is not necessarily equal to B, but is *isomorphic* to B.

Definition A.2.42. Categories C and D are said to be *equivalent* if there exist functors $F : \mathsf{C} \to \mathsf{D}$ and $G : \mathsf{D} \to \mathsf{C}$ such that $G \circ F \cong \mathrm{id}_{\mathsf{C}}$ and $F \circ G \cong \mathrm{id}_{\mathsf{D}}$. The functors F and G are called *quasi-inverses*, and we write $\mathsf{C} \simeq \mathsf{D}$. This entails that for every $A \in \mathsf{C}$, there is a $B \in \mathsf{D}$ with $G(B) \cong A$, and for every $B \in \mathsf{D}$, there is an $A \in \mathsf{C}$ with $F(A) \cong B$.

Example A.2.43. One of the simplest examples of an equivalence of categories is the relationship between the one-object category $\mathbf{1}$ and the category $\mathbf{2}_I$, which has two objects and a single isomorphism between them. We can visualize this as:

$$* \quad \simeq \quad a \xrightarrow{\cong} b$$

More precisely, there is a unique functor $! : \mathbf{2}_I \to \mathbf{1}$, and a functor $L : \mathbf{1} \to \mathbf{2}_I$ defined by $L(*) = a$. Clearly, the composition $! \circ L$ is equal to $\mathrm{id}_{\mathbf{1}}$, and $L \circ ! \cong \mathrm{id}_{\mathbf{2}_I}$, since both objects a and b in $\mathbf{2}_I$ are isomorphic. Thus, $\mathbf{1} \simeq \mathbf{2}_I$.

A.2.6 Adjoints

If we further weaken the notion of an equivalence of categories, we arrive at the concept of an *adjunction*.

Definition A.2.44. Given categories C and D, a pair of functors L : C $\to$ D and R : D $\to$ C form an *adjunction* $L \dashv R$ if there exists a natural isomorphism:

$$\mathrm{Hom}_{\mathsf{D}}(L(A), B) \xrightarrow{\Phi_{A,B}} \mathrm{Hom}_{\mathsf{C}}(A, R(B)).$$

One says that R is right adjoint to L, or that L is left adjoint to R. Such an adjunction is denoted by $L \dashv R$, where the turn of the symbol $\dashv$ always points to the left adjoint.

Example A.2.45. Consider the set of real numbers $\mathbb{R}$ and the set of integers $\mathbb{Z}$, each viewed as partial order categories. There is an inclusion functor $\mathrm{inc} : \mathbb{Z} \hookrightarrow \mathbb{R}$ which simply maps each integer to itself. This inclusion has a left adjoint $L : \mathbb{R} \to \mathbb{Z}$.
To determine this left adjoint L, we use the definition of an adjunction: for all $N \in \mathbb{Z}$ and $R \in \mathbb{R}$, we have a natural isomorphism:

$$\mathrm{Hom}_{\mathbb{Z}}(L(R), N) \cong \mathrm{Hom}_{\mathbb{R}}(R, \mathrm{inc}(N)).$$

Since both $\mathbb{Z}$ and $\mathbb{R}$ are partial orders, the hom-sets contain at most one morphism. Hence, this isomorphism reduces to the logical equivalence:

$$L(R) \leq N \text{ if and only if } R \leq \mathrm{inc}(N) = N.$$

Take $R = 7.27$ as an example. Then the inequality $R \leq N$ holds precisely when N is an integer greater than or equal to 7.27. That is:

$$7.27 \nleq 5, \quad 7.27 \nleq 6, \quad 7.27 \nleq 7, \quad 7.27 \leq 8, \quad 7.27 \leq 9, \quad \ldots$$

By the condition above, we must then have:

$$L(7.27) \nleq 5, \quad L(7.27) \nleq 6, \quad L(7.27) \nleq 7, \quad L(7.27) \leq 8, \quad L(7.27) \leq 9, \quad \ldots$$

From this, we conclude that $L(7.27) = 8$. In general, $L(R)$ is the least integer greater than or equal to R, *i.e.*, the ceiling function:

$$L(r) = \lceil r \rceil.$$

Thus, the inclusion functor inc has $\lceil\ \rceil$ as a left adjoint, *i.e.*, $\lceil\ \rceil \dashv \mathrm{inc}$. The unit of this adjunction is the natural transformation $\eta : \mathrm{id}_{\mathbb{R}} \Rightarrow \mathrm{inc} \circ \lceil\ \rceil$, which expresses the inequality $r \leq \lceil r \rceil$ for all $r \in \mathbb{R}$. The counit of the adjunction is the identity, since for any integer n, it holds that $\lceil N \rceil = N$.

Definition A.2.46. Let F : C $\to$ D and G : D $\to$ E be functors. It is said that G *preserves coproducts* if whenever L is a coproduct of F, then $G(L)$ is a coproduct of $G \circ F$.

Theorem A.2.47. *[127, Section 4.6] Left adjoints preserve coproducts.*

A.2.7 Monoidal categories

Definition A.2.48. A **monoid** is a triple $(M, \cdot, u)$, where M is a set equipped with a binary operation $\cdot \colon M \times M \to M$ and a distinguished element $u \in M$ called the *unit*, satisfying the following axioms for all $x, y, z \in M$:

(Associativity) $\qquad x \cdot (y \cdot z) = (x \cdot y) \cdot z,$

(Unit laws) $\qquad u \cdot x = x = x \cdot u.$

Monoidal categories are named so because they are categories equipped with an additional structure that resembles the structure of monoids.

Definition A.2.49. A *monoidal category* consists of a category C equipped with a bifunctor $\otimes : \mathsf{C} \times \mathsf{C} \to \mathsf{C}$ called *tensor product* and a distinguished object $I \in \mathsf{C}$, called *unit* together with natural isomorphisms

$$\alpha_{A,B,C} : A \otimes (B \otimes C) \to (A \otimes B) \otimes C,$$

$$\lambda_A : \mathbb{I} \otimes A \to A, \quad \rho_A : A \otimes \mathbb{I} \to A,$$

known as *associator*, *left unitor*, and *right unitor*, respectively. We will omit the subscripts when no ambiguity arises. Moreover, these natural isomorphisms are required to make the following coherence diagrams commute.

$$\begin{tikzcd}
 & (A \otimes B) \otimes (C \otimes D) \arrow[dr, "\alpha"] & \\
A \otimes (B \otimes (C \otimes D)) \arrow[ur, "\alpha"] \arrow[d, "\mathrm{id} \otimes \alpha"'] & & ((A \otimes B) \otimes C) \otimes D \\
A \otimes ((B \otimes C) \otimes D) \arrow[rr, "\alpha"'] & & (A \otimes (B \otimes C)) \otimes D \arrow[u, "\alpha \otimes \mathrm{id}"']
\end{tikzcd}$$

$$\begin{tikzcd}
A \otimes (\mathbb{I} \otimes A) \arrow[rr, "\alpha"] \arrow[dr, "\mathrm{id} \otimes \lambda_A"'] & & (A \otimes \mathbb{I}) \otimes A \arrow[dl, "\rho_A \otimes \mathrm{id}"] \\
 & A \otimes A &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{I} \otimes \mathbb{I} \arrow[rr, equal] \arrow[dr, "\lambda_I"'] & & \mathbb{I} \otimes \mathbb{I} \arrow[dl, "\rho_I"] \\
 & \mathbb{I} &
\end{tikzcd}$$

Definition A.2.50. A monoidal category is said to be *symmetric* when it is equipped with a natural isomorphism $\mathrm{sw} : A \otimes B \to B \otimes A$ known as *braiding* such that the following diagrams commute.

$$\begin{tikzcd}
A \otimes \mathbb{I} \arrow[rr, "\mathrm{sw}_{A,I}"] \arrow[dr, "\rho_A"'] & & \mathbb{I} \otimes A \arrow[dl, "\lambda_A"] \\
 & A &
\end{tikzcd}$$

$$\begin{tikzcd}
A \otimes (B \otimes C) \arrow[r, "\alpha"] \arrow[d, "\mathrm{id} \otimes \mathrm{sw}"] & (A \otimes B) \otimes C \arrow[r, "\mathrm{sw}"] & C \otimes (A \otimes B) \arrow[d, "\alpha"] \\
A \otimes (C \otimes B) \arrow[r, "\alpha"'] & (A \otimes C) \otimes B \arrow[r, "\mathrm{sw} \otimes \mathrm{id}"'] & (C \otimes A) \otimes B
\end{tikzcd}$$

Definition A.2.51. A monoidal category C is said to be *closed* if for each object A in C the functor $- \otimes A$ has a right adjoint, denoted by $A \multimap -$.

Definition A.2.52. A monoidal category C with coproducts is called *distributive* if for every object A in C the functor $- \otimes A$ preserves coproducts. Explicitly this means that the morphism,

$$[\mathrm{inl} \otimes \mathrm{id}, \mathrm{inr} \otimes \mathrm{id}] : B \otimes A \oplus C \otimes A \to (B \oplus C) \otimes A$$

is actually an isomorphism. We will denote the respective inverse by dist. Note that if C is monoidal closed then it is automatically distributive as left adjoints preserve all colimits.

Example A.2.53. Examples of monoidal closed categories with coproducts include Set and CVect. In Set, the tensor product is the cartesian product, the monoidal unit is the singleton set, the coproduct is the disjoint union, and the internal hom consists of all functions between sets. For CVect, the tensor product is the standard tensor product of complex vector spaces, the unit is the field of complex numbers $\mathbb{C}$, the coproduct is the direct sum, and the internal hom is the space of complex linear maps.

Theorem A.2.54 (*Coherence Theorem for Symmetric Monoidal Categories*)**.** *[127, Section 6.2] Any diagram in a symmetric monoidal category constructed only from associators α, unitors λ, ρ, the symmetry sw, and inverses and their composition and tensor product necessarily commutes if the two underlying permutations are the same.*

This work is financed by National Funds through the FCT - Fundação para a Ciência e a Tecnologia, I.P. (Portuguese Foundation for Science and Technology) within the project *IBEX*, with reference `PTDC/CCI-COM/4280/2021` (https://doi.org/10.54499/PTDC/CCI-COM/4280/2021).