Bruna Filipa Martins Salgado

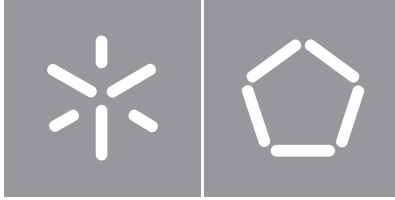# Metric λ-calculus with conditionals: quantum, probabilities and beyond

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

Copyright and Terms of Use for Third Party Work

This dissertation reports on academic work that can be used by third parties as long as the internationally accepted standards and good practices are respected concerning copyright and related rights.

This work can thereafter be used under the terms established in the license below.

Readers needing authorization conditions not provided for in the indicated licensing should contact the author through the RepositóriUM of the University of Minho.

License granted to users of this work:

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Renato Neves, for his invaluable guidance, unwavering patience, and for guiding me into this journey into the fascinating world of formal methods. This journey would not have been possible without his expertise and encouragement.

I am also deeply grateful to my parents, who have always supported and encouraged me to give my best, providing everything I needed along the way, and to my brother, who cheers me on just as much as he annoys me with his antics.

My heartfelt thanks also go to my dear friends Vitória, Margarida, Gabriela, Gonçalo, Rui, Fábio, and Miguel—especially for the cheerful lunches that left me smiling like a madwoman in the afternoons, never failing to lift my spirits during moments of endless frustration.

Juliana, Nico, and Vitor will always have my sincere gratitude for helping me whenever they could (or at least trying to), and for the occasional much-needed laugh. This gratitude also extends to another (semi-)frequent lab visitor, Alexandra.

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, more nuanced notions — such as approximate program equivalence—emerge naturally. It is within this evolving landscape that our work is situated.

Specifically, we build on the work of [36], which introduced a quantalic equational deductive system for the linear λ-calculus, along with a proof of its soundness and completeness. We extend their framework by introducing a metric equation for conditionals and proving its soundness and completeness. Syntactically, to illustrate the utility of this metric equation, we present a metric version of 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 [36].

Keywords   quantitative reasoning, λ-calculus, metric equations

Resumo

Nas últimas décadas, em ciência da computação, tem-se assistido a um esforço no sentido de nos libertarmos das rígidas amarras binárias associadas a noções como igualdade e bisimulação, explorando abordagens mais flexíveis que captem melhor as subtilezas da computação no mundo real. Por exemplo, a noção tradicional de equivalência de programas é puramente dicotómica: dois programas ou são equivalentes, ou não o são. No entanto, em muitos paradigmas computacionais, esta perspetiva binária revela-se demasiado restritiva. Em contextos que envolvem interação com o meio ou dados ruidosos, surgem naturalmente noções mais subtis, como a equivalência aproximada de programas.

É precisamente neste enquadramento que se insere a presente dissertação, ao estender o trabalho de [36], no qual foi introduzido um sistema equacional quantálico para o cálculo-λ, juntamente com as respetivas provas de correção e completude. Mais concretamente, neste trabalho propomos uma equação métrica para condicionais e demonstramos a sua correção e completude. Do ponto de vista sintático, para ilustrar a utilidade desta equação métrica, apresentamos uma versão métrica da extensionalidade do copairing. Do ponto de vista semântico, identificamos cinco categorias que satisfazem os requisitos necessários para interpretar esta equação, demonstrando assim a ampla aplicabilidade da nossa abordagem em vários domínios. Por fim, ilustramos com mais detalhe a utilização da equação métrica nos paradigmas de computação probabilística e quântica. No caso dos modelos quânticos, focamo-nos no fragmento de primeira ordem do cálculo-λ, embora sejam possíveis extensões para ordem superior através de ferramentas categóricas mais avançadas, assim como em [36].

Palavras-chave   raciocínio quantitativo, cálculo-λ, equações métricas

x

Contents

List of Figures

Acronyms

NISQ Noisy Intermediate-Scale Quantum 8

BNF Backus-Naur Form 13

CPTP Completely Positive Trace-Preserving 91

OSR Operator Sum Representation 92

miu Multiplicative involutive unital 99

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

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

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

$\Gamma \rhd v =_\epsilon w$ Metric equation-in-context. 28

V, W, R Typical symbols for vector spaces. 60

$\mathcal{F}$ Field of scalars of a vector space. 60

$\mathcal{B}(V, W)$ Vector space of all bounded linear operators from V to W. 60

$\mathcal{B}(V)$ Vector space of all bounded linear operators from V to itself. 60

$\|\cdot\|_{\mathsf{op}}$ Operator norm. 60

$V \odot W$ Algebraic tensor product of V and W. 60

$\otimes_{\mathsf{meas}}$ Product measure. 66

$\mathcal{M}\mathbb{R}$ Banach space of finite Borel measures on $\mathbb{R}$. 67

$\langle \cdot, \cdot \rangle$ Inner product. 76

$\overline{(-)}$ Complex conjugate operation. 76

$\mathcal{H}, \mathcal{K}, \mathcal{L}$ Typical symbols for Hilbert spaces. 76

$(-)^\dagger$ Adjoint operation. 77

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" [46]. 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) [125], 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* [58]. It is important to note that the term "completeness" here differs from its use in Gödel's 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* [49].

Entscheidungsproblem We have just introduced two of the main pillars of this thesis (which are closely linked to one of its principal results) — 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 [64]. 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 [26]. 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

4

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 can be used to prove properties of programming languages (*e.g.*, that a well-formed program will not crash) and as a tool in the construction of compilers [66].

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— independently addressed the Entscheidungsproblem: 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*, and proved that his definition of computability aligned with Church's. 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 system where functions are treated as first-class citizens—they can be passed as arguments, and returned, capturing a key aspect of higher-order functional programming. 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 $(\lambda x.\, x + 1)(2)$. Although λ-calculus is a powerful tool, it has a few shortcomings. For example, it allows a function to be applied to itself, as in $(\lambda x.\, x\, x)(\lambda x.\, x\, x)$, leading to non-termination. To address such issues, Church introduced the *simply typed λ-calculus* [27], which prevents self-referential paradoxes.

Typed λ-calculus In our work, we use the typed λ-calculus, a variant where each term is "labeled" by a syntactic object called a *type*. Types act as a safeguard against self-referential paradoxes while simultaneously enabling a deep correspondence with logic.

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* [52].

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. With this perspective in mind, in the 1970s, Joachim Lambek established a correspondence between cartesian closed categories and the λ-calculus [74]. That is, types correspond to objects, and terms 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 [73].

Going quantitative

Quantitative λ-calculus 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 used exactly once.

Program equivalence and its underlying theories traditionally rely on a binary notion of equivalence: two programs are either equivalent or not [121]. While this dichotomy is often sufficient for classical programming, it proves too coarse-grained for other computational paradigms. In quantum computing, for example, noise, such as decoherence, affects hardware [119, 85, 93], making it unrealistic to expect an idealized quantum algorithm to run perfectly on a quantum device; only an approximation can be observed.

To address this, [34, 36] incorporate a notion of approximate equivalence into the equational system of the linear λ-calculus by introducing, among other elements, *metric equations* [79, 80]. 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 equational system 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.

Related work Quantitative logics offer a way forward, extending beyond λ-calculus[22, 45]. 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 [79, 80, 81, 67], which explore (generalized) metric universal algebras. In simple terms, an algebra is a set equipped with a number of operations, subjected to axioms expressed as equations. In a (generalized) metric algebra, these axioms are relaxed into (generalized) metric equations rather than strict equalities. In the higher-order setting, [72], following the framework introduced by Mardare [79], investigates the problem of defining quantitative algebras that are capable of interpreting terms in higher-order calculi.

Probabilistic Programming and Quantum Computation

Probabilistic programming 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 statistical reasoning [9]. Within this context, because computers inherently operate on finite representations, exact implementation of probability distributions is infeasible, necessitating appropriate notions of approximate equivalence. As an illustrative example of this challenge, this dissertation addresses random walk approximations.

Quantum computation Quantum computing explores the principles of quantum mechanics to process information. It was first proposed in the 1980s as a means to improve the computational modeling of quantum physical systems. However, it was not until 1994 that the field gained significant attention when Peter Shor introduced an algorithm that, if implemented on a quantum computer, could pose a significant threat to the security of confidential data transmitted over the Internet [107]. While these results are revolutionary, they remain theoretical, as no quantum computer can realise them in practice. On the near horizon are **Noisy Intermediate-Scale Quantum** (**NISQ**) computers. These devices are highly susceptible to noise and errors, which calls for quantitative reasoning. In this dissertation, for example, we study the effects of introducing dephasing into the quantum teleportation protocol.

1.2 Contributions

Our contributions fall into three categories: syntax, semantics, and syntax-semantics.

Syntax

We build on the work of [36] by introducing a metric equation for conditionals. We also illustrate the utility of this metric equation via a simple example: a metric version of the copairing's extensionality. Moreover, we illustrate the usefulness of the metric equational system introduced in [36] by using it as a bridge between our type system and Boolean algebra.

Semantics

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

- The category of metric spaces Met;

- Cocompletion of a category C enriched over metric spaces;

- Category Ban of Banach spaces and short maps;

- $\mathcal{K}(\mathrm{CPS})$, the idempotent completion of the category of quantum operations (*i.e.*, completely positive, trace-nonincreasing superoperators) [99], and Cho's $(\mathrm{W}^*_{\mathrm{CPSU}})^{\mathrm{op}}$, the opposite category of $\mathrm{W}^*_{\mathrm{CPSU}}$, the category of W^*-algebras and normal, completely positive, subunital maps [23].

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 the higher-order setting are possible using more advanced categorical tools, as in [36].

As applications of our work, we investigate two computational paradigms in greater detail: probabilistic (via Ban) and quantum computation (via $\mathcal{K}(\mathrm{CPS})$ and $(\mathrm{W}^*_{\mathrm{CPSU}})^{\mathrm{op}}$). In the probabilistic setting, we use a random walk to reason about approximate equivalence. In the quantum setting, we explore two examples: quantum state discrimination and quantum teleportation.

Syntax-semantics

We prove that our extended metric equational system is sound and complete. Soundness ensures that if a metric equation $t =_\varepsilon s$ can be derived in the calculus, then the distance between all interpretations of t and s is at most ε. Completeness guarantees the converse.

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

9

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 illustrate this idea in an interlude on booleans. Additionally, we prove several folklore results about conditionals that, to our knowledge, are missing from the literature. In Chapter 3, we introduce a metric equation for conditionals, prove soundness and completeness of the resulting system, and present a few models in this setting, along with an illustrative syntactic example. Then, in Chapter 4 and Chapter 5, we focus on reasoning about higher-order probabilistic and first-order quantum programs, respectively, including both the necessary background and illustrative examples for each domain. The thesis concludes with directions for future work in Chapter 6. The reader is assumed to be familiar with the basics of category theory, Banach spaces, and topology. For a more detailed study, the reader may consult [6] for category theory, [70, Chapters 1–2] for Banach spaces, and [2, Chapters 2 and 5] for topology. Alternatively, an overview of these areas' relevant concepts and results is provided in Appendix A.

Chapter 2

Metric Lambda Calculus

This chapter introduces the metric lambda calculus as presented in [36], drawing also from [77, 31, 100]. 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.*, [7].

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 $(\lambda x. x\, x)(\lambda x. x\, x)$, can result 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 (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 **Backus-Naur Form** (**BNF**) below.

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

Note that this is an inductive definition. 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 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 $\mathsf{pm}\ v\ \mathsf{to}\ x \otimes y.w$ is the pattern-matching construct that deconstructs a tensor product into components x and y. The term $v\ \mathsf{to}\ *.w$ is used to discard a variable v (of the unit type). The terms $\mathsf{inl}(v)$ and $\mathsf{inr}(v)$ 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 $(\lambda x.\, v)w$.

2.2.2 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:

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

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

2.2.3 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 essential 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.

2.2.4 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 $\lambda x.\ (\lambda z.\, xz)x$. The proper thing to do is to rename the bound variable *before* the substitution:

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

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.

$$
\begin{aligned}
x[w/x] &= w \\
y[w/x] &= y && \text{if } x \neq y \\
*[w/x] &= * \\
(\lambda x.v)[w/x] &= \lambda x.v \\
(\lambda y.v)[w/x] &= \lambda y.v[w/x] && \text{if } x \neq y \text{ and } y \notin FV(w) \\
\lambda y.v[w/x] &= \lambda y'.v\{y'/y\}[w/y] && \text{if } x \neq y, y \in FV(w), \\
&&& \text{and } y' \text{ is fresh} \\
(v\,u)[w/x] &= v[w/x]\,u[w/x] \\
(f(v_1,\ldots,v_n))[w/x] &= f(v_1[w/x],\ldots,v_n[w/x]) \\
(v \otimes u)[w/x] &= (v[w/x] \otimes u[w/x]) \\
(\text{pm } v \text{ to } y \otimes z.u)[w/x] &= \text{pm } v[w/x] \text{ to } y \otimes z.u[w/x] && \text{if } y \notin FV(w), z \notin FV(w) \\
(\text{pm } v \text{ to } y \otimes z.u)[w/x] &= \text{pm } v[w/x] \text{ to } y' \otimes z. && \text{if } y \in FV(w), z \notin FV(w), \\
&\quad u\{y'/y\}[w/x] && \text{and } y' \text{ is fresh} \\
(\text{pm } v \text{ to } y \otimes z.u)[w/x] &= \text{pm } v[w/x] \text{ to } y \otimes z'. && \text{if } y \notin FV(w), z \in FV(w), \\
&\quad u\{z'/z\}[w/x] && \text{and } z' \text{ is fresh} \\
(\text{pm } v \text{ to } y \otimes z.u)[w/x] &= \text{pm } v[w/x] \text{ to } y' \otimes z'. && \text{if } y \in FV(w), z \in FV(w),
\end{aligned}
$$

$$u\{y'/y\}\{z'/z\}[w/x] \qquad \text{and } y', z' \text{ are fresh}$$

$$(v \text{ to } *.u)[w/x] = v[w/x] \text{ to } *.u[w/x]$$

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

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

$$\mathsf{case}\ v \left\{ \begin{array}{l} \mathsf{inl}(y) \Rightarrow p; \\ \mathsf{inr}(z) \Rightarrow q \end{array} \right\} [w/x] = \mathsf{case}\ v[w/x] \left\{ \begin{array}{l} \mathsf{inl}(y) \Rightarrow \\ p[w/x]; \\ \mathsf{inr}(z) \Rightarrow \\ q[w/x] \end{array} \right\} \quad \text{if } y \notin FV(w), z \notin FV(w)$$

$$(\ldots)$$

$$\mathsf{case}\ v \left\{ \begin{array}{l} \mathsf{inl}(y) \Rightarrow p; \\ \mathsf{inr}(z) \Rightarrow q \end{array} \right\} [w/x] = \begin{array}{l} \mathsf{case}\ v[w/x] \\ \left\{ \begin{array}{l} \mathsf{inl}(y') \Rightarrow p\{y'/y\} \\ \{z'/z\}[w/x]; \\ \mathsf{inr}(z') \Rightarrow q\{y'/y\} \\ \{z'/z\}[w/x] \end{array} \right\} \end{array} \quad \begin{array}{l} \text{if } y \in FV(w), z \in FV(w), \\ \text{and } y', z' \text{ are fresh} \end{array}$$

2.2.5 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 BNF grammar of types for affine lambda calculus is as follows:

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

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.

2.2.6 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 $x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n \triangleright v : \mathbb{A}$ (where $n \geq 1$), which asserts that the term v is a well-typed term of type $\mathbb{A}$ under the assumption that each variable x_i has type $\mathbb{A}_i$, for $1 \leq i \leq n$. The list $x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n$ of typed variables is called the *typing context* of the judgment, and it might be empty. Each variable x_i (where $1 \leq i \leq n$) must occur at most once in $x_1, \ldots, x_n$. 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.* $\Gamma, \Delta, \ldots$) 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 from 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, $\Gamma_1, \ldots, \Gamma_n$, preserving the relative order of variables within each Γ_i [108]. 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 $\Gamma_1, \ldots, \Gamma_n$ is denoted as $\mathsf{Sf}(\Gamma_1; \ldots; \Gamma_n)$. A valid typing derivation is constructed using the inductive rules shown in Figure 1.

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

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

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

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

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

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

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

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

$$1 \quad x : \mathbb{A} \otimes \mathbb{B} \triangleright y : \mathbb{A} \otimes \mathbb{B} \qquad\qquad\qquad\qquad\quad \text{(hyp)}$$

$$2 \quad b : \mathbb{B} \triangleright b : \mathbb{B} \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{(hyp)}$$

$$3 \quad a : \mathbb{A} \triangleright a : \mathbb{A} \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{(hyp)}$$

$$4 \quad b : \mathbb{B}, a : \mathbb{A} \triangleright b \otimes a : \mathbb{B} \otimes \mathbb{A} \qquad\qquad\qquad (2, 3, \otimes_i)$$

$$5 \quad x : \mathbb{A} \otimes \mathbb{B} \triangleright \mathsf{pm}\, x \text{ to } a \otimes b.b \otimes a : \mathbb{B} \otimes \mathbb{A} \qquad (1, 4, \otimes_e)$$

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

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

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

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

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

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

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

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

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

$$z : (\mathbb{A} \oplus \mathbb{B}) \otimes \mathbb{D} \triangleright \text{pm } z \text{ to } x \otimes d. \text{ case } x \left\{ \begin{array}{l} \text{inl}_{\mathbb{B}}(a) \Rightarrow \text{inl}_{\mathbb{B} \otimes \mathbb{D}}(a \otimes d); \\ \text{inr}_{\mathbb{A}}(b) \Rightarrow \text{inr}_{\mathbb{A} \otimes \mathbb{D}}(b \otimes d) \end{array} \right\} : (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})$$

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

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

2 $d : \mathbb{D} \triangleright x : \mathbb{D}$ (hyp)

3 $d : \mathbb{D}, a : \mathbb{A} \triangleright \text{inl}_{\mathbb{B} \otimes \mathbb{D}}(a \otimes d) : (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})$ $(1, 2, \text{inl})$

4 $b : \mathbb{B} \triangleright b : \mathbb{B}$ (hyp)

5 $d : \mathbb{D}, b : \mathbb{B} \triangleright \text{inr}_{\mathbb{A} \otimes \mathbb{D}}(b \otimes d) : (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})$ $(2, 4, \text{inr})$

6 $x : \mathbb{A} \oplus \mathbb{B} \triangleright x : \mathbb{A} \otimes \mathbb{B}$ (hyp)

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

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

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

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 following Curry-style typing judgment:

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

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 [35, 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 $\Gamma, \Gamma', \Delta, \Delta'$ 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,

$$\mathsf{te}(E) \simeq_\pi \mathsf{te}(\Gamma, \Delta)$$
$$\simeq_\pi \mathsf{te}(\Gamma', \Delta')$$
$$\simeq_\pi \mathsf{te}(E')$$

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 $\Gamma, \Gamma', \Delta, \Delta'$ 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. $\square$

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

Proof. Once again, these properties are established in [36, 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 $E_1, x : \mathbb{A}, y : \mathbb{B}, E_2 \triangleright \mathsf{case}\ v\ \{\mathrm{inl}_\mathbb{F}(a) \Rightarrow w;\ \mathrm{inr}_\mathbb{E}(b) \Rightarrow u\} : \mathbb{D}$, 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,

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

(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 $E, x : \mathbb{A} \triangleright \mathsf{case}\ v\ \{\mathrm{inl}_\mathbb{D}(a) \Rightarrow$

$w; \mathrm{inr}_{\mathbb{E}}(b) \Rightarrow u\} : \mathbb{B}$ 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. $\qquad\square$

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

2.2.8 Equations-in-context

The simply typed lambda calculus is a formal language that captures operations like the application of a function to an argument and the elimination of variables. To express these operations, there is a set of 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.

$$
\begin{array}{cl}
(\beta) & (\lambda x : \mathbb{A}.\, v)\, w = v[w/x] \qquad (\eta) \qquad \lambda x : \mathbb{A}.\, (v\, x) = v \\[4pt]
(\beta_{\mathbb{I}_e}) & * \text{ to } *.\, v = v \qquad\qquad (\eta_{\mathbb{I}_e}) \qquad v \text{ to } *.\, w[*/z] = w[v/z] \\[4pt]
(\beta_{\otimes_e}) & \mathsf{pm}\, v \otimes w \text{ to } x \otimes y.\, u = u[v/x, w/y] \\[4pt]
(\eta_{\otimes_e}) & \mathsf{pm}\, v \text{ to } x \otimes y.\, u[x \otimes y/z] = u[v/z] \\[4pt]
(c_{\mathbb{I}_e}) & u[v \text{ to } *.\, w/z] = v \text{ to } *.\, u[w/z] \\[4pt]
(c_{\otimes_e}) & u[\mathsf{pm}\, v \text{ to } x \otimes y.\, w/z] = \mathsf{pm}\, v \text{ to } x \otimes y.\, u[w/z] \\[4pt]
(\beta_{case}^{inl}) & \mathsf{case}\, \mathrm{inl}_{\mathbb{B}}(v)\, \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow w;\, \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow u\} = w[v/x] \\[4pt]
(\beta_{case}^{inr}) & \mathsf{case}\, \mathrm{inl}_{\mathbb{B}}(v)\, \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow w;\, \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow u\} = u[v/y] \\[4pt]
(\eta_{case}) & \mathsf{case}\, v\, \{\mathrm{inl}_{\mathbb{B}}(y) \Rightarrow w[\mathrm{inl}_{\mathbb{B}}(y)/x];\, \mathrm{inr}_{\mathbb{A}}(z) \Rightarrow w[\mathrm{inr}_{\mathbb{A}}(z)/x]\} = w[v/x]
\end{array}
$$

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

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

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

Applying equation β , we have:

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

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 congruence containing Ax, the equations presented in , and closed under exchange and substitution (). 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 :

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

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 $= \mathrm{inr}(*)$ [100]. 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},

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

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

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

Disjunction $(a, b) \triangleq a : \mathbb{I} \oplus \mathbb{I}, b : \mathbb{I} \oplus \mathbb{I} \rhd \mathsf{case}\ a \begin{cases} \mathrm{inl}_{\mathbb{I}}(x) \Rightarrow x\ \mathsf{to}\ *\ .\ \mathsf{dis}(b)\ \mathsf{to}\ *\ .\mathrm{inl}_{\mathbb{I}}(*); \\ \mathrm{inr}_{\mathbb{I}}(y) \Rightarrow y\ \mathsf{to}\ *\ .\ b \end{cases}$

Moreover, negation can be expressed by the following program:

Negation $(a) \triangleq a : \mathbb{I} \oplus \mathbb{I} \rhd \mathsf{case}\ a \begin{cases} \mathrm{inl}_{\mathbb{I}}(x) \Rightarrow \mathrm{inr}_{\mathbb{I}}(x); \\ \mathrm{inr}_{\mathbb{I}}(y) \Rightarrow \mathrm{inl}_{\mathbb{I}}(y) \end{cases}$

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

$$\Gamma, \Delta \rhd \textbf{Conjunction}\ (v, w) \triangleq \Gamma, \Delta \rhd \textbf{Conjunction}\ (a, b)[v/a, w/b]$$

$$\Gamma, \Delta \rhd \textbf{Disjunction}\ (v, w) \triangleq \Gamma, \Delta \rhd \textbf{Disjunction}\ (a, b)[v/a, w/b] \qquad (2.1)$$

$$\Gamma \rhd \textbf{Negation}\ (a) \triangleq \Gamma \rhd \textbf{Negation}\ (a)[v/a]$$

The following property will enable us to verify that the programs we defined above satisfy the desired properties in a systematic and straightforward manner.

Extensionality of the copairing

Proposition 2.2.14. *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.,*

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

Proof. Using the β-equation we have

$$\begin{cases} (\lambda x.v)\, \mathsf{inl}(y) = (\lambda x.w)\, \mathsf{inl}(y) \\ (\lambda x.v)\, \mathsf{inr}(z) = (\lambda x.w)\, \mathsf{inr}(z) \end{cases} \iff \begin{cases} v\,[\mathsf{inl}(y)/x] = w\,[\mathsf{inl}(y)/x] \\ v\,[\mathsf{inr}(z)/x] = w\,[\mathsf{inr}(z)/x] \end{cases} \tag{2.2}$$

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

$$v = \mathsf{case}\, x \left\{ \mathrm{inl}(y) \Rightarrow v\,[\mathsf{inl}(y)/x]; \mathrm{inl}(z) \Rightarrow v\,[\mathsf{inr}(z)/x] \right\} \tag{η_{case}}$$
$$= \mathsf{case}\, x \left\{ \mathrm{inl}(y) \Rightarrow w\,[\mathsf{inl}(y)/x]; \mathrm{inl}(z) \Rightarrow w\,[\mathsf{inr}(z)/x] \right\} = w \quad (\text{Equation 2.2}, \eta_{case})$$

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

Properties

Next, we will show that the programs we have defined verify certain properties of their name-sake 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 2.2.15. $\mathrm{inl}(*)$ *acts as the neutral element for conjunction, whereas* $\mathrm{inr}(*)$ *serves as the absorbing element, i.e., for $\Gamma \rhd v : \mathbb{I} \oplus \mathbb{I}$*

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

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

Conjunction $(\mathrm{inl}(*), v)$

$\triangleq \mathsf{case}\ \mathrm{inl}(*)\ \{\mathrm{inl}(x) \Rightarrow x\ \mathsf{to}\ * . v;\ \mathrm{inr}(y) \Rightarrow y\ \mathsf{to}\ * . \mathsf{dis}(w')\ \mathsf{to}\ * .\mathrm{inr}(*)\}$

$= *\ \mathsf{to}\ * . v$ 　　　　　　　　　　　　　　　　　　　　　　　(β_{case}^{inl})

$= v$ 　　　　　　　　　　　　　　　　　　　　　　　　　　　　　$(\beta_{\mathbb{I}_e})$

For the second equality, by the extensionality of the coparing it suffices to prove that

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

For the first equation, we reason as follows:

Conjunction $(\mathrm{inr}(*), \mathrm{inl}(z))$

$\triangleq \mathsf{case}\ \mathrm{inr}(*) \begin{cases} \mathrm{inl}(x) \Rightarrow x\ \mathsf{to}\ * . \mathrm{inl}(z); \\ \mathrm{inr}(y) \Rightarrow y\ \mathsf{to}\ * . \mathsf{dis}(\mathrm{inl}(z))\ \mathsf{to}\ * .\mathrm{inr}(*) \end{cases}$

$= *\ \mathsf{to}\ * . \mathsf{dis}(\mathrm{inl}(z))\ \mathsf{to}\ * .\mathrm{inr}(*)$ 　　　　　　　　　(β_{case}^{inr})

$= \mathsf{dis}(\mathrm{inl}(z))\ \mathsf{to}\ * .\mathrm{inr}(*)$ 　　　　　　　　　　　　　$(\beta_{\mathbb{I}_e})$

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 lemma above, and equations ax_{dis}, β_{case}^{inr}, and $\beta_{\mathbb{I}_e}$.

Proposition 2.2.16. *The conjunction of two terms is commutative , i.e., for* $\Gamma \triangleright v : \mathbb{I} \oplus \mathbb{I}$ *and* $\Delta \triangleright w : \mathbb{I} \oplus \mathbb{I}$

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

Proof. Once again, by the extensionality of the 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.

Conjunction $(\mathrm{inl}(c), \mathrm{inr}(d))$

$$\triangleq \mathsf{case}\, \mathrm{inl}(c) \left\{ \begin{array}{l} \mathrm{inl}(x) \Rightarrow x \,\mathsf{to}\, *\,.\,\mathrm{inr}(d); \\[4pt] \mathrm{inr}(y) \Rightarrow y \,\mathsf{to}\, *\,.\,\mathsf{dis}(\mathrm{inr}(d)) \,\mathsf{to}\, *\,.\mathrm{inr}(*) \end{array} \right\}$$

$$= c \,\mathsf{to}\, *\,.\,\mathrm{inr}(*) \qquad\qquad\qquad (\beta^{inl}_{case})$$

$$= \mathsf{dis}(\mathrm{inl}(c)) \,\mathsf{to}\, *\,.\mathrm{inr}(*) \qquad\qquad\qquad (ax_{\mathsf{dis}}, \beta^{inl}_{case})$$

$$= d \,\mathsf{to}\, *\,.\,\mathsf{dis}(\mathrm{inl}(c)) \,\mathsf{to}\, *\,.\mathrm{inr}(*) \qquad\qquad\qquad (\eta_{\mathbb{I}_e})$$

$$= \mathsf{case}\, \mathrm{inr}(d) \left\{ \begin{array}{l} \mathrm{inl}(x) \Rightarrow x \,\mathsf{to}\, *\,.\,\mathrm{inl}(c); \\[4pt] \mathrm{inr}(y) \Rightarrow y \,\mathsf{to}\, *\,.\,\mathsf{dis}(\mathrm{inl}(c)) \,\mathsf{to}\, *\,.\mathrm{inr}(*) \end{array} \right\} \qquad (\beta^{inr}_{case})$$

$$\triangleq \textbf{Conjunction}\, (\mathrm{inr}(d), \mathrm{inl}(c))$$

$\square$

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

$$\Gamma \rhd \textbf{Negation}(\textbf{Negation}(v)) = v.$$

Proof. By the extensionality of the 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.

Negation(**Negation**$(\mathrm{inl}(z))$)

$$\triangleq \textbf{Negation}(\mathsf{case}\, \mathrm{inl}(z)\, \{\mathrm{inl}(x) \Rightarrow \mathrm{inr}(x);\, \mathrm{inr}(y) \Rightarrow \mathrm{inl}(y)\})$$

$$= \textbf{Negation}(\mathrm{inr}(z)) \qquad\qquad\qquad (\beta^{inl}_{case})$$

$$= \mathrm{inl}(z) \qquad\qquad\qquad (\beta^{inr}_{case})$$

$\square$

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 $\Gamma \rhd v : \mathbb{I} \oplus \mathbb{I}$ and $\Delta \rhd w : \mathbb{I} \oplus \mathbb{I}$,*

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

Proof. Once again, by the extensionality of the 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.

Disjunction(**Negation**$(\mathrm{inl}(c))$, **Negation**$(\mathrm{inr}(d))$)

$$\triangleq \mathsf{case}\ \mathsf{case}\ \mathrm{inl}(c) \begin{Bmatrix} \mathrm{inl}(a) \Rightarrow \mathrm{inr}(a); \\ \mathrm{inl}(b) \Rightarrow \mathrm{inl}(b) \end{Bmatrix} \begin{Bmatrix} \mathrm{inl}(x) \Rightarrow \dots; \\ \mathrm{inl}(y) \Rightarrow y\ \mathsf{to}\ *\,.\,\mathbf{Negation}(\mathrm{inr}(d)) \end{Bmatrix}$$

$$= c\ \mathsf{to}\ *\,.\,\mathsf{case}\ \mathrm{inr}(d) \begin{Bmatrix} \mathrm{inl}(x) \Rightarrow \mathrm{inr}(x); \\ \mathrm{inl}(y) \Rightarrow \mathrm{inl}(y) \end{Bmatrix} \qquad\qquad (\beta_{case}^{inl}, \beta_{case}^{inr})$$

$$= \mathsf{case}\ c\ \mathsf{to}\ *\,.\mathrm{inr}(d) \begin{Bmatrix} \mathrm{inl}(x) \Rightarrow \mathrm{inr}(x); \\ \mathrm{inl}(y) \Rightarrow \mathrm{inl}(y) \end{Bmatrix} \qquad\qquad (c_{\mathbb{I}_e})$$

$$= \mathsf{case}\ \mathsf{case}\ \mathrm{inl}(c) \begin{Bmatrix} \mathrm{inl}(x) \Rightarrow x\ \mathsf{to}\ *\,.\,\mathrm{inr}(d); \\ \mathrm{inr}(y) \Rightarrow y\ \mathsf{to}\ *\,.\,\mathsf{dis}(\mathrm{inr}(d))\ \mathsf{to}\ *\,.\mathrm{inr}(*) \end{Bmatrix} \begin{Bmatrix} \dots; \\ \dots \end{Bmatrix} \qquad (\beta_{case}^{inl})$$

$$\triangleq \mathbf{Negation}(\mathbf{Conjunction}(\mathrm{inl}(c), \mathrm{inr}(d)))$$

$\square$

The remaining properties such can be proven through similar reasoning.

2.2.10 Metric equational system

Metric equations [79, 80] are a strong candidate for reasoning about approximate program equivalence. These equations take the form of $\Gamma \triangleright v =_\epsilon w$, meaning terms v and w 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 $\Gamma \triangleright v = w : \mathbb{A}$ in Figure 2, which in this setting abbreviate $\Gamma \triangleright w =_0 v : \mathbb{A}$, are also part of the metric equational system.

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

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

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

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

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

Figure 3: Metric equational system

Here, $\mathsf{perm}(\Gamma)$ 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 $v =_\infty w$. The rule that follows conveys that if the maximum distance between two terms v and w is q, and the maximum distance between terms v' and w' is r, then the maximum distance between the tensor products $v \otimes v'$ and $w \otimes w'$ is $q + r$, *i.e.*, the distances compound additively. The remaining rules follow similar reasoning.

Example 2.2.19. To illustrate the usefulness of these equations, consider the program **SwapTensorf** 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:

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

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

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

Consider we have the axiom $f(a) =_\varepsilon f^\varepsilon(a)$. Then, it is possible to show that **SwapTensorf** $=_\varepsilon$ **SwapTensorf**$^\varepsilon$ using our metric equational system. The proof is as follows.

$$1 \quad f(a)^\varepsilon =_\varepsilon f(a)$$

$$2 \quad b =_0 b \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{refl})$$

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

$$4 \quad x \otimes y =_0 x \otimes y \qquad\qquad\qquad\qquad\qquad (\text{refl})$$

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

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

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

holds, 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

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

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

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

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 $A_1, ..., A_n \in \mathsf{C}$, the n-tensor $((A_1 \otimes A_2) \otimes \ldots) \otimes A_n$ 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.

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

$$\overline{[\![x : \mathbb{A} \triangleright x : \mathbb{A}]\!] = \mathrm{id}_{[\![\mathbb{A}]\!]}} \qquad\qquad \overline{[\![- \triangleright * : \mathbb{I}]\!] = \mathrm{id}_{[\![\mathbb{I}]\!]}}$$

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

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

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

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

$$\frac{[\![\Gamma \triangleright v : \mathbb{A}]\!] = m}{[\![\Gamma \triangleright \mathsf{inl}_{\mathbb{B}}(v) : \mathbb{A} \oplus \mathbb{B}]\!] = \mathrm{inl} \cdot m} \qquad\qquad \frac{[\![\Gamma \triangleright v : \mathbb{B}]\!] = m}{[\![\Gamma \triangleright \mathsf{inr}_{\mathbb{A}}(v) : \mathbb{A} \oplus \mathbb{B}]\!] = \mathrm{inr} \cdot m}$$

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

Figure 4: Judgment interpretation

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

$$[\![\mathsf{ax}]\!] : \qquad [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma_1, \ldots, \Gamma_n]\!] \xrightarrow{\mathsf{sp}_{\Gamma; \Delta}} [\![\Gamma_1]\!] \otimes \ldots \otimes [\![\Gamma_n]\!]$$
$$\xrightarrow{m_1 \otimes \ldots \otimes m_n} [\![\mathbb{A}_1]\!] \otimes \ldots \otimes [\![\mathbb{A}_n]\!] \xrightarrow{[\![f]\!]} [\![\mathbb{A}]\!]$$

$$[\![\mathsf{hyp}]\!] : \qquad [\![\mathbb{A}]\!] \xrightarrow{\mathrm{id}_{[\![\mathbb{A}]\!]}} [\![\mathbb{A}]\!]$$

$$[\![\mathbb{I}_i]\!] : \qquad [\![\mathbb{I}]\!] \xrightarrow{\mathrm{id}_{[\![\mathbb{I}]\!]}} [\![\mathbb{I}]\!]$$

$$[\![\otimes_e]\!] : \qquad [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma, \Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{m \otimes \mathrm{id}} ([\![\mathbb{A}]\!] \otimes [\![\mathbb{B}]\!]) \otimes [\![\Delta]\!]$$

$$\xrightarrow{\mathsf{sw}} [\![\Delta]\!] \otimes ([\![\mathbb{A}]\!] \otimes [\![\mathbb{B}]\!]) \xrightarrow{\alpha} ([\![\Delta]\!] \otimes [\![\mathbb{A}]\!]) \otimes [\![\mathbb{B}]\!] \xrightarrow{\mathsf{jn}_{\Delta;\mathbb{A};\mathbb{B}}} [\![\Delta, \mathbb{A}, \mathbb{B}]\!]$$

$$\xrightarrow{n} [\![\mathbb{D}]\!]$$

$$[\![\otimes_i]\!] : \qquad [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma, \Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{m \otimes n} [\![\mathbb{A}]\!] \otimes [\![\mathbb{B}]\!]$$

$$[\![\mathbb{I}_e]\!] : \qquad [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma, \Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{m \otimes \mathrm{id}} [\![\mathbb{I}]\!] \otimes [\![\Delta]\!] \xrightarrow{\lambda} [\![\Delta]\!] \xrightarrow{n} [\![\mathbb{A}]\!]$$

$$[\![\multimap_i]\!] : \qquad [\![\Gamma]\!] \xrightarrow{\overline{m \cdot \mathsf{jn}_{\Gamma;\mathbb{A}}}} [\![\mathbb{A}]\!] \multimap [\![\mathbb{B}]\!] \qquad\qquad ([\![\Gamma]\!] \otimes [\![\mathbb{A}]\!] \xrightarrow{\mathsf{jn}_{\Gamma;\mathbb{A}}} [\![\Gamma, \mathbb{A}]\!] \xrightarrow{m} [\![\mathbb{B}]\!])$$

$$[\![\multimap_e]\!] : \qquad [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma, \Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{m \otimes n} ([\![\mathbb{A}]\!] \multimap [\![\mathbb{B}]\!]) \otimes [\![\mathbb{A}]\!] \xrightarrow{\mathsf{app}} [\![\mathbb{B}]\!]$$

$$[\![\mathsf{inl}]\!] : \qquad [\![\Gamma]\!] \xrightarrow{m} [\![\mathbb{A}]\!] \xrightarrow{\mathrm{inl}} [\![\mathbb{A} \oplus \mathbb{B}]\!]$$

$$[\![\mathsf{inr}]\!] : \qquad [\![\Gamma]\!] \xrightarrow{m} [\![\mathbb{B}]\!] \xrightarrow{\mathrm{inr}} [\![\mathbb{A} \oplus \mathbb{B}]\!]$$

$$[\![\mathsf{case}]\!] : \qquad [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma, \Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{b \otimes \mathrm{id}} ([\![\mathbb{A}]\!] \oplus [\![\mathbb{B}]\!]) \otimes [\![\Delta]\!]$$

$$\xrightarrow{\mathrm{dist}} ([\![\mathbb{A}]\!] \otimes [\![\Delta]\!]) \oplus ([\![\mathbb{B}]\!] \otimes [\![\Delta]\!])$$

$$\xrightarrow{\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathsf{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathsf{sw}} [\![\Delta, \mathbb{A}]\!] \oplus [\![\Delta, \mathbb{B}]\!] \xrightarrow{[p,q]} [\![\mathbb{D}]\!]$$

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

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

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

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

Proof. This lemma is proved in [36, Lemma 2.2] for the lambda calculus without conditionals, so we only need to address the conditional cases.

We begin with the exchange property. The rules involving injections are straightforward. As for the rule case, we distinguish between the scenarios where both variables ($x : \mathbb{A}$ and $y : \mathbb{B}$) are in Γ, both are in Δ, or they are distributed across Γ and Δ. We begin with the first case.

$$[\![\Gamma, x, y, \Delta \triangleright \mathsf{case}\ v\ \{\mathsf{inl}(a) \Rightarrow w;\ \mathsf{inr}(b) \Rightarrow u\}]\!]$$

$$\triangleq [[\![w]\!], [\![u]\!]] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot ([\![v]\!] \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$$

$$= [[\![w]\!], [\![u]\!]] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot ([\![v]\!] \cdot \mathsf{exch} \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \qquad \text{(Induction)}$$

$$= \ldots \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot (\mathsf{exch} \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$$

$$= \ldots \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \cdot \mathsf{exch} \qquad\qquad\qquad\qquad\qquad \text{(Coherence)}$$

$$\triangleq \llbracket \Gamma, y, x, \Delta \triangleright \mathsf{case}\ v\ \{\mathsf{inl}(a) \Rightarrow w;\ \mathsf{inr}(b) \Rightarrow u\} \rrbracket \cdot \mathsf{exch}$$

Let us now focus on the second case, *i.e.* both variables live in Δ.

$$\llbracket \Gamma, x, y, \Delta \triangleright \mathsf{case}\ v\ \{\mathsf{inl}(a) \Rightarrow w;\ \mathsf{inr}(b) \Rightarrow u\} \rrbracket$$

$$\triangleq [\llbracket w \rrbracket, \llbracket u \rrbracket] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$$

$$= [\llbracket w \rrbracket \cdot \mathsf{exch}, \llbracket u \rrbracket \cdot \mathsf{exch}] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \qquad \text{(Induction)}$$

$$= [\llbracket w \rrbracket, \llbracket u \rrbracket] \cdot (\mathsf{exch} \oplus \mathsf{exch}) \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \ldots \qquad \text{(Coproduct laws)}$$

$$= [\llbracket w \rrbracket, \llbracket u \rrbracket] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot (\mathrm{id} \otimes \mathsf{exch} \oplus \mathrm{id} \otimes \mathsf{exch}) \cdot \mathrm{dist} \cdot \ldots \qquad \text{(Coherence)}$$

$$= \ldots \cdot \mathrm{dist} \cdot (\mathrm{id} \otimes \mathsf{exch}) \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \qquad\qquad\qquad \text{(Naturality)}$$

$$= \ldots \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \cdot \mathsf{exch} \qquad\qquad\qquad\qquad \text{(Coherence)}$$

$$\triangleq \llbracket \Gamma, y, x, \Delta \triangleright \mathsf{case}\ v\ \{\mathsf{inl}(a) \Rightarrow w;\ \mathsf{inr}(b) \Rightarrow u\} \rrbracket \cdot \mathsf{exch}$$

The proof the for the third case follows directly from the coherence theorem for symmetric monoidal categories.

Regarding the substitution rule, once again the cases involving the injections follow directly by induction on the derivation tree. For the rule (case), we distinguish between the scenarios where the variable x is in Γ or in Δ. We start with the first case.

$$\llbracket E, Z \triangleright \mathsf{case}\ v\ \{\mathsf{inl}(a) \Rightarrow w;\ \mathsf{inr}(b) \Rightarrow u\}[t/x] \rrbracket$$

$$\triangleq [\llbracket w \rrbracket, \llbracket u \rrbracket] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v[t/x] \rrbracket \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$$

$$= [\llbracket w \rrbracket, \llbracket u \rrbracket] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot ((\llbracket v \rrbracket \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp}) \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \quad \text{(Induction)}$$

$$= \ldots \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot (\mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp} \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$$

$$= \ldots \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot (\mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp} \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \cdot \mathsf{jn} \cdot \mathsf{sp}$$

$$= \ldots \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot (\mathsf{jn} \cdot \mathsf{sp} \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp} \qquad \text{(Naturality)}$$

$$= \ldots \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp} \qquad\qquad\qquad \text{(Coherence)}$$

$$\triangleq \llbracket E, x \triangleright \mathsf{case}\ v\ \{\mathsf{inl}(x) \Rightarrow w;\ \mathsf{inr}(y) \Rightarrow u\} \rrbracket \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp}$$

With respect to the second case, *i.e.*, x lives in Δ, we have:

$\llbracket E, Z \triangleright \mathsf{case}\ v\ \{\mathsf{inl}(a) \Rightarrow w;\ \mathsf{inr}(b) \Rightarrow u\}[t/x]\rrbracket$

$\triangleq [\llbracket \Delta, Z, a \triangleright w[t/x]\rrbracket, \llbracket \Delta, Z, b \triangleright u[t/x]\rrbracket] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathrm{id})$

$\quad \cdot \mathsf{sp} \cdot \mathsf{sh}$

$= [\llbracket \Delta, a, Z \triangleright w[t/x]\rrbracket, \llbracket \Delta, b, Z \triangleright u[t/x]\rrbracket] \cdot (\mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw}) \cdot \ldots \quad$ (Exch., Cop.)

$= [\llbracket w \rrbracket \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp}, \llbracket u \rrbracket \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp}] \cdot (\mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{exch} \quad$ (Induction)

$\quad \cdot \mathsf{jn} \cdot \mathrm{sw}) \cdot \ldots$

$= \big[\llbracket w \rrbracket \cdot \mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \cdot (\mathrm{id} \otimes \mathsf{jn}) \cdot (\mathrm{sw} \otimes \mathrm{id}) \cdot (\mathrm{id} \otimes \mathsf{sp}) \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp}, \quad$ (Coherence)

$\quad\quad \llbracket u \rrbracket \cdot \mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \cdot (\mathrm{id} \otimes \mathsf{jn}) \cdot (\mathrm{sw} \otimes \mathrm{id}) \cdot (\mathrm{id} \otimes \mathsf{sp}) \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp}\big] \cdot \ldots$

$= \big[\llbracket w \rrbracket \cdot \mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \cdot (\mathrm{id} \otimes \mathsf{jn}) \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot (\mathrm{sw} \otimes \mathrm{id}) \cdot (\mathrm{id} \otimes \mathsf{sp}) \cdot \mathsf{sp}, \quad$ (Naturality)

$\quad\quad \llbracket u \rrbracket \cdot \mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \cdot (\mathrm{id} \otimes \mathsf{jn}) \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot (\mathrm{sw} \otimes \mathrm{id}) \cdot (\mathrm{id} \otimes \mathsf{sp}) \cdot \mathsf{sp}\big] \cdot \ldots$

$\quad \cdot (\mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw}) \cdot \ldots$

$= [\llbracket w \rrbracket, \llbracket u \rrbracket] \cdot \big(\mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \cdot (\mathrm{id} \otimes \mathsf{jn}) \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot (\mathrm{id} \otimes \mathsf{sp}) \oplus \mathsf{exch} \cdot \mathsf{jn} \cdot \mathrm{sw} \quad$ (Coherence,

$\quad \cdot (\mathrm{id} \otimes \mathsf{jn}) \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot (\mathrm{id} \otimes \mathsf{sp})\big) \cdot \ldots \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ Cop. laws)

$= [\llbracket \Delta, x, a \triangleright w \rrbracket, \llbracket \Delta, x, b \triangleright u \rrbracket] \cdot \big(\mathsf{jn} \cdot \mathrm{sw} \cdot (\mathrm{id} \otimes (\mathsf{jn} \cdot \mathrm{id} \otimes \llbracket t \rrbracket \cdot \mathsf{sp})) \oplus \mathsf{jn} \cdot \mathrm{sw} \quad$ (Exchange,

$\quad \cdot (\mathrm{id} \otimes (\mathsf{jn} \cdot \mathrm{id} \otimes \llbracket t \rrbracket \cdot \mathsf{sp}))\big) \cdot \mathrm{dist} \ldots \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ Cop. laws)

$= \ldots \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\mathrm{id} \otimes (\mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp})) \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \quad$ (Cop. laws,

$\quad \cdot \mathsf{sp} \cdot \mathsf{sh} \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ Naturality)

$= \ldots \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot (\mathrm{id} \otimes (\mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp})) \cdot \mathsf{sp} \cdot \mathsf{sh} \cdot \mathsf{jn} \cdot \mathsf{sp}$

$= \ldots \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot (\mathrm{id} \otimes \mathsf{jn} \cdot \mathsf{sp}) \cdot \mathsf{sp} \cdot \mathsf{sh} \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp} \quad$ (Naturality)

$= \ldots \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh} \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp} \quad$ (Coherence)

$\triangleq \llbracket E, x \triangleright \mathsf{case}\ v\ \{\mathsf{inl}(a) \Rightarrow w;\ \mathsf{inr}(b) \Rightarrow u\}\rrbracket \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes \llbracket t \rrbracket) \cdot \mathsf{sp}$

$\square$

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 $\llbracket X \rrbracket$, 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 [36, Theorem 2.3] for the lambda calculus without conditionals, it suffices to consider the cases involving conditionals.

The soundeness of the equations for conditionals follow from Lemma 2.3.1, the coherence theorem for symmetric monoidal categories, naturality, and the universal property of the coproduct. We will provide a complete proof for the β_{case}^{inl} and η_{case} equations, noting that the proof for the β_{case}^{inr} equation follows analogously from the first. For the β_{case}^{inl} equation, we reason as follows:

$[\![\Delta, \Gamma \rhd \mathsf{case\ inl}(v)\ \{\mathsf{inl}(x) \Rightarrow w;\ \mathsf{inr}(y) \Rightarrow u\}]\!]$

$\triangleq [[\![w]\!], [\![u]\!]] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot \mathrm{sw} \cdot (\mathrm{inl} \cdot [\![v]\!] \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$

$= \ldots \cdot \mathrm{dist} \cdot (\mathrm{inl} \otimes \mathrm{id}) \cdot ([\![v]\!] \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$

$= \ldots \cdot \mathrm{dist} \cdot [\mathrm{inl} \otimes \mathrm{id}, \mathrm{inr} \otimes \mathrm{id}] \cdot \mathrm{inl} \cdot ([\![v]\!] \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$ $\qquad$ (Coproduct laws)

$= [[\![w]\!], [\![u]\!]] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{inl} \cdot ([\![v]\!] \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$

$= [\![w]\!] \cdot \mathsf{jn} \cdot \mathrm{sw} \cdot ([\![v]\!] \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$ $\qquad$ (Coproduct laws)

$= [\![w]\!] \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes [\![v]\!]) \cdot \mathrm{sw} \cdot \mathsf{sp} \cdot \mathsf{sh}$ $\qquad$ (Naturality)

$= [\![w]\!] \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes [\![v]\!]) \cdot \mathsf{sp}$ $\qquad$ (Coherence)

$= [\![w[v/x]]\!]$ $\qquad$ (Lemma 2.3.1)

Regarding the η_{case} equation, we have:

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

$\triangleq [[\![w[\mathsf{inl}(y)/x]]\!], [\![w[\mathsf{inr}(z)/x]]\!]] \cdot (\mathsf{jn} \cdot \mathrm{sw} \oplus \mathsf{jn} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot ([\![v]\!] \otimes \mathrm{id})$

$\quad \cdot \mathsf{sp} \cdot \mathsf{sh}$

$= [\![w]\!] \cdot [\mathsf{jn} \cdot (\mathrm{id} \otimes \mathrm{inl} \cdot [\![y]\!]) \cdot \mathsf{sp} \cdot \mathsf{jn} \cdot \mathrm{sw}, \mathsf{jn} \cdot (\mathrm{id} \otimes \mathrm{inr} \cdot [\![z]\!]) \cdot \mathsf{sp} \cdot \mathsf{jn} \cdot \mathrm{sw}] \cdot \ldots$ $\quad$ (Lem. 2.3.1,

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ Cop. laws)

$= [\![w]\!] \cdot \mathsf{jn} \cdot [(\mathrm{id} \otimes \mathrm{inl}) \cdot \mathrm{sw}, (\mathrm{id} \otimes \mathrm{inr}) \cdot \mathrm{sw}] \cdot \ldots$ $\qquad$ (Cop. laws)

$= [\![w]\!] \cdot \mathsf{jn} \cdot \mathrm{sw} \cdot [\mathrm{inl} \otimes \mathrm{id}, \mathrm{inr} \otimes \mathrm{id}] \cdot \mathrm{dist} \cdot ([\![v]\!] \otimes \mathrm{id}) \cdot \mathsf{sp} \cdot \mathsf{sh}$ $\qquad$ (Naturality,

$\qquad$ Cop. laws)

$= [\![w]\!] \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes [\![v]\!]) \cdot \mathrm{sw} \cdot \mathsf{sp} \cdot \mathsf{sh}$ $\qquad$ (Naturality)

$$= [\![w]\!] \cdot \mathsf{jn} \cdot (\mathrm{id} \otimes [\![v]\!]) \cdot \mathsf{sp} \qquad\qquad\qquad\qquad\qquad \text{(Coherence)}$$

$$\triangleq [\![w[v/x] : \mathbb{D}]\!] \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Lem. 2.3.1)}$$

$$\square$$

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. Completeness arises from constructing the syntactic category $\mathsf{Syn}(T)$ of a λ-theory T. The syntactic category of T has as objects the types of T and as morphisms $A \to B$ the equivalence classes (w.r.t. provability) of terms v for which we can derive $x : \mathbb{A} \triangleright v : \mathbb{B}$. This theorem is proved in [36, Lemma 2.6] for the lambda calculus without conditionals, so we only need to address the cases involving conditionals.

By employing the equations β_{case}^{inl}, β_{case}^{inr}, and η_{case} in Figure 2, we show that the universal property of the coproduct is satisfied in $\mathsf{Syn}(T)$. We note that the distributivity follows from the category being closed.

In the syntactic category, the coproduct $[p, q]$ can be seen as the equivalence class

$$[z : \mathbb{A} \oplus \mathbb{B} \triangleright \mathsf{case}\ z\ \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow p;\ \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow q\} : \mathbb{D}] .$$

The proof of the coproduct diagram comutes follow directly from the equations β_{case}^{inl} and β_{case}^{inr} in Figure 2, and substituition (Lemma 2.2.10). Specifically, for the left triangle in the coproduct diagram, we have that:

$$[\mathsf{case}\ z\ \{\mathrm{inl}(x) \Rightarrow p;\ \mathrm{inr}(y) \Rightarrow q\}] \cdot [\mathrm{inl}(x)]$$

$$= [\mathsf{case}\ \mathrm{inl}(x)\ \{\mathrm{inl}(a) \Rightarrow p[a/x];\ \mathrm{inr}(b) \Rightarrow q[b/y]\}] \qquad \text{(Lemma 2.2.10)}$$

$$= [p[a/x][x/a]] \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\beta_{\mathsf{case}}^{\mathsf{inl}})$$

$$= [p]$$

The proof for the right triangle in the coproduct diagram is analogous.

Regarding the unicity of the coproduct, the key aspect of the proof lies in proving that the following equality holds:

$$[z : \mathbb{A} \oplus \mathbb{B} \triangleright \mathsf{case}\ z\ \{\mathrm{inl}(x) \Rightarrow m[\mathrm{inl}(x)/z];\ \mathrm{inr}(y) \Rightarrow m[\mathrm{inr}(y)/z]\} : \mathbb{D}] = [z : \mathbb{A} \oplus \mathbb{B} \triangleright m : \mathbb{D}].$$

This equality follows direct from the η_{case} equation. With this equality at hand, unicity is automatically proven, given that considering any morphism m' from $\mathbb{A} \oplus \mathbb{B}$ to $\mathbb{D}$ and the coproduct diagram, we have that

$$[m'[\mathrm{inl}(x)/z]] = [p]$$

and

$$[m'[\mathrm{inr}(y)/z]] = [q] \,.$$

As a result,

$$[m'] = [\mathsf{case}\ z\ \{\mathrm{inl}(x) \Rightarrow m'[\mathrm{inl}(x)/z];\ \mathrm{inr}(y) \Rightarrow m'[\mathrm{inr}(x)/z]\}]$$
$$= [\mathsf{case}\ z\ \{\mathrm{inl}(x) \Rightarrow p;\ \mathrm{inr}(y) \Rightarrow q\}] \,.$$

$\square$

Remark 2.3.5. Although in higher-order lambda calculus the distributive property follows directly from the fact that the syntactic category is closed, this is not the case for first-order lambda calculus, where the category is not required to be closed. Nevertheless, we will show that the distributive morphism is an isomorphism in $\mathrm{Syn}(T)$, thereby establishing completeness for first-order lambda calculus. Note that our treatment of first-order λ-calculus implicitly assumes that the underlying categories of the models are distributive.

Proposition 2.3.6. *Consider a first-oder linear λ-theory T. Then an equation $\Gamma \rhd v = w : \mathbb{A}$ is a theorem of T if and only if it is satisfied by all first-order models of the theory.*

Proof. In the syntatic category the distributive property, dist, corresponds to the class

$$\big[z : (\mathbb{A} \oplus \mathbb{B}) \otimes \mathbb{D} \rhd \mathsf{pm}\ z\ \mathsf{to}\ x \otimes d.\ \mathsf{case}\ x\ \{\mathrm{inl}_\mathbb{B}(a) \Rightarrow \mathrm{inl}_{\mathbb{B} \otimes \mathbb{D}}(a \otimes d);$$
$$\mathrm{inr}_\mathbb{A}(b) \Rightarrow \mathrm{inr}_{\mathbb{A} \otimes \mathbb{D}}(b \otimes d/z')\}$$
$$: (\mathbb{A} \otimes \mathbb{D}) \oplus (\mathbb{B} \otimes \mathbb{D})\big] \,.$$

The fact that the distributivity morphism is an isomorphism follows by proving that the distributivity morphism is a copairing (by reasoning similar to the previous case), together with the well-known categorical result that coproducts are unique up to isomorphism (Proposition A.2.29). We note that there is an additional step consisting of verifying that dist satisfies the identities

$$\mathrm{dist} \cdot \mathrm{inl}' = \mathrm{inl} \quad \text{and} \quad \mathrm{dist} \cdot \mathrm{inr}' = \mathrm{inr},$$

where $\mathrm{inl}', \mathrm{inr}'$ are the injections of this distributivity morphism. This is immediate once it is established that dist a copairing.

This reduces to proving that the class

$$\big[z : (\mathbb{A} \otimes \mathbb{B}) \otimes \mathbb{D} \triangleright \mathsf{pm}\ z\ \mathsf{to}\ x \otimes d.\ \mathsf{case}\ x\ \{\mathrm{inl}_\mathbb{B}(a) \Rightarrow p[a \otimes d/z'];$$
$$\mathrm{inr}_\mathbb{A}(b) \Rightarrow q[b \otimes d/z']\} : \mathbb{E}\big],$$

equipped with the injections

$$[x : \mathbb{A} \otimes \mathbb{D} \triangleright \mathsf{pm}\ x\ \mathsf{to}\ a \otimes d.\ \mathrm{inl}_\mathbb{B}(a) \otimes d : (\mathbb{A} \oplus \mathbb{B}) \otimes \mathbb{D}],$$
$$[x : \mathbb{B} \otimes \mathbb{D} \triangleright \mathsf{pm}\ x\ \mathsf{to}\ b \otimes d.\ \mathrm{inr}_\mathbb{A}(b) \otimes d : (\mathbb{A} \oplus \mathbb{B}) \otimes \mathbb{D}]$$

is a copairing, $[p, q]$.

The proof of the coproduct diagram comutes follows from equations $\eta_{\otimes_e}, \beta_{\otimes_e}, \beta_{\mathsf{case}}^{\mathsf{inl}}$, and $\beta_{\mathsf{case}}^{\mathsf{inr}}$.

Specifically, for the left triangle in the coproduct diagram, we have:

$$\left[\mathsf{pm}\ z\ \mathsf{to}\ x \otimes d.\ \mathsf{case}\ x \left\{ \begin{array}{l} \mathrm{inl}(a) \Rightarrow p[a \otimes d/z']; \\ \mathrm{inr}(b) \Rightarrow q[b \otimes b/z'] \end{array} \right\} \right] \cdot [\mathsf{pm}\ x\ \mathsf{to}\ a \otimes d.\ \mathrm{inl}(a) \otimes d]$$

$$= \left[\mathsf{pm}\ (\mathsf{pm}\ x\ \mathsf{to}\ a \otimes d.\ \mathrm{inl}(a) \otimes d)\ \mathsf{to}\ x' \otimes d'.\ \mathsf{case}\ x' \left\{ \begin{array}{l} \mathrm{inl}(a') \Rightarrow p[a' \otimes d'/z']; \\ \mathrm{inr}(b) \Rightarrow q[b \otimes d'/z'] \end{array} \right\} \right]$$

$$= \left[\mathsf{pm}\ \mathrm{inl}(a) \otimes d\ \mathsf{to}\ x' \otimes d'[x/a \otimes d].\ \mathsf{case}\ x' \left\{ \begin{array}{l} \mathrm{inl}(a') \Rightarrow p[a' \otimes d'/z']; \\ \mathrm{inr}(b) \Rightarrow q[b \otimes d'/z'] \end{array} \right\} \right] \qquad (\eta_{\otimes_e})$$

$$= \left[\mathsf{case}\ \mathrm{inl}(a) \left\{ \begin{array}{l} \mathrm{inl}(a') \Rightarrow p[a' \otimes d/z']; \\ \mathrm{inr}(b) \Rightarrow q[b \otimes d/z'] \end{array} \right\} [x/a \otimes d] \right] \qquad (\beta_{\otimes_e})$$

$$= [p\,[a \otimes d/z'][x/a \otimes d]] \qquad (\beta_{\mathsf{case}}^{\mathsf{inl}})$$

$$= [p]$$

The proof for the right triangle in the coproduct diagram is analogous.

To prove unicity, we first establish that the following equality, known as the *syntactic fusion law*, holds:

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

This equality follows from the extensionality of the copairing and equations $\beta_{\mathsf{case}}^{\mathsf{inl}}$, and $\beta_{\mathsf{case}}^{\mathsf{inr}}$. By the extensionality of the copairing, it suffices to prove the equility for $\mathrm{inl}(b)$ and $\mathrm{inr}(c)$. We

present the explicit proof for inl(b); the proof for inr(c) follows similar reasoning.

$$v\left[(\mathsf{case}\ \mathrm{inl}(b)\ \{\mathrm{inl}(x) \Rightarrow w;\ \mathrm{inr}(y) \Rightarrow u\})\,/z\right]$$

$$= v[w[b/x]\,/z] \qquad\qquad\qquad (\beta_{\mathsf{case}}^{\mathrm{inl}})$$

$$= v[[w/z]\,[b/x]]$$

$$= \mathsf{case}\ \mathrm{inl}(b)\ \{\mathrm{inl}(x) \Rightarrow v[w/z];\ \mathrm{inr}(y) \Rightarrow v[u/z]\} \quad (\beta_{\mathsf{case}}^{\mathrm{inl}})$$

Next, to establish that this morphism is unique, we establish that given any morphism m from $(\mathbb{A} \otimes \mathbb{B}) \otimes \mathbb{D}$ to $\mathbb{E}$, $z : (\mathbb{A} \otimes \mathbb{B}) \otimes \mathbb{D} \triangleright m : \mathbb{E}$, we have

$$\left[\mathsf{pm}\ z\ \mathsf{to}\ x \otimes d.\ \mathsf{case}\ a \begin{cases} \mathrm{inl}(a) \Rightarrow m[\mathrm{inl}(a) \otimes d/z]; \\ \mathrm{inr}(b) \Rightarrow m[\mathrm{inr}(b) \otimes d/z] \end{cases}\right]$$

$$= \left[\mathsf{pm}\ z\ \mathsf{to}\ x \otimes d.\ m \left[\mathsf{case}\ x \begin{cases} \mathrm{inl}(a) \Rightarrow \mathrm{inl}(a) \otimes d; \\ \mathrm{inr}(b) \Rightarrow \mathrm{inr}(b) \otimes d \end{cases} /z \right]\right] \qquad \text{(syntactically fusion law)}$$

$$= \left[\mathsf{pm}\ z\ \mathsf{to}\ x \otimes d.\ m \left[\mathsf{case}\ x \begin{cases} \mathrm{inl}(a) \Rightarrow y \otimes d\,[\mathrm{inl}(a)/y]; \\ \mathrm{inr}(b) \Rightarrow y \otimes d\,[\mathrm{inr}(b)/y] \end{cases} /z \right]\right]$$

$$= [\mathsf{pm}\ z\ \mathsf{to}\ x \otimes d.\ m\,[y \otimes d\,[x/y]\,/z]] \qquad\qquad\qquad (\eta_{\mathsf{case}})$$

$$= [m[x \otimes d/z][z/x \otimes d]] \qquad\qquad\qquad (\eta_{\otimes_e})$$

$$= [m]$$

Finally, observe that,

$$[m[\mathsf{pm}\ x\ \mathsf{to}\ a \otimes d.\ \mathrm{inl}(a) \otimes d/z]]$$

$$= [\mathsf{pm}\ x\ \mathsf{to}\ a \otimes d.m[\mathrm{inl}(a) \otimes d/z]] \qquad\qquad (c_{\otimes_e})$$

$$= [m[\mathrm{inl}(a) \otimes d/z][a \otimes d/x]] \qquad\qquad (\eta_{\otimes_e})$$

Similarly we obtain $[m[\mathsf{pm}\ x\ \mathsf{to}\ b \otimes d.\ \mathrm{inr}(b) \otimes d/z]] = [m[\,\mathrm{inr}(b) \otimes d/z][b \otimes d/x]]$.

$\square$

Example 2.3.7. 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 $\{n : \mathbb{I} \rightarrow N \mid n \in \mathbb{N}\}$, 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 $: N \rightarrow \mathbb{I}$. In Set we

have $[\![\mathbb{I}]\!] = \{*\}$, and define $[\![N]\!] = \mathbb{N}$, $[\![n]\!] = \{*\} \to \mathbb{N}$, $* \mapsto n$, and $[\![\mathsf{dis}]\!] = !$, where $!$ denotes the terminal map. Consider the following λ-term:

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

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

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

Next, consider the λ-terms below.

$$\textbf{Dis2nd} \triangleq - \triangleright \lambda x : N \otimes \mathbb{I}.\mathsf{pm}\ x\ \mathsf{to}\ a \otimes b.b\ \mathsf{to}\ *.a : N \otimes N \multimap N$$
$$\textbf{Dis2nd}\ (1(*) \otimes *)$$

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:

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

The resulting program is interpreted as follows:

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

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 2.3.8. 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 2.3.9. 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 $A \multimap (-) : \mathsf{C} \to \mathsf{C}$ 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 that 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 2.3.10. 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 $\llbracket X \rrbracket$ 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 2.3.11 (Soundness). *[36, 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 $q \geq d(\llbracket \Gamma \ \triangleright v : \mathbb{A} \rrbracket, \llbracket \Gamma \triangleright w : \mathbb{A} \rrbracket)$.*

Next, we will provide a proof sketch of the completeness result in [36] 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 $\mathtt{Values}(\mathbb{A}, \mathbb{B})$ of values v such that $x : \mathbb{A} \triangleright v : \mathbb{B}$. We equip $\mathtt{Values}(\mathbb{A}, \mathbb{B})$ with the function $d : \mathtt{Values}(\mathbb{A}, \mathbb{B}) \times \mathtt{Values}(\mathbb{A}, \mathbb{B}) \to [0, \infty]$ defined by,

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

Given that the equations $\Gamma \triangleright v = w : \mathbb{A}$ are abbreviations of $\Gamma \triangleright v =_0 w : \mathbb{A}$, $\mathtt{Values}((\mathbb{A}, \mathbb{B}), d)$ 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 $(\mathtt{Values}(\mathbb{A}, \mathbb{B}), d)/\sim$, which is a Met-category.

Completeness arises from constructing the syntactic category $\mathrm{Syn}(T)$ of the underlying theory T and then showing that provability of $\Gamma \rhd v =_q w : \mathbb{A}$ in T is equivalent to $d(\llbracket v \rrbracket, \llbracket w \rrbracket) \geq q$ in the category $\mathrm{Syn}(T)$. We use the category $(\mathtt{Values}(\mathbb{A}, \mathbb{B}), d)/\sim$ to this end. Note that the quotienting process identifies all terms $x : \mathbb{A} \rhd v : \mathbb{B}$ and $x : \mathbb{A} \rhd w : \mathbb{B}$ 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 $+$ distributes over suprema. This yields the desired category $\mathrm{Syn}(T)$ 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 $\Gamma \rhd v =_q v' : \mathbb{A}$ with $q \in [0, \infty]$ is satisfied by $\mathrm{Syn}(T)$, 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 2.3.12 (Completeness)**.** *[36, Theorem 3.16] Consider a metric λ-theory. A metric equation in context $\Gamma \rhd v =_q w : \mathbb{A}$ 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.

3.1 System

Our system for conditionals is presented in Figure 5.

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

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

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 a metric 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.

Metric Copairing's Extensionality

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

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

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

$$v = \mathsf{case}\; x \left\{ \mathsf{inl}(y) \Rightarrow v\,[\mathsf{inl}(y)/x]; \mathsf{inl}(z) \Rightarrow v\,[\mathsf{inr}(z)/x] \right\} \hspace{2em} (\eta_{case})$$

$$=_{\sup\{\varepsilon_1,\varepsilon_2\}} \mathsf{case}\; x \left\{ \mathsf{inl}(y) \Rightarrow w\,[\mathsf{inl}(y)/x]; \mathsf{inl}(z) \Rightarrow w\,[\mathsf{inr}(z)/x] \right\} = w \hspace{1em} (\text{Figure 5}, \eta_{case})$$

As a result, we have $v =_{\sup\{\varepsilon_1,\varepsilon_2\}} w$ and derive $\lambda x.v =_{\sup\{\varepsilon_1,\varepsilon_2\}} \lambda x.w$ using the our 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 $f_1, f_2 \in C(A, C)$ and $g_1, g_2 \in C(B, C)$, 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 [36, Example 3.8], the category Met is a symmetric monoidal closed Met-category. As a result, it suffices to show that for all morphisms $f_1, f_2 \in \text{Met}(A, C)$ and $g_1, g_2 \in \text{Met}(B, C)$, the following inequality holds:

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

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

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

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

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

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

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

$\square$

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 [36, Section 3] and [50, 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. *[38, 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*

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

Soundeness and Completeness

The proofs in this section are based on the proofs of Theorem 2.3.11 and Theorem 2.3.12 in [36].

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 $\Gamma \rhd v =_q w : \mathbb{A}$ results from the rules in Figures 2, 3 and 5 then $q \geq d(\llbracket \Gamma \ \rhd v : \mathbb{A} \rrbracket, \llbracket \Gamma \rhd w : \mathbb{A} \rrbracket)$.*

Proof. We follow the same strategy as in [36]. 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 $\Gamma \rhd w =_0 v : \mathbb{A}$. By Theorem 2.3.3, these equations are sound for symmetric monoidal closed categories with binary coproducts, *i.e.* if $v = w$, then $\llbracket v \rrbracket = \llbracket w \rrbracket$ in C. Then by the definition of metric space we obtain $d(\llbracket v \rrbracket, \llbracket w \rrbracket) = d(\llbracket w \rrbracket, \llbracket v \rrbracket) = 0$. 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,

$$d(\llbracket \, \mathsf{case}\, v\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w;\, \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u\} \rrbracket, \llbracket \, \mathsf{case}\, v'\, \{\mathsf{inl}_{\mathbb{B}}(x) \Rightarrow w';\, \mathsf{inr}_{\mathbb{A}}(y) \Rightarrow u'\} \rrbracket)$$

$$= d(\llbracket\llbracket w \rrbracket, \llbracket u \rrbracket\rrbracket \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E,$$
$$\qquad \llbracket\llbracket w' \rrbracket, \llbracket u' \rrbracket\rrbracket \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v' \rrbracket \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E)$$

$$\leq d(\llbracket\llbracket w \rrbracket, \llbracket u \rrbracket\rrbracket \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v \rrbracket \otimes \mathsf{id}), \llbracket\llbracket w' \rrbracket, \llbracket u' \rrbracket\rrbracket \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus$$
$$\qquad \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist} \cdot (\llbracket v' \rrbracket \otimes \mathsf{id}))$$

$$\leq d(\llbracket v \rrbracket \otimes \mathsf{id}, \llbracket v' \rrbracket \otimes \mathsf{id}) + d(\llbracket\llbracket w \rrbracket, \llbracket u \rrbracket\rrbracket \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist}, \llbracket\llbracket w' \rrbracket, \llbracket u' \rrbracket\rrbracket \cdot$$
$$\qquad (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist})$$

$$\leq q + d(\llbracket\llbracket w \rrbracket, \llbracket u \rrbracket\rrbracket \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw}) \cdot \mathrm{dist}, \llbracket\llbracket w' \rrbracket, \llbracket u' \rrbracket\rrbracket \cdot (\mathsf{jn}_{\Delta;\mathbb{A}} \cdot \mathrm{sw} \oplus \mathsf{jn}_{\Delta;\mathbb{B}} \cdot \mathrm{sw})$$

$$\cdot\,\mathrm{dist})$$

$$\leq q + d([\![w]\!], [\![u]\!]], [\![w']\!], [\![u']\!]])$$

$$\leq q + \sup(d([\![w]\!], [\![w']\!]), d([\![u]\!], [\![u']\!]))$$

$$\leq q + \sup\{r, s\}$$

The second step follows from the fact that $\mathrm{sp}_{\Gamma;\Delta} \cdot \mathrm{sh}_E$ 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.

$\square$

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 pseudometric 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 [36]. Completeness arises from constructing the syntactic category $\mathrm{Syn}(T)$ 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 $\mathrm{Syn}(T)$. We use the category $(\mathtt{Values}(\mathbb{A}, \mathbb{B}), d)/\sim$ 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 $(\mathrm{Syn}(T))$, *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 that following implication holds:

$$
\begin{cases}
\inf\{q \mid v =_q w\} \leq 0 \\[2mm]
\inf\{r \mid v' =_r w'\} \leq 0
\end{cases}
\implies
\inf\left\{ q \ \middle| \ \begin{array}{l} \text{case } z \ \{\text{inl}(x) \Rightarrow v; \ \text{inr}(y) \Rightarrow w\} =_q \\[1mm] \text{case } z \ \{\text{inl}(x) \Rightarrow v'; \ \text{inr}(y) \Rightarrow w'\} \end{array} \right\} \leq 0.
$$

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

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

(3.1)

With the equality above, we have

$$
\begin{cases}
\inf\{q \mid v =_q w\} \leq 0 \\[2mm]
\inf\{r \mid v =_r w\} \leq 0
\end{cases}
$$

$$
\implies \sup\{\inf\{q \mid v =_q w\}, \inf\{r \mid v =_r w\}\} \leq 0
$$

$$
\implies \inf\{\sup\{q, r\} \mid v =_q w, v =_r w\} \leq 0 \qquad\qquad \text{(Equation 3.1)}
$$

$$
\implies \left\{ q \ \middle| \ \begin{array}{l} \text{case } z \ \{\text{inl}(x) \Rightarrow v; \ \text{inr}(y) \Rightarrow w\} =_q \\[1mm] \text{case } z \ \{\text{inl}(x) \Rightarrow v'; \ \text{inr}(y) \Rightarrow w'\} \end{array} \right\} \leq 0
$$

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

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

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 $\Gamma \rhd v =_q v' : \mathbb{A}$ with $q \in [0, \infty]$ is satisfied by $\mathsf{Syn}(T)$, then it is a theorem of the linear metric λ-theory. By assumption, $d([v], [v']) = d(v, v') = \inf \{r \mid v =_r v'\} \le q$. 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 $A \subseteq \{r \mid v =_r v'\}$ such that $x \ge \inf A$. 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. $\qquad\square$

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:

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

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 $i \in I$,

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

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 $\Phi, \Phi' \in \mathsf{C}^+(A, C)$ and $\Psi, \Psi' \in \mathsf{C}^+(B, C)$, 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:

$$d([\Phi, \Psi], [\Phi', \Psi'])$$
$$= d(\xi, \xi') = \sup\{d'(\xi_i, \xi_i') \mid i \in I' \cup I''\}$$
$$= \sup\{\{d'(\phi_i, \phi_i') \mid i \in I'\} \cup \{d'(\psi_i, \psi_i') \mid i \in I''\}\}$$
$$= \sup\{\sup\{d'(\phi_i, \phi_i') \mid i \in I'\}, \sup\{d'(\psi_i, \psi_i') \mid i \in I''\}\} \qquad \text{(Lemma 3.3.5)}$$
$$= \sup\{d(\Phi, \Phi'), d(\Psi, \Psi')\}$$

□

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.

Part II

Applications

Chapter 4

Probabilistic Programming

Computer science and probability theory have shared a fruitful relationship since the early days [39]. Over the years, probabilistic algorithms have emerged as powerful tools across diverse domains—from machine learning [89] and robotics [113] to computational linguistics [78]. These algorithms also play a pivotal role in modern cryptography, particularly in public-key systems [54], and tackle computationally intractable problems [82].

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 [55], while more recent innovations, such as Anglican [115], 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 [32, 33], 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 [36], 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 to tensor products of Banach spaces based on [95]. 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 [37, 13, 2]. Finally, using our calculus, we study approximate equivalence in the context of a random walk on the real line.

4.1   Tensor Product in Banach spaces

Before introducing the category of Banach spaces and proving it is a model, we first establish some notation and define the tensor product in this context, as it is less familiar than the standard basics of Banach space theory.

First, we establish some notation. The letters V, W, R will often refer to vector spaces, and $\mathcal{F}$ represents the field of scalars of a vector space. We denote by $\mathcal{B}(V, W)$ the vector space of all bounded linear operators from V to V, and we write $\mathcal{B}(V)$ for $\mathcal{B}(V, V)$. The corresponding operator norm is denoted by $\| \cdot \|_{\mathbf{op}}$.

Definition 4.1.1 (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 $g = h \circ \otimes$.

$$
\begin{array}{ccc}
V \times W & \xrightarrow{\ \otimes\ } & V \odot W \\
 & {}_{g}\searrow & \big\downarrow {\scriptstyle h} \\
 & & R
\end{array}
$$

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

$$
\begin{array}{ccc}
V_1 \times V_2 & \longrightarrow & V_1 \odot V_2 \\
\downarrow & & \vdots \\
W_1 \times W_2 & \longrightarrow & W_1 \odot W_2
\end{array}
$$

Definition 4.1.2. [95, 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 4.1.3. [95, 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 Cauchy completion of the algebraic tensor product $V \odot W$ with respect to the projective norm $\|\cdot\|_\pi$.

4.2 The category Ban

In this section, leveraging results from [36], 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 R be Banach spaces. Let $T : V \to R$ and $S : W \to R$ be short maps. Then, it holds that*

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

Proof. We calculate,

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

$$= \sup\left\{\|v\| \cdot \|T\left(v/\|v\|\right)\| + \|w\| \cdot \|S\left(w/\|w\|\right)\| \mid \|v\| + \|w\| = 1\right\}$$

$$= \sup\left\{\sup\left\{\|T(v)\| \mid \|v\| = 1\right\}, \sup\left\{\|S(w)\| \mid \|w\| = 1\right\}\right\}$$

$$= \sup\{\|T\|_{\mathsf{op}}, \|S\|_{\mathsf{op}}\}$$

In the second to last step we observe that the expression

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

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

$\square$

Theorem 4.2.3. *[36, Theorem 4.3] The category* Ban *is a symmetric monoidal closed* Met-*category in which both the tensor product and the internal hom-functor (currying) are non-expansive.*

Theorem 4.2.4. *The category* Ban *is a symmetric monoidal closed* Met-*category with binary coproducts in which both the tensor product and the internal hom-functor (currying) are non-expansive.*

Proof. This follows from Theorem 4.2.3 and Lemma 4.2.2. $\square$

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 generalize 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 [5] 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 $A^c \in \mathcal{B}$ (*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 se-

quence 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. $A_1, A_2, \cdots \in \mathcal{B} \implies \bigcup_i A_i \in \mathcal{B}$ (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*. We note, however, that a measure can be negative; as a result, we will drop the requirement $\lambda(\cdot) \geq 0$.

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 $\mathbb{R}$), λ, and its generalizations to $\mathbb{R}^n$, λ_n. It is characterized as the unique measure on the Borel sets whose value on every interval is its length, *i.e.*, for any interval $[a, b] \subset \mathbb{R}$, $\lambda([a, b]) = b - a$. The Lebesgue measure, enjoys the properties listed in the theorem below.

Theorem 4.3.5. *[13, Theorem 1.7.3] Let B be a Lebesgue measurable set of finite measure. Then:*

1. $\lambda_n(B + v) = \lambda_n(B)$ *for any vector $v \in \mathbb{R}^n$;*

2. $\lambda_n(\alpha B) = |\alpha|^n \lambda_n(B)$ *for any real number α.*

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

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

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 4.3.6. 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 4.3.7. 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 4.3.8. 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 $\{B_0, \ldots, B_n\}$ is 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 finer 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 4.3.9. 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 4.3.10. 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.

Definition 4.3.11. Let μ be a measure on a measurable space $(S, \mathcal{B})$. The measures μ^+ and μ^- are defined as follows:

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

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

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

Proposition 4.3.12. *[13, Section 3.1] For a measure μ, it holds that:*

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

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 $[\![CoinToss_p]\!] : \mathbb{R} \to \mathbb{R} \oplus \mathbb{R},\ 1 \mapsto (p, 1-p)$. We also take a pre-determined set of measures

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

which are interpreted as

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

for all $a, b, p_1, \dots, p_n, x_1, \dots, x_n \in \mathbb{Q}$, 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 $(\mu, \nu) \mapsto +_*(\mu \otimes_{\textsf{meas}} \nu)$. Finally, we consider a pre-determined set of jumps $j : \mathbb{I} \to (\texttt{real} \multimap \texttt{real})$ interpreted as

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

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 $jl, jr : \mathbb{I} \to (\texttt{real} \multimap \texttt{real})$ we can describe a single step of the random walks a follows,

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

Given an argument $r : \texttt{real}$ representing the walker's current position, the program **step** performs a random jump: with probability p, the walker jumps to the left, and with probability $1 - p$ the walker jumps to the right.

Now that we have defined a single step of the walk, we introduce the λ-term

$$\textbf{apply-n} = \lambda f_1, \dots, f_n, r.\ f_1(f_2(\dots(f_n(r)))),$$

which composes a sequence of n functions and applies them to an initial input r. This allows us to define a random walk of n steps starting from the origin as

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

Recall that jumps $j : \mathbb{I} \to (\mathtt{real} \multimap \mathtt{real})$ are interpreted as

$$\llbracket j \rrbracket (1) = \mu \mapsto +_* (\mu \otimes_{\mathsf{meas}} \llbracket m_0 \rrbracket (*)),$$

where $m_0 \in m$. It is straightforward to prove that the following axiom is sound for each of them:

$$j(*) =_0 \lambda x. + (x, m_0(*)). \tag{4.1}$$

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

$$\llbracket jl \rrbracket (1) = \mu \mapsto +_* (\mu \otimes_{\mathsf{meas}} \mathtt{unif}(0, -1)) \qquad \llbracket jr \rrbracket (1) = \mu \mapsto +_* (\mu \otimes_{\mathsf{meas}} \mathtt{unif}(1, 0)).$$

Operationally, jl corresponds to a jump to the left with magnitude between 0 and 1, and analogously for jr. Suppose we have another jump $\llbracket jr^\delta \rrbracket : \mathbb{R} \to (\mathcal{M}\mathbb{R} \multimap \mathcal{M}\mathbb{R})$ whose interpretation is that of jr except for the fact that $\mathtt{unif}(0, 1)$ is replaced by $\mathtt{unif}(0, 1 + \delta)$. What will be the effect on the random walk when replacing jr by jr^δ? Observe that we can put an upper bound between the terms $\mathtt{unif}(0, 1)(*)$ and $\mathtt{unif}(0, 1 + \delta)(*)$, *semantically* by computing the norm $\|\mathtt{unif}(0, 1) - \mathtt{unif}(0, 1 + \delta)\|$. Attending to Proposition 4.3.12, we have,

$$\|\mathtt{unif}(0, 1) - \mathtt{unif}(0, 1 + \delta)\|$$
$$= (\mathtt{unif}(0, 1) - \mathtt{unif}(0, 1 + \delta))^+ (\mathbb{R}) + (\mathtt{unif}(0, 1) - \mathtt{unif}(0, 1 + \delta))^- (\mathbb{R})$$

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

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

The first follows from Definition 4.3.11; the second step uses the countable additivity of measures; the third follows from the second property in Theorem 4.3.5; the fourth step follows from the definition of *supremum*.

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 $2 \cdot \frac{\delta}{1+\delta}$.

Then, with this in hand, one can put an upper bound between $jr(*)$ and $jr^\delta(*)$ via the previous axioms (Equation 4.1) and our deductive metric system, $jr(*) =_{2 \cdot \frac{\delta}{1+\delta}} jr^\delta$.

Additionally, suppose $CoinToss_p$ is replaced by $CoinToss_q$. We calculate:

$$\|[\![CoinToss_p(*)]\!] - [\![CoinToss_q(*)]\!]\| = \|(p, 1-p) - (q, 1-q)\|$$

$$= \|(p-q, q-p)\| = 2|p-q|$$

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

$$\text{case } CoinToss_p(*)\{\mathrm{inl}(x) \Rightarrow jl(x); \mathrm{inr}(y) \Rightarrow jr(y)\}$$

$$=_0 \text{ case } CoinToss_p \{\mathrm{inl}(x) \Rightarrow jl(y); \mathrm{inr}(y) \Rightarrow x \text{ to } * . \, jr(*)\}$$

$$=_{2 \cdot \left(|p-q| + \frac{\delta}{1+\delta}\right)} \text{ case } CoinToss_q\{\mathrm{inl}(x) \Rightarrow jl(y); \mathrm{inr}(y) \Rightarrow x \text{ to } * . \, jr^\delta(*)\}$$

$$=_0 \text{ case } CoinToss_q\{\mathrm{inl}(x) \Rightarrow jl(x); \mathrm{inr}(y) \Rightarrow jr^\delta\}$$

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 $|p-q|$ tend to 0.

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

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

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 probability p_i. Now, consider another jump jl^{q_i} whose interpretation corresponds to that of jl except 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 4.3.10, we compute,

$$\left\| \sum_i p_i \delta_{-x_i} - \sum_i q_i \delta_{-x_i} \right\|$$

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

$$= \sum_i |p_i - q_i|$$

Here, using the inequality

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

we observe that the supremum is achieved when each Dirac mass is placed in a distinct partition.

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

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

which converges to 0 as $|p - q|$ and $|p_i - q_i|$ tend to 0 for all $1 \leq i \leq n$.

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 [47]. 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 [107]. 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 [93]. Unfortunately, general-purpose error correction techniques [21, 56, 109] 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 [48]. 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 [117] and noise-adaptive optimization [83]. 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 [61, 20]. 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 [127, 105]. 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 [99]. 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 [101]. This was followed by a denotational semantics for a fragment of the language in [103]. In subsequent work, they extended the quantum lambda calculus to include recursion and infinite types, along with its operational semantics [104]. Later, they proposed an alternative approach to its denotational semantics [86].

These works adopt Schrödinger's picture, in which quantum programs are interpreted as maps between quantum states (*i.e.*, density operators). In contrast, [23, 24] consider Heisenberg's picture, in which programs are modeled as maps between observables (*i.e.*, self-adjoint operators). Particularly, [23] presents a model based on W^*-algebras, which can be viewed as an infinite-dimensional extension of [99]. Moreover, [24] presents a model of Selinger and Valiron's quantum lambda calculus [101, 102, 104], also based on W^*-algebras, and proves

74

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 [25]. 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, [65] and [112] 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 [65] 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 [36], 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) [119].

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 [85, 119], while the mathematical foundations are also based on [62, 30, 29]. The next section presents the fundamentals of W^*-algebras, drawing primarily from [96, 111, 120]. We then show that both $\mathcal{K}(\mathsf{CPS})$, the idempotent completion of the category of quantum operations (i.e., completely positive, trace-nonincreasing super-operators), and $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$, the opposite category of W^*-algebras with normal, completely positive, subunital maps [120, 23], 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 complex vector space V is a function from a mapping $V \times V$ to $\mathbb{C}$, $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{C}$, that satisfies the following properties for all $v, w, w_1, \ldots, w_n \in V$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{C}$.

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 $v = (\alpha_1, \ldots, \alpha_n), w = (\beta_1, \ldots, \beta_n) \in \mathbb{C}^n$ 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 $\mathcal{H}, \mathcal{K}, \mathcal{L}$ 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. *[119, Corollary 1.4] Let $\mathcal{H}$ be a Hilbert space. Every normal operator $A \in \mathcal{B}(\mathcal{H})$ 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,

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

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

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

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

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:

$$\|A\|_1 = \mathsf{Tr}\sqrt{A^\dagger A}$$

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 $d(\rho, \rho') = \|\rho - \rho'\|_1$.

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 $\mathsf{Tr}(|A|) < \infty$, 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* [91]. 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 an associated *Bloch vector* given by $(\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$.
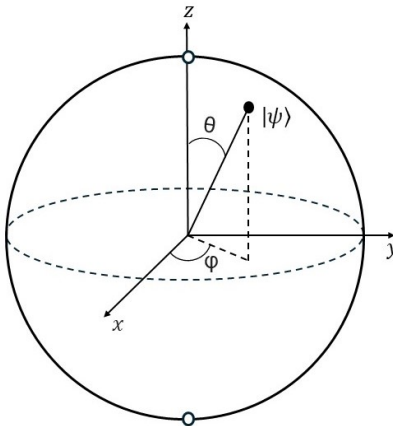
Figure 6: Bloch sphere representation of a qubit

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 [85].

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 $|x_1 \ldots x_n\rangle$ and so a quantum state of such a system is specified by 2^n amplitudes. For instance, a two-qubit state can be written as

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

It should be noted that unfortunately, no simple generalization of the Bloch sphere 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 [44] 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 [10]. 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 [4]. 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 [114] and Anton Zeilinger [87] conducted similar experiments, demonstrating that entanglement persists over distances of several kilometers. More recently, [124] 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:

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

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

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. *[85] 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:

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

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:

$$CU(|0\rangle \otimes |\psi\rangle) = |0\rangle \otimes |\psi\rangle$$

$$CU(|1\rangle \otimes |\psi\rangle) = |1\rangle \otimes U|\psi\rangle.$$

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

$$|\psi'\rangle = \frac{M_0 |\psi\rangle}{|\alpha|} = \frac{\alpha}{|\alpha|}|0\rangle = |0\rangle \text{ (with } p = p_0) \quad \text{and}$$

$$|\psi''\rangle = \frac{M_1 |\psi\rangle}{|\beta|} = \frac{\beta}{|\beta|}|1\rangle = |1\rangle \text{ (with } p = p_1)$$

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 $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, where p_i 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 $p_m = \mathsf{Tr}(M_m\rho M_m^\dagger)$. After observing outcome m, the post-measurement state collapses to:

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

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

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

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 following 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

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

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 as

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

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

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

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 [62].

5.2.6 Quantum Channels and Operations

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.

In this perspective, we focus on quantum states, represented by density operators. Consequently, we are also interested in *state transformers*—operators that map density operators to density operators.

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* (denoted $\Phi \geq 0$) if it sends positive operators to positive operators, *i.e.* $A \geq 0 \Rightarrow \Phi(A) \geq 0$, for all $A \geq 0 \in \mathcal{T}(\mathcal{H})$.

Definition 5.2.23. A super-operator $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ is *hermitian-preserving* if it sends hermitian operators to hermitian operators, *i.e.* $A = A^{\dagger} \Rightarrow (\Phi(A))^{\dagger} = \Phi(A)$, for all hermitan operators $A \in \mathcal{T}(\mathcal{H})$.

Definition 5.2.24. 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 5.2.25. A super-operator $\Phi : \mathcal{T}(\mathcal{H}_1) \to \mathcal{T}(\mathcal{H}_2)$ is *completely positive* if the super-operator $\Phi \otimes \mathrm{id}_{\mathcal{T}(\mathcal{K})} : \mathcal{T}(\mathcal{H}_1 \otimes \mathcal{K}) \to \mathcal{T}(\mathcal{H}_2 \otimes \mathcal{K})$ is positive for any Hilbert space $\mathcal{K}$.

Definition 5.2.26. A super-operator Φ is called *trace-preserving* (resp. *trace-nonincreasing*) if $\mathrm{Tr}\,(\Phi A) = \mathrm{Tr}(A)$ (resp. ($\mathrm{Tr}\,(\Phi A) \leq \mathrm{Tr}(A)$)).

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) [106].

The following property will be helpful later on.

Theorem 5.2.27. *[119, Theorem 2.25] Let $\Phi \in \mathcal{T}(H) \rightarrow \mathcal{T}(K)$ be a super-operator. The following statements are equivalent:*

1. *Φ is Hermitian preserving.*

2. *There exist positive maps $\Phi_0, \Phi_1 : \mathcal{T}(H) \rightarrow \mathcal{T}(K)$ for which $\Phi = \Phi_0 - \Phi_1$.*

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

$$\rho_E = \sum_i p_i \, |i\rangle \, \langle i|$$

where $|i\rangle$ 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,

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

where $\{|\mu\rangle\}$ 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,

$$
\begin{aligned}
\rho_S' &= \sum_\mu \langle\mu| U\rho_S \otimes \sum_i p_i |i\rangle\langle i| U^\dagger |\mu\rangle \\
&= \sum_{\mu i} \sqrt{p_i} \langle\mu| U |i\rangle \rho_S \sqrt{p_i} \langle i| U^\dagger |\mu\rangle \\
&= \sum_{\mu i} K_{\mu i} \rho_S K_{\mu i}^\dagger
\end{aligned}
$$

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

Definition 5.2.28. The equation $\rho_S' = \sum_{\mu i} K_{\mu i} \rho_S K_{\mu i}^\dagger$ is called an ***Operator Sum Representation (OSR)***. An OSR can be thought of as a quantum channel that maps ρ_S to $\sum_{\mu i} K_{\mu i} \rho_S K_{\mu i}^\dagger$, given this map is CPTP ([Section V-VI] [75], [85, Chapter 8.2.3]).

In the definition above, the condition $\sum_\mu K_\mu K_\mu^\dagger = I$ is implicit [75, 85]. However, following the convention in [119], we will refer to any operator $\Phi : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ expressible in the form

$$
\Phi(A) = \sum_i K_i A K_i^\dagger,
$$

where $K_i \in \mathcal{B}(H, K)$, as a *Kraus representation*—regardless of whether the Kraus operators $\{K_i\}$ satisfy the normalization condition.

The following property will be useful in this context.

Theorem 5.2.29. *[119, Theorem 2.22] Let $\Phi : \mathcal{T}(H, K)$ be super-operator. The following statements are equivalent:*

1. *Φ is completely positive.*

2. *Φ admits a Kraus representation.*

In particular, a quantum operation Φ can always be expressed as $\Phi(\rho) = \sum_i K_i \rho K_i^\dagger$, with operators K_i satisfying $\sum_i K_i K_i^\dagger \leq I$, *i.e.*, $I - \sum_i K_i^\dagger K_i$ is positive [85, Theorem 8.1].

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 5.2.30. 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 $\{M_m\}$.

In the case of measurements in the computational basis, these correspond to what are known as *quantum-to-classical* channels, i.e., channels $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{H})$ satisfying $\Phi = \Delta\Phi$, where Δ is the completely dephasing channel defined by

$$\Delta(\rho) = \sum_i (\langle i | \rho | i \rangle) | i \rangle \langle i |, \tag{5.5}$$

with $\{|i\rangle\}$ being an orthonormal basis for $\mathcal{H}$ [119, Proposition 2.36].

5.2.7 Norms on quantum operations

Definition 5.2.31. 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 $\|\Phi \otimes I_{\mathcal{T}(\mathcal{H})}\|_1 \leq \|\Phi\|_1$ does not hold [119]. As a result, the diamond norm, which is based on the trace norm, is used instead in the context of quantum operations.

Definition 5.2.32. Given a super-operator $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$, the diamond norm, $\|\cdot\|_\diamond$, is defined as:

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

The following properties will be useful in .

Theorem 5.2.33. *Let* $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ *be a positive map. Then it holds that*

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

Corollary 5.2.34. *Let* $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ *be a quantum operation, it holds that:*

$$\|\Phi\|_\diamond \leq 1.$$

Proof. Given that Φ is a quantum operation, if follows that $\Phi \otimes \operatorname{id}_{\mathcal{T}(\mathcal{H})}$ is a positive trace-nonincreasing super-operator. Let $\Psi = \Phi \otimes \operatorname{id}$, it holds that,

$$
\begin{aligned}
\|\Phi\|_\diamond &= \|\Psi\|_1 \\
&= \max\left\{\mathsf{Tr}\left(\Psi(uu^\dagger)\right) \mid \|u\|_2 = 1\right\} && \text{(Theorem 5.2.33)} \\
&= \max\left\{\sum_i \sum_j \mathsf{Tr}\left(\Psi_{ij}(u_i u_i^\dagger)\right) \mid \left\|\left(u_1, \ldots, u_s^2\right)^T\right\|_2 = 1\right\} \\
&\leq \max\left\{\sum_i \mathsf{Tr}\left(u_i u_i^\dagger\right) \mid \sqrt{\sum_i \|u_i\|_2^2} = 1\right\} && \text{(Ψ is trace-nonincreasing)} \\
&= 1
\end{aligned}
$$

$\square$

Theorem 5.2.35. *[119, Theorem 3.46] Let* $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ *be a super-operator. Then*

$$\left\|\Phi \otimes \operatorname{id}_{\mathcal{T}(\mathcal{L})}\right\|_1 \leq \|\Phi\|_\diamond,$$

with equality holding under the assumption that $\dim(\mathcal{H}) \geq \dim(\mathcal{L})$.

Proposition 5.2.36. *[119, Proposition 3.48] For all super-operators* $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ *and* $\Psi \in \mathcal{T}(\mathcal{K}) \to \mathcal{T}(\mathcal{L})$*, it holds that*

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

Theorem 5.2.37. *[119, Theorem 3.49] Let* $\Phi : \mathcal{T}(\mathcal{H}_1) \to \mathcal{T}(\mathcal{K}_1)$ *and* $\Psi : \mathcal{T}(\mathcal{H}_2) \to \mathcal{T}(\mathcal{K}_2)$ *be super-operators. Then it holds that*

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

Theorem 5.2.38. *[119, Theorem 3.51] Let $\Phi : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ be a Hermitian-preserving map. Then it holds that*

$$\|\Phi\|_{\diamond} = \max \|(\Phi \otimes \mathrm{id})(vv^*) \mid \|v\|_2 = 1, v \in \mathcal{T}(\mathcal{H})\|_1.$$

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

Theorem 5.2.39. *[119, Theorem 3.55] Let $n \leq m$, let $V_0, V_1 \in \mathcal{T}(\mathcal{H}, \mathcal{K})$ be isometries, and define CPTP operators $\Phi_0, \Phi_1 : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{K})$ as*

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

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 5.2.40. *[119, 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 5.2.41. 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
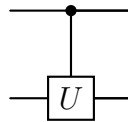
$$|\psi_0\rangle \text{———}$$
$$|\psi_1\rangle \text{———}$$

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
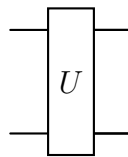
$$\boxed{H}$$
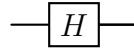
The controlled-not gate, which is a two-qubit gate, is represented as
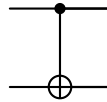
Similarly, the controlled-U gate, where U is an unitary single-qubit gate, is represented as

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 [122]. 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 W^*-Algebras

In this section, we are no longer restricted to finite-dimensional vector spaces; *i.e.*, the term "vector space" now also encompasses infinite-dimensional ones.

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 [63]. Born and Jordan subsequently recognized that these arrays should follow the rules of matrix multiplication [16]. To address the mathematical challenges posed by "infinite matrices", Von Neumann formalized these ideas using operators on Hilbert spaces, more concretely, *Von Neumann* algebras [84]. This gave rise to the study of operator algebras, which are now applied

in various domains in quantum theory, including quantum statistical mechanics [17], quantum field theory [3, 59], and quantum information theory [68]. 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 are more suitable for this purpose. For instance, whereas C^*-algebras correspond to noncommutative geometry [28], W^*-algebras can be viewed as noncommutative analogues of measure theory or probability, aligning with the probabilistic nature of quantum physics [60, 76]. Moreover, there is previous work in this setting: in [23] 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 5.4.2.

5.3.1 C^*-Algebras

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

Definition 5.3.1. 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 $\alpha \in \mathbb{C}$,

$$(a^*)^* = a, \quad (ab)^* = b^*a^*, \quad (\alpha a)^* = \overline{\alpha}a^*, \quad \text{and} \quad (a+b)^* = a^* + b^*;$$

4. a complete norm $\|\cdot\|$ such that $\|ab\| \le \|a\| \cdot \|b\|$ for all $a, b \in \mathscr{A}$, and

$$\|a^*a\| = \|a\|^2.$$

This last equality is called the C^*-*identity*.

Remark 5.3.2. 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 5.3.3. 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 5.3.4. 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 5.3.5. *[90, 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 5.3.6. *[23, 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 5.3.7. 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 5.3.8. *[111, Theorem 9.18.] Every C^*-algebra admits a faithful representation.*

Matrices over C^*-Algebras

Definition 5.3.9. 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 5.3.10. *[88, 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 $\mathcal{M}_n(\mathcal{B}(\mathcal{H})) = \mathcal{B}(\mathcal{H}^{\oplus n})$ [88, Exercises 1.1 and 1.2]. That is, given a faithful representation $(\mathcal{H}, \pi)$ of $\mathscr{A}$— where π is an isometry by Proposition 5.3.5—we have, for $(A_{ij}) \in \mathcal{M}_n(\mathscr{A})$,

$$\|(A_{ij})\| \triangleq \left\| \big(\pi(A_{ij}) \big) \right\|_{\mathrm{op}} = \sup \{ \left\| \big(\pi(A_{ij}) \big)(v) \right\| \mid \|v\|_2 = 1, v \in \mathcal{H}^{\oplus n} \},$$

where $\big(\pi(A_{ij}) \big)(v)$ is of the form

$$\big(B_{ij} \big) \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \displaystyle\sum_{j=1}^n B_{1j} v_j \\ \vdots \\ \displaystyle\sum_{j=1}^n B_{nj} v_j \end{pmatrix},$$

for $(B_{ij}) \in \mathcal{M}_n(\mathcal{B}(H))$ and $(v_1, \dots, v_n) \in H^{\oplus n}$.

These matrices are important because they are used to define complete positivity in this setting.

Definition 5.3.11. 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

$$\mathcal{M}_n(\Phi) : \mathcal{M}_n(\mathscr{A}) \to \mathcal{M}_n(\mathscr{B}), \quad \mathcal{M}_n(\Phi)\big(A_{ij}\big) := \big(\Phi(A_{ij})\big).$$

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 presented a definition of complete positivity in the setting of bounded/trace-class operators over finite-dimensional Hilbert spaces (Definition 5.2.25). Given $\mathcal{B}(\mathcal{H})$ is an example of a C^*-algebra, we will check that the definitions are equivalent in the aforementioned setting.

Proposition 5.3.12. *The definitions of completely positivity presented (Definition 5.2.25 and Definition 5.3.11) are equivalent for maps*

$$\Phi : \mathcal{B}(\mathcal{H} \otimes_2 \mathcal{K}_1) \to \mathcal{B}(\mathcal{H} \otimes_2 \mathcal{K}_2),$$

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 [119]. Therefore, Definition 5.2.25 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 5.3.11. Let $\mathcal{M}_n$ denote the vector space of complex $n \times n$ matrices. Here we note the isometric isomorphisms $\mathcal{M}_n \otimes \mathcal{M}_m \cong \mathcal{M}_{nm}$, $\mathcal{B}(\mathcal{H}) \cong \mathcal{M}_n$ with $\dim(\mathcal{H}) = n$ [119], and $\mathcal{M}_n(\mathcal{B}(\mathcal{H})) \cong \mathcal{M}_n \otimes \mathcal{B}(\mathcal{H})$ [42, 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 $i\big((a_{i,j})\big) = \sum_{i,j} |i\rangle \langle j| \otimes a_{ij}$, where $\{|i\rangle \langle j|\}_{i,j=1}^n$ denotes any orthonormal basis for $\mathcal{M}_n$. As a result, $b := i\big((b_{i,j})\big) = \sum_{i,j} |i\rangle \langle j| \otimes b_{ij}$ and $b^* := \left(\sum_{i,j} |i\rangle \langle j| \otimes b_{ij} \right)^* = \sum_{j,i} |j\rangle \langle i| \otimes b_{ij}^*$. Next, we calculate:

$$b^* b = \left(\sum_{i,j} |j\rangle \langle i| \otimes b_{ij}^* \right) \left(\sum_{i,j} |i\rangle \langle j| \otimes b_{ij} \right)$$

$$= \sum_{i,k,l,j} |i\rangle \langle k| |l\rangle \langle j| \otimes b_{k,i}^* b_{l,j}$$

$$= \sum_{i,k,l,j} |i\rangle \langle j| \otimes \left(\sum_k b_{k,i}^* b_{k,j} \right) \qquad (|i\rangle \langle k| |l\rangle \langle j| = \delta_{kl} |i\rangle \langle j|)$$

$$= i\big((b_{i,j})^*(b_{i,j})\big),$$

thereby demonstrating that i is positive.

The equivalence of definitions 5.2.25 and 5.3.11 follows from the fact that the two diagrams below commute and the composition of positive maps is positive. In 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 $n \in \mathbb{N}$.

$$
\begin{array}{ccc}
\mathcal{B}(\mathcal{H} \otimes \mathcal{K}_1) & \xrightarrow{\mathrm{id}_{\mathcal{T}(\mathcal{H})} \otimes \Phi} & \mathcal{B}(\mathcal{H} \otimes \mathcal{K}_2) \\
\big\uparrow \cong & & \big\downarrow \cong \\
\mathcal{M}_{nm} & & \mathcal{M}_{no} \\
\big\uparrow \cong & & \big\downarrow \cong \\
\mathcal{M}_n \otimes \mathcal{M}_m & & \mathcal{M}_n \otimes \mathcal{M}_o \\
\Big\uparrow {\scriptstyle [a_{i,j}] \mapsto \sum_{i,j} |i\rangle\langle j| \otimes a_{ij}} & & \Big\downarrow {\scriptstyle A \otimes B \mapsto [(a_{i,j}B)_{i,j}]} \\
\mathcal{M}_n(\mathcal{M}_m) & \xrightarrow[\mathcal{M}_n(\Phi)]{} & \mathcal{M}_n(\mathcal{M}_o)
\end{array}
$$

$$
\begin{array}{ccc}
\mathcal{B}(\mathcal{H} \otimes \mathcal{K}_1) & \xrightarrow{\mathrm{id}_{\mathcal{T}(\mathcal{H})} \otimes \Phi} & \mathcal{B}(\mathcal{H} \otimes \mathcal{K}_2) \\
\big\downarrow \cong & & \big\uparrow \cong \\
\mathcal{M}_{nm} & & \mathcal{M}_{no} \\
\big\downarrow \cong & & \big\uparrow \cong \\
\mathcal{M}_n \otimes \mathcal{M}_m & & \mathcal{M}_n \otimes \mathcal{M}_o \\
\Big\downarrow {\scriptstyle A \otimes B \mapsto [(a_{i,j}B)_{i,j}]} & & \Big\uparrow {\scriptstyle [a_{i,j}] \mapsto \sum_{i,j} |i\rangle\langle j| \otimes a_{ij}} \\
\mathcal{M}_n(\mathcal{M}_m) & \xrightarrow[\mathcal{M}_o(\Phi)]{} & \mathcal{M}_n(\mathcal{M}_o)
\end{array}
$$

$\square$

The following result we be useful later on.

Proposition 5.3.13 (Proposition 2.3). *[23] Let $\Phi : \mathscr{A} \to \mathscr{B}$ be a $*$-homomorphism between C^*-algebras. Then Φ is completely positive.*

Direct sums of C^*-Algebras

Definition 5.3.14. 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 5.3.15. 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 omitted unless ambiguity arises. The completion $\mathscr{A}_1 \check{\otimes} \mathscr{A}_2$ 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 C^*-norm* (respectively, C^*-tensor product).

5.3.2 W^*-Algebras

The letters $\mathscr{M}, \mathscr{N}, \mathscr{T}$ will typically denote W^*-algebras.

Basics of W^*-Algebras

Definition 5.3.16. 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 [96, Corollary 1.13.3].

Remark 5.3.17. 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 [96, Chapter 1.7].

Definition 5.3.18. The weak* topology on $\mathscr{M}$ induced by the predual V, which is the coarsest (weakest) topology that makes all functions $V^* \to \mathbb{C}, \quad v^* \mapsto v^*(v)$ continuous for each $v \in V$, 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 5.3.19. *[29, Theorem 19.2] Let $\mathcal{H}$ be a Hilbert space. The dual of $\mathcal{T}(\mathcal{H})$ is isometrically isomorphic to $\mathcal{B}(\mathcal{H})$ via the map*

$$\Phi : \mathcal{B}(\mathcal{H}) \to \mathcal{T}(\mathcal{H})^*, \quad \Phi(T)(-) = \mathrm{tr}(T(-)), \quad \text{for all } A \in \mathcal{T}(\mathcal{H}).$$

Example 5.3.20. For a Hilbert space $\mathcal{H}$, $\mathcal{B}(\mathcal{H})$ is a W^*-algebra (see the Proposition immediately above).

Remark 5.3.21. While the terms W^*-algebras and von Neumann algebras are often used interchangeably (*e.g.*, in [120]), the latter typically denotes concrete ultraweakly closed C^*-subalgebras of $\mathcal{B}(\mathcal{H})$.

Definition 5.3.22. 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 5.3.14.

Proposition 5.3.23. *[120, 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 5.3.14) is itself a W^*-algebra, and the canonical projections*

$$\pi_j : \bigoplus_i A_i \to A_j, \quad \text{given by } \pi_j(a) = a(j),$$

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 [120], rather than the more common approach based on ultraweak completion of the spatial tensor product of C^*-algebras (Definition 5.3.15) as in [111, 96]. We adopt this definition because we will

primarily use its properties to prove that $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ is a symmetric monoidal Met-category with binary coproducts, in which the tensor product is nonexpansive. The approach in [120] 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 [120, Theorem 111 VII].

Definition 5.3.24. 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 $\beta(m_1 m_2, n_1 n_2) = \beta(m_1, n_1)\,\beta(m_2, n_2)$ 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 5.3.25. 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 ultra-weakly 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 $\varphi_1 : \mathscr{M} \to \mathbb{C}$ and $\varphi_2 : \mathscr{N} \to \mathbb{C}$, the product functional $\gamma(\varphi_1, \varphi_2) : \mathscr{T} \to \mathbb{C}$ exists and is positive.

3. The product functionals $\gamma(\varphi_1, \varphi_2)$ of normal positive functionals φ_1 and φ_2 form a *faithful collection* of normal positive functionals on $\mathscr{T}$ (*i.e.*, $t \in \mathscr{T}_+$ is zero iff $\gamma(\varphi_1, \varphi_2)(t) = 0$ for all such functionals).

Definition 5.3.26. 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 5.3.27. 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 [120, Definition 112 II (3)].

Definition 5.3.28. 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 4.1.1 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.

Definition 5.3.29. 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 4.1.1, but for the W^*-tensor product rather than the algebraic case. Later, in Section 5.4.2, we will make use of this result for proving that $\mathsf{W}^*_{\mathsf{CPSU}}$ is a first-order model.

Theorem 5.3.30. *[120, 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 W^*-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 5.3.31. *[120, 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 $\mathrm{W}^*_{\mathsf{CPSU}}$ is a first-order model in Section 5.4.2.

Proposition 5.3.32. *[120, Proposition 115 II] Given normal completely positive maps $\Phi :$ $\mathscr{M}_1 \to \mathscr{N}_1$ and $\Psi : \mathscr{M}_2 \to \mathscr{N}_2$ between W^*-algebras, there exists a unique normal completely positive map*

$$\Phi \overline{\otimes} \Psi : \mathscr{M}_1 \overline{\otimes} \mathscr{M}_2 \to \mathscr{N}_1 \overline{\otimes} \mathscr{N}_2$$

such that

$$(\Phi \overline{\otimes} \Psi)(m \otimes n) = \Phi(m) \otimes \Psi(n) \quad \text{for all } m \in \mathscr{M}_1,\ n \in \mathscr{M}_2.$$

Moreover $\Phi \overline{\otimes} \Psi$ is (sub)unital if Φ and Ψ are (sub)unital.

Proposition 5.3.33. *[120, Proof 115 III] Let $\mathscr{M}$ and $\mathscr{N}$ be W^*-algebras. Given normal completely positive maps $\Phi : \mathscr{M} \to \mathscr{T}$ and $\Psi : \mathscr{N} \to \mathscr{T}$. We may take $\Phi \overline{\otimes} \Psi := \beta_{\bar{\otimes}}$ as in the theorem Theorem 5.3.30. It holds that, $\|\beta_{\odot}(s)\|_{w^*} \le \|\Phi\|_{op} \|\Psi\|_{op} \|s\|_{w^*}$, given an element $s \in \mathscr{M} \overline{\otimes} \mathscr{N}$.*

Proposition 5.3.34. *[120, Corollary 119 IV] There is a unique normal $*$-isomorphism*

$$\alpha_{\mathscr{M},\mathscr{N},\mathscr{T}} : \mathscr{M} \overline{\otimes} (\mathscr{N} \overline{\otimes} \mathscr{T}) \longrightarrow (\mathscr{M} \overline{\otimes} \mathscr{N}) \overline{\otimes} \mathscr{T},$$

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 5.3.35. *[120, 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 5.3.36. *[120, Proposition 117 III] Given W^*-algebras $\mathscr{M}$ and $(\mathscr{N}_i)_{i \in I}$, we have a natural isomorphism*

$$\mathscr{M} \mathbin{\overline{\otimes}} \bigoplus_{i \in I} \mathscr{N}_i \;\cong\; \bigoplus_{i \in I} \mathscr{M} \mathbin{\overline{\otimes}} \mathscr{N}_i.$$

That is, the spatial tensor product distributes over (possibly infinite) direct sums.

Corollary 5.3.37. *[23, Theorem 3.2] Let $\mathscr{M}, \mathscr{N}, \mathscr{T}$ be W^*-algebras. Then the canonical map*

$$\langle \mathrm{id} \mathbin{\overline{\otimes}} \pi_1, \mathrm{id} \mathbin{\overline{\otimes}} \pi_2 \rangle : \mathscr{M} \mathbin{\overline{\otimes}} (\mathscr{N} \oplus \mathscr{T}) \to (\mathscr{M} \mathbin{\overline{\otimes}} \mathscr{N}) \oplus (\mathscr{M} \mathbin{\overline{\otimes}} \mathscr{T})$$

is a unital $*$-isomorphism, and therefore, isometric.

Theorem 5.3.38. *[120, 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 5.3.39. *The category $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ is a distributive symmetric monoidal category.*

Proof. It follows directly from Propositions 5.3.23 and 5.3.36 and Theorem 5.3.38 □

The following result will be useful for demonstrating that $\mathsf{W}^*_{\mathsf{CPSU}}$ is a first-order model in Section 5.4.2.

5.4 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 [15]. 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 $\mathcal{K}(\mathsf{CPS})$, the idempotent completion of the category of quantum operations . 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 $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$, the opposite of the category $\mathsf{W}^*_{\mathsf{CPSU}}$, whose objects are W^*-algebras and morphisms are completely positive subunital maps between them.

5.4.1   Schrödinger's picture

The category CPTP of quantum channels was shown in [36] to form a symmetric monoidal Met-category. We begin by presenting the more general symmetric monoidal category CPS of quantum operations, which can similarly be shown to be a symmetric monoidal Met-category.

Definition 5.4.1. The category CPS is the category whose objects are natural numbers $n \geq 1$ and whose morphisms $n \to m$ are quantum operations $\mathcal{M}_n \to \mathcal{M}_m$.

The following proposition establishes that CPS is a first-order λ-calculus model without conditionals drawing from [36].

Proposition 5.4.2. *The category* CPS *is a symmetric monoidal* Met-*category in which the tensor product is non-expansive.*

Proof. Here we follow the same reasoning as [36, Proof of Proposition 4.1, Section 4.3]. First, we establish that CPS is Met-enriched. By unpacking the relevant definitions, this reduces to proving the following: for all CPS-morphisms $\Phi, \Phi' : n \to m$ and $\Psi, \Psi' : m \to o$ the inequation $\|\Phi - \Phi'\|_{\diamond} + \|\Psi - \Psi'\|_{\diamond} \geq \|\Psi\Phi - \Psi'\Phi'\|_{\diamond}$ holds. We proceed as follows:

$$
\begin{aligned}
&\|\Phi - \Phi'\|_{\diamond} + \|\Psi - \Psi'\|_{\diamond} \\
&\geq \|(\Phi - \Phi')\Psi\|_{\diamond} + \|\Phi'(\Psi - \Psi')\|_{\diamond} \quad \text{(Proposition 5.2.36, and Corollary 5.2.34)} \\
&= \|\Phi\Psi - \Phi'\Psi\|_{\diamond} + \|\Phi'\Psi - \Phi'\Psi'\|_{\diamond} \\
&\geq \|\Phi\Psi - \Phi'\Psi + \Phi'\Psi - \Phi'\Psi'\|_{\diamond} \qquad\qquad \{\text{Triangle inequality}\} \\
&= \|\Psi\Phi - \Psi'\Phi'\|_{\diamond} .
\end{aligned}
$$

Next, to prove that $\|\Phi - \Phi'\|_\diamond + \|\Psi - \Psi'\|_\diamond \geq \|\Psi \otimes \Phi - \Psi' \otimes \Phi'\|_\diamond$, we calculate

$$\|\Psi - \Psi'\|_\diamond + \|\Phi - \Phi'\|_\diamond$$

$$\geq \|\mathrm{id} \otimes (\Psi - \Psi')\|_\diamond + \|\mathrm{id} \otimes (\Phi - \Phi')\|_\diamond \qquad \text{(Theorem 5.2.37 and Corollary 5.2.34)}$$

$$= \|\mathrm{id} \otimes \Psi - \mathrm{id} \otimes \Psi'\|_\diamond + \|\mathrm{id} \otimes \Phi - \mathrm{id} \otimes \Phi'\|_\diamond$$

$$\geq \|(\mathrm{id} \otimes \Psi) \cdot (\Phi \otimes \mathrm{id}) - (\mathrm{id} \otimes \Psi') \cdot (\Phi' \otimes \mathrm{id})\|_\diamond \qquad \text{(CPS is a Met-category)}$$

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

$$\square$$

Since, unfortunately CPS it does meet the pre-requisite of having coproducts, a natural candidate for interpreting quantum programs with coproducts is the category CPS^+, obtained via the coproduct cocompletion of CPS. However, for CPS^+ 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 $\rho = \left(\begin{smallmatrix} a & c \\ d & b \end{smallmatrix}\right) \in \mathcal{M}_2$ to a classical bit $(a, d) \in \mathbb{C} \oplus \mathbb{C}$. Unfortunately, this is not the case, since $\mathsf{CPS}^+(2, 1{+}1)$ consists of operations $\Phi : 2 \to 1$ followed by a left or right injection. Consequently, we cannot access both measurement outcomes simultaneously. We could consider introducing measurements via a "product completion". However, a similar problem would arise for the coproduct given $\mathsf{CPS}^+(1 + 1, 2)$ consists of operations corresponding to a left or right projection followed by an operation $\Phi : 1 \to 2$. This suggests that $+$ should be a biproduct. However, as observed in [99], not all morphisms of the biproduct are required: the coproduct structure and its projection morphisms suffice. We will see next that what we need is not a biproduct completion but rather an *idempotent completion*, *i.e.*, creating a new category where all idempotent morphisms split. This is done via the Karoubi envelope which formally adds these splittings to the category [15].

Definition 5.4.3. For any category C, its *Karoubi envelope* (or *idempotent completion*), denoted $\mathcal{K}(\mathsf{C})$, is the category whose objects are pairs (A, e_A), where A is an object in C and $e_A \colon A \to A$ is an idempotent morphism in C, *i.e.*, $e_A \cdot e_A = e_A$. A morphism $f \colon (A, e_A) \to (B, e_B)$ in $\mathcal{K}(\mathsf{C})$ is a morphism $A \to B$ in C such that $f \circ e_A = f = e_B \circ f$. The identity morphism on an object (A, e_A) is $e_A \colon A \to A$. Composition of morphisms is inherited from C: if $f \colon (A, e_A) \to (B, e_B)$ and $g \colon (B, e_B) \to (C, e_C)$ are morphisms in $\mathcal{K}(\mathsf{C})$, then $g \circ f$ is defined as their composition in C.

The idea is that both the coproducts and the measurement operation, in a sense, "live hidden" within the category and what is required is to "draw them out" in a suitable way. By

adding splittings of idempotents, we effectively construct direct sums, and then carve co-products and measurements out of them.

Definition 5.4.4. The Karoubi envelope $\mathcal{K}(\mathrm{CPS})$ of CPS has as objects pairs (n, P) with $P : \mathbb{C}^{n \times n} \to \mathbb{C}^{n \times n}$ an idempotent quantum operation. A morphism $\Phi : (n, P) \to (m, Q)$ in $\mathcal{K}(\mathrm{CPS})$ is a CPS-morphism $n \to m$ such that $Q \cdot \Phi \cdot P = \Phi$. The identity of (n, P) is precisely P and composition is inherited from CPS.

The following result will be useful to show that the category $\mathcal{K}(\mathrm{CPS})$ inherits relevant structure from CPS.

Proposition 5.4.5. *[14, Proposition 6.5.9] Let* C *be a symmetric monoidal category. Then its Karoubi envelope* C *inherits a canonical symmetric monoidal structure.*

With this result in hand, we can now establish the following proposition.

Proposition 5.4.6. $\mathcal{K}(\mathrm{CPS})$ *is a symmetric monoidal* Met-*category in which the tensor product is non-expansive.*

Proof. First, since CPS is already a Met-category $\mathcal{K}(\mathrm{CPS})$, becomes a Met-category as well, by stipulating the metric in $\mathcal{K}(\mathrm{CPS})(n, m)$ to be the respective restriction of the metric in $\mathrm{CPS}(n, m)$. Second, $\mathcal{K}(\mathrm{CPS})$ inherits symmetric monoidal structure from CPS (Proposition 5.4.5), where in particular the tensor of $(n, P) \otimes (m, Q)$ will be $(nm, P \otimes Q)$ and the tensoring of morphisms is as in CPS. The category $\mathcal{K}(\mathrm{CPS})$ is thus even a symmetric monoidal Met-category in which the tensor product is non-expansive. $\square$

In order to prove that $\mathcal{K}(\mathrm{CPS})$ is a model of our framework, it remains to show that it has binary coproducts, it is distributive, and the copairing satisfies the inequality in Definition 3.2.1. Recall that $\mathcal{K}(\mathrm{CPS})$ is the idempotent split completion of CPS. We will now show that the former has binary coproducts, in particular that these are precisely splits relative to a certain kind of idempotent (*i.e.*, projection). Interestingly, we begin by establishing the copairing as a CPS-morphism, which we then use to define binary coproducts.

Take a quantum operation $\Phi : n \to m$ and consider $[\Phi, 0] : n + o \to m$ defined by,

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \longmapsto \Phi(A)$$

By an appeal to Kraus' representation theorem (Theorem 5.2.29) one easily shows that this map is completely positive, as detailed next.

$$[\Phi, 0] \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \Phi(A)$$

$$= \sum_i M_i A M_i^\dagger \qquad \text{(Theorem 5.2.29)}$$

$$= \sum_i \begin{pmatrix} M_i & 0 \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} M_i^\dagger \\ 0 \end{pmatrix}$$

Observe the existence of an analogous operator $[0, \Phi] : o + n \to m$. Recall as well that Kraus' representation theorem entails that the addition of completely-positive operators will also be completely positive. This means that given quantum operations $\Phi : n \to m$ and $\Psi : o \to m$ their "co-pairing",

$$[\Phi, \Psi] := [\Phi, 0] + [0, \Psi]$$

will be completely positive as well. Moreover, it is trace-nonincreasing, since subtracting the sum of the corresponding Kraus operators from the block identity matrix results in a positive operator, as it corresponds to a block-diagonal matrix with positive operators on the diagonal. Consider as well the quantum operations $e_l : n \to n + m$ and $e_r : m \to n + m$,

$$A \longmapsto \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix} \qquad\qquad D \longmapsto \begin{pmatrix} 0 & 0 \\ 0 & D \end{pmatrix}$$

which have Kraus' operators $\begin{pmatrix} I \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ I \end{pmatrix}$ respectively.

We now define the "direct sum" of quantum operations.

Definition 5.4.7. Given quantum operations $\Phi : n \to o$ and $\Psi : m \to p$ we define $\Phi \oplus \Psi : n + m \to o + p$ as $\Phi \oplus \Psi := [e_1 \cdot \Phi, e_2 \cdot \Psi]$.

Observe that $\Phi \oplus \Psi$ is a quantum operation by construction. In fact, the operator $\Phi \oplus \Psi$ is a particular case of a quantum-to-classical channel , and specifically $\mathrm{id} \oplus \mathrm{id}$ is a block matrix generalisation of the completely dephasing channel Δ (Equation 5.5). The crucial observation then is that idempotents of the form $P \oplus Q$ give rise to coproducts in $\mathcal{K}(\mathsf{CPS})$, as we will show in the following proposition.

Proposition 5.4.8. *The category* $\mathcal{K}(\mathrm{CPS})$ *has binary coproducts. For objects* (n, P) *and* (m, Q), *their coproduct* $(n, P) + (m, Q)$ *is given by the object* $(n + m, P \oplus Q)$. *The co-pairing map* $[\Phi, \Psi] : m \oplus n \to o$ *is defined as above,* $[\Phi, \Psi] := [\Phi, 0] + [0, \Psi]$. *The injections* inl *and* inr *are defined as* $\mathrm{inl} := (P \oplus Q) \cdot e_l \cdot P$ *and* $\mathrm{inr} := (P \oplus Q) \cdot e_r \cdot Q$.

Proof. Given $\mathcal{K}(\mathrm{CPS})$-morphisms $\Phi : (n, P) \to (o, R)$ and $\Psi : (m, Q) \to (o, R)$ it follows from straightforward calculations that its copairing is a $\mathcal{K}(\mathrm{CPS})$-morphism.

The fact that $[\Phi, \Psi] \cdot \mathrm{inl} = \Phi$ follows from the idempotency of P and the fact that $\Phi \cdot P = \Phi$. The same reasoning applies to $[\Phi, \Psi] \cdot \mathrm{inr} = \Psi$. In order to prove unicity consider a suitably typed operator Ω such that $\Omega \cdot \mathrm{inl} = \Phi$ and $\Omega \cdot \mathrm{inr} = \Psi$. Then we reason,

$$
\Omega \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \Omega \begin{pmatrix} P(A) & 0 \\ 0 & Q(D) \end{pmatrix} \qquad (\Omega \text{ is a } \mathcal{K}(\mathrm{CPS})\text{-morphism})
$$

$$
= \Omega(\mathrm{inl}(A) + \mathrm{inr}(D))
$$

$$
= \Omega(\mathrm{inl}(A)) + \Omega(\mathrm{inr}(D))
$$

$$
= \Phi(A) + \Psi(D)
$$

$$
= [\Phi, \Psi] \begin{pmatrix} A & B \\ C & D \end{pmatrix}
$$

$\square$

Proposition 5.4.9. *The category* $\mathcal{K}(\mathrm{CPS})$ *is strictly distributive.*

Proof. Take objects (n, P), (m, Q), (o, R), and note that the following equations are sound.

$$
\begin{cases} \big((n, P) + (m, Q)\big) \otimes (o, R) = (no + mo, (P \oplus Q) \otimes R) \\ (n, P) \otimes (o, R) + (m, Q) \otimes (o, R) = (no + mo, (P \otimes R) \oplus (Q \otimes R)) \end{cases}
$$

They tell that the two relevant composite objects are the same if $(P \oplus Q) \otimes R = (P \otimes R) \oplus$

$(Q \otimes R)$ – and thus we reason,

$$(P \oplus Q) \otimes R \begin{pmatrix} A \otimes B & C \otimes D \\ E \otimes F & G \otimes H \end{pmatrix}$$

$$= (P \oplus Q) \otimes R \left(\begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix} \otimes B + \begin{pmatrix} 0 & C \\ 0 & 0 \end{pmatrix} \otimes D + \begin{pmatrix} 0 & 0 \\ E & 0 \end{pmatrix} \otimes F + \begin{pmatrix} 0 & 0 \\ 0 & G \end{pmatrix} \otimes H \right)$$

$$= \begin{pmatrix} P(A) & 0 \\ 0 & 0 \end{pmatrix} \otimes R(B) + \begin{pmatrix} 0 & 0 \\ 0 & Q(G) \end{pmatrix} \otimes R(H)$$

$$= \begin{pmatrix} P(A) \otimes R(B) & 0 \\ 0 & Q(G) \otimes R(H) \end{pmatrix}$$

$$= (P \otimes R) \oplus (Q \otimes R) \begin{pmatrix} A \otimes B & C \otimes D \\ E \otimes F & G \otimes H \end{pmatrix}$$

which establishes an equality between the two composite objects. The final step is to prove that the equation $[\mathrm{inl} \otimes \mathrm{id}, \mathrm{inr} \otimes \mathrm{id}] = \mathrm{id}$ holds which by unicity reduces to the equations,

$$\begin{cases} \mathrm{inl} \otimes \mathrm{id} = \mathrm{inl} \\ \mathrm{inr} \otimes \mathrm{id} = \mathrm{inr} \end{cases}$$

whose proof is direct. $\qquad\square$

Our next step is to tell how the co-pairing defined above for $\mathcal{K}(\mathsf{CPS})$ interacts with the diamond norm. This is our basis for establishing $\mathcal{K}(\mathsf{CPS})$ is a symmetric monoidal Met-category with binary coproducts, as desired.

Proposition 5.4.10. *The following equation holds for all hermitian-preserving superoperators* $\Phi : \mathcal{M}_n \to \mathcal{M}_o$ *and* $\Psi : \mathcal{M}_m \to \mathcal{M}_o$ *and vectors* $v \in \mathcal{M}_n$

$$\sup\{\|\Phi\|_1, \|\Psi\|_1\} \geq \max\{\|[\Phi, \Psi](vv^*)\|_1 \mid \|v\|_2 = 1\}.$$

Proof. With the key observation that when $\|v\| = 1$, $vv^\dagger$ is a density operator [119, Eq. 2.12]

114

and hence positive, we proceed as follows.

$$\max\{\|[\Phi, \Psi](vv^*)\|_1 \mid \|v\|_2 = 1\}$$

$$= \max\left\{\|\Phi(A) + \Psi(D)\|_1 \,\middle|\, \left\|\begin{pmatrix} A & B \\ C & D \end{pmatrix}\right\|_1 = 1\right\} \qquad (\|v\|_2 = 1 \implies \|vv^\dagger\|_1 = 1)$$

$$\leq \sup\left\{\|\Phi(A) + \Psi(D)\|_1 \,\middle|\, \left\|\begin{pmatrix} A & 0 \\ 0 & D \end{pmatrix}\right\|_1 = 1\right\} \qquad (\|\mathrm{id} \oplus \mathrm{id}\| = 1)$$

$$= \sup\left\{\|\Phi(A) + \Psi(D)\|_1 \mid \|A\|_1 + \|D\|_1 = 1\right\} \qquad (\left(\begin{smallmatrix} A & 0 \\ 0 & D \end{smallmatrix}\right) \text{ is positive})$$

$$\leq \sup\left\{\|\Phi(A)\|_1 + \|\Psi(D)\|_1 \mid \|A\|_1 + \|D\|_1 = 1\right\}$$

$$= \sup\big\{\, \|A\|_1 \|\Phi((1/\|A\|_1)A)\|_1$$

$$\qquad + \|D\|_1 \|\Psi((1/\|D\|_1)D)\|_1 \mid \|A\|_1 + \|D\|_1 = 1\big\}$$

$$\leq \sup\big\{\, \sup\{\|\Phi(A)\|_1 \mid \|A\|_1 = 1\},$$

$$\qquad \sup\{\|\Phi(D)\|_1 \mid \|D\|_1 = 1\}\big\}$$

$$= \sup\{\|\Phi\|, \|\Psi\|\}$$

$\square$

With this property, we have all the means to show that $\mathcal{K}(\mathsf{CPS})$ is a symmetric monoidal Met-category with binary coproducts.

Proposition 5.4.11. *For all quantum operations $\Phi, \Phi' : n \to o$ and $\Phi, \Phi' :\ m \to o$, it holds that*

$$\|[\Phi - \Phi', \Psi - \Psi']\|_\diamond \leq \sup\{\|\Phi - \Phi'\|_\diamond, \|\Psi - \Psi'\|_\diamond\}.$$

Proof. We reason as follows:

$$\|[\Phi - \Phi', \Psi - \Psi']\|_\diamond$$

$$= \|[\Phi, \Psi] \otimes \mathrm{id}_{n+m} - [\Phi', \Psi'] \otimes \mathrm{id}_{n+m}\|_1$$

$$= \|[\Phi \otimes \mathrm{id}_{n+m}, \Psi \otimes \mathrm{id}_{n+m}] - [\Phi' \otimes \mathrm{id}_{n+m}, \Psi' \otimes \mathrm{id}_{n+m}]\|_1 \qquad \text{(Proposition 5.4.9)}$$

$$\leq \max\{\|[(\Phi - \Psi) \otimes \mathrm{id}_{n+m}, (\Psi - \Psi') \otimes \mathrm{id}_{n+m}](vv^*)\|_1 \mid \|v\|_2 = 1\} \quad \text{(Thm. 5.2.27, 5.2.38)}$$

$$\leq \sup\{\|\Phi \otimes \mathrm{id}_{n+m}\|_1, \|\Psi \otimes \mathrm{id}_{n+m}\|_1\} \qquad \text{(Proposition 5.4.10)}$$

$$\leq \sup\{\|\Phi\|_\diamond, \|\Psi\|_\diamond\} \qquad \text{(Theorem 5.2.35)}$$

$\square$

Theorem 5.4.12. *The category* $\mathcal{K}(\mathrm{CPS})$ *is a symmetric monoidal* Met-*category with binary coproducts in which the tensor product is non-expansive.*

Proof. It follows directly from Proposition 5.4.6 and Proposition 5.4.11. $\qquad\square$

5.4.2 Heisenberg's picture

In the previous quantum model, we considered Schrödinger's picture—that is, morphisms between quantum states (*i.e.*, density operators). In [23], 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 $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$, the opposite of the category $\mathsf{W}^*_{\mathsf{CPSU}}$ whose objects are W^*-algebras and morphisms are completely positive subunital maps between them. It is shown that Selinger's category Q is equivalent to the finite-dimensional subcategory of $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$, $(\mathsf{FdW}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ [23]. Now, we will prove that $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ is a model of our first-order lambda calculus.

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

$$\left\| \Phi \odot \mathrm{id}_{\mathscr{A}} \colon \mathscr{A} \odot \mathscr{A} \to \mathscr{A} \odot \mathscr{A} \right\|_{\mathsf{op}} = \infty,$$

i.e. the map $\Phi \odot \mathrm{id}_{\mathscr{A}}$ is unbounded under the usual operator norm [19, Prop. 3.5.2]. Due to these limitations, the *completely bounded norm* becomes the natural choice for studying maps between C^*-algebras [42, 88].

Definition 5.4.13. Given a map $\Phi\colon \mathscr{A} \to \mathscr{B}$, define

$$\left\| \Phi \right\|_{\mathsf{cb}} := \sup_n \left\| \mathcal{M}_n(\Phi) \right\|_{\mathsf{op}}.$$

Φ is said to be *completely bounded* if $\left\| \Phi \right\|_{\mathsf{cb}w^*}$ is finite. When restricted to the space of completed bounded operators, the map $\left\| \cdot \right\|_{\mathbf{cb}}$ is called the *completely bounded norm*. It can equivalently be written as

$$\left\| \Phi \right\|_{\mathsf{cb}} = \sup_{\mathscr{C}} \left\| \mathrm{id}_{\mathscr{C}} \,\check{\otimes}\, \Phi \right\|_{\mathsf{op}} = \sup_n \left\| \mathrm{id}_{\mathcal{M}_n} \,\check{\otimes}\, \Phi \right\|_{\mathsf{op}},$$

where the supremum ranges over all C^*-algebras $\mathscr{C}$ [92, Introduction, p. 4].

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 5.4.14. Let $\Phi : \mathscr{N}_1 \to \mathscr{N}_2$ be a normal map between W^*-algebras, define,

$$\|\Phi\|_{\mathrm{cb}w^*} = \sup_{\mathscr{M}} \|\mathrm{id}_{\mathscr{M}} \overline{\otimes} \Phi\|_{\mathrm{op}},$$

where the supremum ranges over all W^*-algebras $\mathscr{M}$. Φ is said to be W^* *completely bounded* if $\|\Phi\|_{\mathrm{cb}w^*}$ is finite. When restricted to the space of W^* completed bounded operators, the map $\|\cdot\|_{\mathbf{cb}w^*}$ is called the W^* *completely bounded norm*.

Note that, in the definition above, positive definiteness is direct. Positive scalability and the triangle inequality (see Definition A.3.1) follow, respectively, from the following properties of *supremum* (see [126, Chapter 2, Section 8-9]):

1. Let $\alpha \in \mathbb{R}$, and let $A \subset \mathbb{R}$ be a non-empty set possessing a *supremum*. Define $\alpha A := \{\,\alpha \cdot a \mid a \in A\,\}$. If $\alpha \geq 0$, then

$$\sup(\alpha A) = \alpha \sup A.$$

2. For any two non-empy subsets $A, B \subset \mathbb{R}$, each posessing a *supremum*, let $A + B := \{\,a + b \mid a \in A,\ b \in B\,\}$. It holds that

$$\sup(A + B) = \sup A + \sup B.$$

Our motivation for the use of this norm instead of the completely bounded 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. 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 5.4.15. *Given normal completely positive maps Φ and Ψ between W^*-algebras, it holds that*

$$\|\Phi \overline{\otimes} \Psi\|_{op} \leq \|\Phi\|_{op} \|\Psi\|_{op}.$$

Proof. It follows directly from Theorem 5.3.30 and Proposition 5.3.33. $\qquad\square$

Corollary 5.4.16. *Given a normal completely positive map Φ between W^*-algebras, it holds that*

$$\|\Phi\|_{cbw^*} \leq \|\Phi\|_{op}.$$

Proof. It follows from Proposition 5.4.15 and the definition of W^* completely bounded norm (Definition 5.4.14). $\square$

Proposition 5.4.17. *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. We reason as follows:

$\|\Phi \cdot \Psi\|_{\mathsf{cb}}$

$= \sup_{\mathscr{M}} \{\|\mathrm{id}_{\mathscr{M}} \overline{\otimes} (\Phi \cdot \Psi)\|_{\mathsf{op}} \}$

$\leq \sup_{\mathscr{M}} \{\|\mathrm{id}_{\mathscr{M}} \overline{\otimes} \Phi\|_{\mathsf{op}} \cdot \|\mathrm{id}_{\mathscr{M}} \overline{\otimes} \Psi\|_{\mathsf{op}} \}$ $\qquad$ (submultiplicativity of $\|\cdot\|_{\mathsf{op}}$ w.r.t composition)

$= \sup_{\mathscr{M}} \{\|\mathrm{id}_{\mathscr{M}} \overline{\otimes} \Phi\|_{\mathsf{op}} \} \cdot \sup_{\mathscr{M}}, \{\|\mathrm{id}_{\mathscr{M}} \overline{\otimes} \Psi\|_{\mathsf{op}} \}$ $\qquad$ ($\cdot$ distributes over $\sup$)

$\leq \sup_{\mathscr{M}_1} \{\|\mathrm{id}_{\mathscr{M}_1} \overline{\otimes} \Phi\|_{\mathsf{op}} \} \cdot \sup_{\mathscr{M}_2}, \{\|\mathrm{id}_{\mathscr{M}_2} \overline{\otimes} \Psi\|_{\mathsf{op}} \}$ $\qquad$ ($A \subset B \implies \sup A \leq \sup B$)

$= \|\Phi\|_{\mathsf{cb}} \cdot \|\Psi\|_{\mathsf{cb}}$

$\square$

Proposition 5.4.18. *Given a W^* completely bounded normal map Φ between W^*-algebras, it holds that*

$$\|\Phi \overline{\otimes} \mathrm{id}\|_{cbw^*} \leq \|\Phi\|_{cbw^*} \quad \textit{and} \quad \|\mathrm{id} \overline{\otimes} \Phi\|_{cbw^*} \leq \|\Phi\|_{cbw^*}.$$

Proof. By Proposition 5.3.35 and Proposition 5.3.13, 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 5.4.16, we obtain $\|\mathrm{sw}\|_{\mathsf{cb}w^*} \leq \|\mathrm{sw}\|_{\mathsf{op}} = 1$. Thus,

$\|\Phi \overline{\otimes} \mathrm{id}\|_{\mathsf{cb}w^*}$

$\leq \|\mathrm{sw}\|_{\mathsf{cb}w^*} \|\Phi \overline{\otimes} \mathrm{id}\|_{\mathsf{cb}w^*}$

$\leq \|\mathrm{sw} \cdot \Phi \overline{\otimes} \mathrm{id}\|_{\mathsf{cb}w^*}$ $\qquad$ ($\|\cdot\|_{\mathsf{cb}w^*}$ is submultiplicative w.r.t composition)

$= \|\mathrm{id} \overline{\otimes} \Phi\|_{\mathsf{cb}w^*}.$

At last, we need to prove that

$$\|\mathrm{id} \,\overline{\otimes}\, \Phi\|_{\mathrm{cb}w^*} \leq \|\Phi\|_{\mathrm{cb}w^*} ,$$

which follows direcly from the definition of the norm (Definition 5.4.14) and the fact that the associator, α, is an isometry with respect to the operator norm (Proposition 5.3.34).

$\square$

Proposition 5.4.19. *Given W^* completely bounded normal maps $\Phi : \mathscr{M} \to \mathscr{N}_1$ and $\Psi : \mathscr{M} \to \mathscr{N}_2$ between W^*-algebras, it holds that*

$$\|\langle \Phi, \Psi \rangle\|_{cbw^*} \leq \max\{\|\Phi\|_{cbw^*}, \|\Psi\|_{cbw^*}\}.$$

Proof. By Corollary 5.3.37 and Definition 5.3.14 we have

$$
\begin{aligned}
\|\langle \Phi, \Psi \rangle\|_{\mathrm{cb}w^*} &= \sup_{\mathscr{T}} \|\mathrm{id}_{\mathscr{T}} \,\overline{\otimes}\, \langle \Phi, \Psi \rangle\|_{\mathrm{op}} \\
&= \sup_{\mathscr{T}}\{\|\mathrm{dist}\|_{\mathrm{op}} \|\mathrm{id}_{\mathscr{T}} \,\overline{\otimes}\, \langle \Phi, \Psi \rangle\|_{\mathrm{op}}\} && \text{(Corollary 5.3.37)} \\
&\leq \sup_{\mathscr{T}} \|\mathrm{dist} \cdot \mathrm{id}_{\mathscr{T}} \,\overline{\otimes}\, \langle \Phi, \Psi \rangle\|_{\mathrm{op}} && (\|\cdot\|_{\mathrm{op}} \text{ is submultiplicative w.r.t composition)} \\
&\leq \sup_{\mathscr{T}} \|\langle \mathrm{id}_{\mathscr{T}} \overline{\otimes} \Phi, \mathrm{id}_{\mathscr{T}} \overline{\otimes} \Phi \rangle\|_{\mathrm{op}} \\
&= \max\{\sup_{\mathscr{T}} \|\mathrm{id}_{\mathscr{T}} \overline{\otimes} \Phi\|_{\mathrm{op}}, \sup_{\mathscr{T}} \|\mathrm{id}_{\mathscr{T}} \overline{\otimes} \Psi\|_{\mathrm{op}}\} && \text{(Definition 5.3.14)} \\
&\leq \max\{\|\Phi\|_{\mathrm{cb}w^*}, \|\Psi\|_{\mathrm{cb}w^*}\}.
\end{aligned}
$$

$\square$

Theorem 5.4.20. $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathrm{op}}$ *is a symmetric monoidal* Met-*category with binary coproducts in which the tensor product is non-expansive.*

Proof. Firstly, note that the copairing in $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathrm{op}}$ corresponds to the pairing in $\mathsf{W}^*_{\mathsf{CPSU}}$. As a result, by proof of Proposition 5.4.2 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. $\|(\Phi - \Phi')\Psi\|_{\mathrm{cb}w^*} \leq \|\Phi - \Phi'\|_{\mathrm{cb}w^*}$ and $\|\Phi'(\Psi - \Psi')\|_{\mathrm{cb}w^*} \leq \|\Psi - \Psi'\|_{\mathrm{cb}w^*}$. 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 5.4.16 and Proposition 5.3.6 the inequalities hold.

2. $\|\Phi \overline{\otimes} \mathrm{id}\|_{\mathrm{cb}w^*} \leq \|\Phi\|_{\mathrm{cb}w^*}$ and $\|\mathrm{id} \overline{\otimes} \Phi\|_{\mathrm{cb}w^*} \leq \|\Phi\|_{\mathrm{cb}w^*}$. It follows directly from Proposition 5.4.18.

3. $\|\langle \Phi - \Phi', \Psi - \Psi'\rangle\|_{\mathrm{cb}w^*} \leq \max\{\|\Phi - \Phi'\|_{\mathrm{cb}w^*}, \|\Psi - \Psi'\|_{\mathrm{cb}w^*}\}$. It follows directly from Proposition 5.4.19.

$\square$

5.5   Examples

We now illustrate the use of (first-order) λ-calculus with conditionals for describing quantum programs. To this effect, we consider a type $\mathtt{qbit}$ 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, $q : \mathbb{I} \oplus \mathbb{I} \to \mathtt{qbit}$, the measurement of a qubit, $meas : \mathtt{qbit} \to \mathbb{I} \oplus \mathbb{I}$, and pre-determined sets of operations on n-qubits, $U, CPTP : \mathtt{qbit}, \dots, \mathtt{qbit} \to \mathtt{qbit}^{\otimes n}$. The former includes unitary operations, as the Hadamard gate $H : \mathtt{qbit} \to \mathtt{qbit}$, the not-gate $X : \mathtt{qbit} \to \mathtt{qbit}$, and the cnot-gate $CNOT : \mathtt{qbit}, \mathtt{qbit} \to \mathtt{qbit}^{\otimes 2}$, and the latter set includes operations such as dephasing with probability p, $D_p : \mathtt{qbit}, \mathtt{qbit} \to \mathtt{qbit}^{\otimes 2}$. We consider as well a pre-determined set of quantum states $|\psi\rangle : \mathbb{I} \to \mathtt{qbit}$ and a discard operation $\mathrm{disc} : \mathtt{qbit} \to \mathbb{I}$.

$\mathcal{K}(\mathrm{CPS})$ forms a model of the metric λ-theory for quantum computation via the following interpretation: $[\![\mathbb{I}]\!] = (1, \mathrm{id})$, $[\![\mathtt{qbit}]\!] = (2, \mathrm{id})$, $[\![q]\!]\left(\left(\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right)\right) = \left(\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right)$, $[\![|\psi\rangle]\!](1) = |\psi\rangle\langle\psi|$, $[\![meas]\!](\rho) = \mathrm{id} \oplus \mathrm{id}(\rho)$, and $[\![\mathrm{disc}]\!](\rho) = \mathsf{Tr}(\rho)$. For unitary operations U we define $[\![U]\!] = U\rho U^\dagger$. For completely positive trace-preserving operators $CPTP$, defined as $CPTP(\rho) = \sum_i K_i \rho K_i^\dagger$, we define $[\![CPTP]\!] = CPTP(\rho)$.

Let us now apply this machinery to two well-known problems in quantum computation and quantum information.

5.5.1   Quantum state discrimination

Example 5.5.1 (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 $R_{y,\theta} :$ qbit $\rightarrow$ qbit, representing a single-qubit rotation by an angle θ around the y-axis, this process is described by the following λ-term:

CoinToss $= - \triangleright meas(R_{y,\theta}(|0\rangle)) : \mathbb{I} \oplus \mathbb{I}$

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 [41] where

$$
U_1(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}
$$

$$
U_2(\phi, \lambda) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{pmatrix}
$$

$$
U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{pmatrix}
$$

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 $|\psi\rangle$.

$$
\begin{aligned}
U_3(\theta, \phi, \lambda) |\psi\rangle &= U_3(\theta, \phi, \lambda) \left(\cos(\alpha/2) |0\rangle + e^{i\beta}\sin(\alpha/2) |1\rangle \right) \\
&= \left(\cos(\alpha/2)\cos(\theta/2) - e^{i(\lambda+\beta)}\sin(\alpha/2)\sin(\theta/2) \right) |0\rangle \\
&\quad + \left(e^{i\phi}\cos(\alpha/2)\sin(\theta/2) + e^{i(\beta+\lambda+\phi)}\sin(\alpha/2)\cos(\theta/2) \right) |1\rangle
\end{aligned}
$$

Designating $U_3(\theta, \phi, \lambda) |\psi\rangle = a |0\rangle + b |1\rangle$, one has

$$
\begin{aligned}
aa^* &= |\cos(\alpha/2)\cos(\theta/2) - e^{i(\lambda+\beta)}\sin(\alpha/2)\sin(\theta/2)|^2 \\
&= \cos^2(\alpha/2)\cos^2(\theta/2) - 2\cos(\beta+\lambda)\cos(\alpha/2)\cos(\theta/2)\sin(\alpha/2)\sin(\theta/2) \\
&\quad + \sin(\alpha/2)^2\sin^2(\theta/2) \\
&= \cos^2(\alpha/2)\cos^2(\theta/2) + \sin^2(\alpha/2)\sin^2(\theta/2) - 1/2\cos(\beta+\lambda)\sin(\alpha)\sin(\theta)
\end{aligned}
$$

$$= \cos^2((\theta + \alpha)/2) + (1/2)(1 - \cos(\beta + \lambda))\sin(\alpha)\sin(\theta)$$

$$a^*b = \big(\cos(\alpha/2)\cos(\theta/2) - e^{-i(\lambda+\beta)}\sin(\alpha/2)\sin(\theta/2)\big)\big(e^{i\phi}\cos(\alpha/2)\sin(\theta/2)$$

$$+ e^{i(\beta+\lambda+\phi)}\sin(\alpha/2)\cos(\theta/2)\big)$$

$$= (1/2)\big(e^{i\phi}\cos^2(\alpha/2)\sin(\theta) + e^{i(\beta+\lambda+\phi)}\sin(\alpha)\cos^2(\theta/2) - e^{-i(\beta+\lambda-\phi)}\sin(\alpha)\sin^2(\theta/2)$$

$$- e^{i\phi}\sin^2(\alpha/2)\sin(\theta/2)\big)$$

$$= (1/2)\big(e^{i\phi}\cos(\alpha)\sin(\theta) + \sin(\alpha)\big(e^{i(\beta+\lambda+\phi)}\cos^2(\theta/2) - e^{-i(\beta+\lambda-\phi)}\sin^2(\theta/2)\big)\big)$$

Then, we calculate the vector Bloch of $U_3(\theta, \phi, \lambda)\,|\psi\rangle$,

$$x = 2\mathsf{Im}\,(a^*b) = \cos(\phi)\cos(\alpha)\sin(\theta) + \sin(\alpha)\big(\cos(\beta + \lambda + \phi)\cos^2(\theta/2)$$

$$- \cos(\beta + \lambda - \phi)\sin^2(\theta/2)\big)$$

$$y = 2\mathsf{Re}\,(a^*b) = \sin(\phi)\cos(\alpha)\sin(\theta) + \sin(\alpha)\big(\sin(\beta + \lambda + \phi)\cos^2(\theta/2) \qquad (5.6)$$

$$+ \sin(\beta + \lambda - \phi)\sin^2(\theta/2)\big)$$

$$z = 2aa^* - 1 = 2\cos^2((\theta + \alpha)/2) + (1 - \cos(\beta + \lambda))\sin(\alpha)\sin(\theta) - 1$$

As a result, we have,

$$\big\|U_3(\theta, 0, 0)\,|\psi\rangle\langle\psi|\,U_3(\theta, 0, 0)^\dagger - U_3(\theta + \epsilon, 0, 0)\,|\psi\rangle\langle\psi|\,U_3(\theta + \epsilon, 0, 0)^\dagger\big\|_1$$

$$= \|(\cos(\alpha)(\sin(\theta) - \sin(\theta + \epsilon)) + \sin(\alpha)\cos\beta\,(\cos(\theta) - \cos(\theta + \epsilon))\,, 0,$$

$$2(\cos^2((\theta + \alpha)/2) - \cos^2((\theta + \epsilon + \alpha)/2)) + \sin(\alpha)(\sin(\theta) - \sin(\theta + \epsilon))$$

$$(1 - \cos(\beta)))\|_2$$

$$\leq \|(\cos(\alpha)\epsilon + \sin(\alpha)\cos(\beta)\epsilon, 0, 2\epsilon + \sin(\alpha)(1 - \cos(\beta))\epsilon)\|_2$$

$$= \sqrt{\epsilon^2\big(\cos^2(\alpha) + \sin(2\alpha)\cos(\beta) + \sin^2(\alpha)\cos^2(\beta) + 4 + 4(\sin(\alpha)(1 - \cos(\beta)))}$$

$$\overline{+ \sin^2(\alpha)(1 - 2\cos(\beta) + \cos^2(\beta))\big)}$$

$$= \epsilon\sqrt{1 + 4\sin(\alpha) + 4 + 2\cos^2(\beta)\sin^2(\alpha)\cos(\beta)(\sin(2\alpha) - 4\sin(\alpha) - 2\sin^2(\alpha))}$$

$$\leq \epsilon\sqrt{(9 + 2 + 6)} = \sqrt{17}\epsilon$$

The first inequality arises from both functions $\cos$ and $\sin$ being Lipschitz continuous. Attending to Theorem 5.2.39, it follows that

$$\|R_{y,\theta} - R_{y,\theta+\epsilon}\|_\diamond \leq \sqrt{17}\epsilon$$

Using our metric deductive system, we can easily conclude that **CoinToss** $=_{\sqrt{17}\epsilon}$ **CoinToss**$^\epsilon$, where **CoinToss**$^\epsilon$ is the the judgement that results from replacing $R_{y,\theta}$ by $R_{y,\theta+\epsilon}$.

Example 5.5.2 (Quantum state discrimination). Quantum state discrimination is a pivotal challenge in quantum communications [8, 119] and quantum cryptography [53]. 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 [8]. 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:

$$\textbf{StatePreparation} = b : \mathbb{I} \oplus \mathbb{I} \triangleright \mathsf{case}\, b\, \{\mathrm{inl}_{\mathbb{B}}(x) \Rightarrow |\psi_0\rangle \,;\, \mathrm{inr}_{\mathbb{A}}(y) \Rightarrow |\psi_1\rangle\} : \texttt{qbit}$$

$$\textbf{HMeasure} = x : \texttt{qbit} \triangleright \mathit{meas}(U(x)) : \mathbb{I} \oplus \mathbb{I}$$

$$\textbf{Discrimination} = -\, \triangleright \textbf{HMeasure}[\textbf{StatePreparation}[\textbf{CoinToss}(*)/b]/x] : \mathbb{I} \oplus \mathbb{I}$$

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

$$U_1(\lambda)|\psi\rangle = \cos\left(\frac{\alpha}{2}\right)|0\rangle + e^{i(\beta+\lambda)}\sin\left(\frac{\alpha}{2}\right)|1\rangle.$$

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

$$\left\| U_1(\lambda)|\psi\rangle\langle\psi| U_1(\lambda)^\dagger - U_1(\lambda + \epsilon_1)|\psi\rangle\langle\psi| U_1(\lambda + \epsilon_1)^\dagger \right\|_1$$
$$= \left\| (\sin(\alpha)(\cos(\beta + \lambda) - \cos(\beta + \lambda + \epsilon_1)), \sin(\alpha)(\sin(\beta + \lambda) - \sin(\beta + \lambda + \epsilon_1)), 0) \right\|_2$$
$$\leq \left\| (\cos(\beta + \lambda) - \cos(\beta + \lambda + \epsilon_1), \sin(\beta + \lambda) - \sin(\beta + \lambda + \epsilon_1), 0) \right\|_2$$
$$\leq \left\| (\epsilon_1, \epsilon_1, 0) \right\|_2$$
$$= \sqrt{2}\epsilon_1$$

Attending to , it follows that

$$\left\| U_1(\lambda) - U_1(\lambda + \epsilon_1) \right\|_\diamond \leq \sqrt{2}\epsilon_1.$$

Given the erroneous R_y gate is bounded by $3\epsilon_2$, we observe that the R_y gate amplifies errors more significantly than the R_z gate for errors of the same magnitude.

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 $U =_{\sqrt{2}\epsilon_1 + \sqrt{17}\epsilon_2} U^{\epsilon_1,\epsilon_2}$. This equation implies that if $\epsilon_1 = \epsilon_2$, the erroneous U_3 gate contributes almost thrice as much as the erroneous U_1 gate to the upper bound on the total distance between the ideal and erroneous unitary.

Next, we deduce **HMeasure** $=_{\sqrt{2}\epsilon_1+\sqrt{17}\epsilon_2}$ **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

$$\textbf{Discrimination} =_{\sqrt{2}\epsilon_1+2\sqrt{17}\epsilon_2} \textbf{Discrimination}^{\epsilon_1,\epsilon_2},$$

where **Discrimination**$^{\epsilon_1,\epsilon_2}$ denotes the judgement that results from replacing **HMeasure** by **HMeasure**$^{\epsilon_1,\epsilon_2}$ and **CoinToss** by **CoinToss**$^{\epsilon_2}$.

Observe that the distance between the ideal and erroneous quantum state discrimination tends to 0 as ϵ_1 and ϵ_2 tend to zero, as expected. Additionally, since an erroneous U_3 gate affects both **HMeasure** and **CoinToss**, we find that when ϵ_1 and ϵ_2 are of equal magnitude, the erroneous U_3 contributes almost six times as much as the erroneous U_1 to the upper bound on the total distance between the ideal and erroneous unitary.

5.5.2   Quantum teleportation protocol

[11] 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 [18, 57, 69].

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.
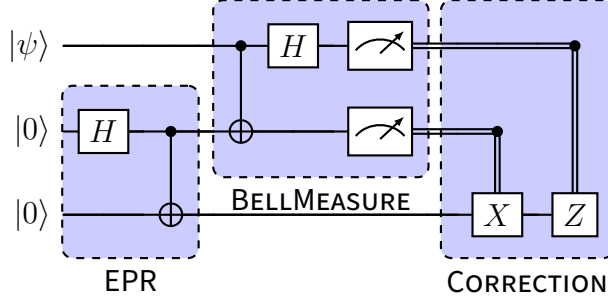
Figure 7: Quantum Teleportation Protocol

We first describe each of the rectangles filled in blue separately, and using standard quantum gate operations, namely $H :$ qbit $\rightarrow$ qbit, $X :$ qbit $\rightarrow$ qbit, $Z :$ qbit $\rightarrow$ qbit, and $CNOT :$ qbit, qbit $\rightarrow$ qbit $\otimes$ qbit:

EPR $= CNOT(H\,|0\rangle, |0\rangle) :$ qbit $\otimes$ qbit

BellMeasure $= q_1 :$ qbit, $q_2 :$ qbit $\triangleright$ pm $CNOT(q_1, q_2)$ to $x \otimes y$.

$$meas(H(x)) \otimes meas(y) : (\mathbb{I} \oplus \mathbb{I}) \otimes (\mathbb{I} \oplus \mathbb{I})$$

Correction $= q :$ qbit, $x : \mathbb{I} \oplus \mathbb{I}, y : \mathbb{I} \oplus \mathbb{I} \triangleright$

$$\text{case } x \begin{cases} \text{inl}(x_0) \Rightarrow x_0 \text{ to } * . \text{case } y \begin{cases} \text{inl}(y_0) \Rightarrow y_0 \text{ to } * . q; \\ \text{inr}(y_1) \Rightarrow y_1 \text{ to } * . X \end{cases} ; \\ \text{inr}(x_1) \Rightarrow x_1 \text{ to } * . \text{case } y \begin{cases} \text{inl}(y_0) \Rightarrow y_0 \text{ to } * . Z(q); \\ \text{inr}(y_1) \Rightarrow y_1 \text{ to } * . Z(X(q)) \end{cases} \end{cases} : \text{qbit}$$

Designating the qubit to be teleported as qb_0, one then describes the teleportation procedure in λ-calculus as follows:

QTP $= qb_0 :$ qbit $\triangleright$ pm **EPR** to $qb_1 \otimes qb_2$.

pm **BellMeasure** $[qb_0/q_1, qb_1/q_2]$ to $c_0 \otimes c_1$.

Correction $[qb_2/q, c_0/x, c_1/y] :$ qbit

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 $U_1(\lambda + \epsilon_1)$ and $U_3(\theta, \phi + \epsilon_2, \lambda + \epsilon_3)$, respectively. Recall from the previous example that we established the

upper bound $\|U_1(\lambda) - U_1(\lambda + \epsilon_1)\|_\diamond \leq \sqrt{2}\epsilon_1$. The Hadamard gate, H, is the composition $U_3(\pi/2, 0, 0) \cdot U_1(\pi)$. Recall Equation 5.6, we calculate,

$$\left\| U_3(\pi/2,0,0)\,|\psi\rangle\,\langle\psi|\,U_3(\pi/2,0,0)^\dagger - U_3(\pi/2,\epsilon_2,\epsilon_3)\,|\psi\rangle\,\langle\psi|\,U_3(\pi/2,\epsilon_2,\epsilon_3)^\dagger \right\|_1$$

$$= \|(\cos(\alpha)(1 - \cos\epsilon_2) + \sin(\alpha)(1/2(\cos(\beta + \epsilon_2 + \epsilon_3) - \cos(\beta - \epsilon_2 + \epsilon_3))),$$

$$\cos(\alpha)(1 - \sin\epsilon_2) + (1/2)\sin(\alpha)(\sin(\beta) - \sin(\beta + \epsilon_2 + \epsilon_3)$$

$$+ \sin(\beta) - \sin(\beta - \epsilon_2 + \epsilon_3)), (\cos(\beta + \epsilon_3) - \cos(\beta))\sin(\alpha))\|_2$$

$$\leq \|(\cos(\alpha)\epsilon_2 + (1/2)\sin(\alpha)(\epsilon_2 + \epsilon_3), \cos(\alpha)\epsilon_2 + (1/2)\sin(\alpha)(\epsilon_2 + \epsilon_3 + |\epsilon_3 - \epsilon_2|),$$

$$\sin(\alpha)\epsilon_3)\|_2$$

$$< \|(\epsilon_2 + (1/2)(\epsilon_2 + \epsilon_3), \epsilon_2 + (1/2)(\epsilon_2 + \epsilon_3 + |\epsilon_3 - \epsilon_2|), \epsilon_3)\|_2$$

$$\leq 3\epsilon_2 + 2\epsilon_3 + |\epsilon_2 - \epsilon_3|$$

Attending to Theorem 5.2.39, 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

$$H =_{\sqrt{2}\epsilon_1 + 3\epsilon_2 + 2\epsilon_3 + |\epsilon_2 - \epsilon_3|} H^{\epsilon_1, \epsilon_2, \epsilon_3}. \tag{5.7}$$

The gate X can be implemented as $U_3(\pi, 0, \pi)$. Given Equation 5.6, we compute,

$$\left\| U_3(\pi,0,\pi)\,|\psi\rangle\,\langle\psi|\,U_3(\pi,0,\pi)^\dagger - U_3(\pi,\epsilon_2,\pi+\epsilon_3)\,|\psi\rangle\,\langle\psi|\,U_3(\pi,\epsilon_2,\pi+\epsilon_3)^\dagger \right\|_1$$

$$= \|(\sin(\alpha)(\cos(\beta + \epsilon_3 - \epsilon_2) - \cos(\beta)), \sin(\alpha)(\sin(\beta) - \sin(\beta + \epsilon_3 - \epsilon_2)), 0)\|_2$$

$$\leq \|(|\epsilon_3 - \epsilon_2|, |\epsilon_3 - \epsilon_2|, 0)\|_2$$

$$= \sqrt{2}|\epsilon_3 - \epsilon_2|$$

Recall that the gate U_3 is defined as

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{pmatrix}.$$

Thus, when $\epsilon_2 = \epsilon_3 = \epsilon$, the unitaries $U_3(\pi, 0, \pi)$ and $U_3(\pi, \epsilon_2, \pi + \epsilon_3)$ differ only by a global phase factor $e^{i\epsilon}$. Therefore, it is reasonable for the distance between X and its erroneous version to tend to 0 as $|\epsilon_3 - \epsilon_2|$ converges to 0.

127

Considering Theorem 5.2.39, 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

$$X =_{\sqrt{2}|\epsilon_3 - \epsilon_2|} X^{\epsilon_2, \epsilon_3}. \tag{5.8}$$

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

$$Z =_{\sqrt{2}\epsilon_1} Z^{\epsilon_1}. \tag{5.9}$$

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 (5.8) and (5.9) and our metric deductive system we have that

$$\textbf{Correction} =_{\sqrt{2}(\epsilon_1 + |\epsilon_3 - \epsilon_2|)} \textbf{Correction}^{\epsilon_1, \epsilon_2, \epsilon_3}. \tag{5.10}$$

We observe, as expected and in light of our previous remark regarding the upper bound on the distance between X and its erroneous version, that when both the error of the U_1 gate, ϵ_1, and the absolute difference between the errors of the U_1 gate, $|\epsilon_3 - \epsilon_2|$, tend to 0, the distance between **Correction** and **Correction**$^{\epsilon_1, \epsilon_2, \epsilon_3}$ also tends to 0. Moreover, note that when $\epsilon_1 = |\epsilon_2 + \epsilon_3|$, both gates contribute equally to the upper bound on the distance between **Correction** and its erroneous version.

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 applying 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 $\rho = |\alpha|^2 |0\rangle \langle 0| + \alpha\overline{\beta} |0\rangle \langle 1| + \overline{\alpha}\beta |1\rangle \langle 0| + |\beta|^2 |1\rangle \langle 1|$. Using these Kraus operators, it is possible to easily verify that after applying the dephasing channel with probability p, the resulting operator ρ' is given by:

$$\rho' = D_p(\rho) = D_0\rho D_0^\dagger + D_1\rho D_1^\dagger = |\alpha|^2 |0\rangle \langle 0| + (1-p)\alpha\overline{\beta} |0\rangle \langle 1| + (1-p)\overline{\alpha}\beta |1\rangle \langle 0| + |\beta|^2 |1\rangle \langle 1|$$

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:

$$\|\mathrm{id}(\rho) - D_p(\rho)\|_1$$

$$\left\|\alpha\overline{\beta}\,|0\rangle\,\langle 1| + \overline{\alpha}\beta\,|1\rangle\,\langle 0| - (1-p)\alpha\overline{\beta}\,|0\rangle\,\langle 1| - (1-p)\overline{\alpha}\beta\,|1\rangle\,\langle 0|\right\|_1$$

$$= p \cdot \left\|\alpha\overline{\beta}\,|0\rangle\,\langle 1| + \overline{\alpha}\beta\,|1\rangle\,\langle 0|\right\|_1$$

$$= p \cdot \mathsf{Tr}\left(\sqrt{\left(\alpha\overline{\beta}\,|0\rangle\,\langle 1| + \overline{\alpha}\beta\,|1\rangle\,\langle 0|\right)^2}\right)$$

$$= p \cdot \mathsf{Tr}\left(\sqrt{|\alpha|^2|\beta|^2(|0\rangle\,\langle 0| + |1\rangle\,\langle 1|)}\right)$$

$$= 2 \cdot p \cdot |\alpha||\beta|$$

$$\leq p$$

The last step arises from the fact that the expression is maximized when $|\alpha| = |\beta| = 1/\sqrt{2}$. Considering Theorem 5.2.40, it holds that

$$\|\mathrm{id} - D_p\|_\diamond \leq \sqrt{2p}$$

Consequently, we can postulate the following axiom:

$$\mathrm{id} =_{\sqrt{2p}} D_p. \tag{5.11}$$

Note that the upper bound on the distance is directly proportional to the probability of *dephasing*, p, as expected.

If a dephasing channel acts on the first qubit of the EPR state, we are interested in reasoning about the following judgements:

$$\mathbf{EPR} = (\mathrm{id} \otimes \mathrm{id})(\mathit{CNOT}(H\,|0\rangle\,, |0\rangle)) : \mathtt{qbit} \otimes \mathtt{qbit}$$

$$\mathbf{EPR}^{\epsilon_1,\epsilon_2,\epsilon_3,p} = (D_p \otimes \mathrm{id})(\mathit{CNOT}(H^{\epsilon_1,\epsilon_2,\epsilon_3}\,|0\rangle\,, |0\rangle)) : \mathtt{qbit} \otimes \mathtt{qbit}$$

Given axioms in equations (5.7) and (5.11), using our metric deductive system, we infer that

$$\mathbf{EPR} =_{\sqrt{2}\epsilon_1 + 3\epsilon_2 + 2\epsilon_3 + |\epsilon_2 - \epsilon_3| + \sqrt{2p}} \mathbf{EPR}^{\epsilon_1,\epsilon_2,\epsilon_3,p} \tag{5.12}$$

Once again, we observe that, as expected, when the errors of gates U_1 and U_2, as well as the probability of dephasing p tend to 0, so does the distance between **EPR** and **EPR**$^{\epsilon_1,\epsilon_2,\epsilon_3,p}$. Additionally, we note that when $\epsilon_1 = \epsilon_2$, $\epsilon_1 = \epsilon_3$, or $\epsilon_1 = |\epsilon_2 - \epsilon_3|$, the gate U_3 consistently dominates the upper bound on the distance between **EPR** and **EPR**$^{\epsilon_1,\epsilon_2,\epsilon_3,p}$.

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 [98, 118].

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. Attending to the definition of trace norm for matrices and Equation 5.1, we calculate,

$$\begin{aligned}
&\|\mathrm{id}(\rho) - A_\gamma(\rho)\|_1 \\
&= \Big\||\alpha|^2\,|0\rangle\,|0\rangle + \alpha\overline{\beta}\,|0\rangle\,|1\rangle + \overline{\alpha}\beta\,|1\rangle\,|0\rangle + |\beta|^2\,|1\rangle\,|1\rangle - \big((|\alpha|^2 + \gamma|\beta|^2)\,|0\rangle\langle 0| \\
&\quad + \sqrt{1-\gamma}\,\alpha\overline{\beta}\,|0\rangle\langle 1| + \sqrt{1-\gamma}\,\overline{\alpha}\beta\,|1\rangle\langle 0| + (1-\gamma)|\beta|^2\,|1\rangle\langle 1|\big)\Big\|_1 \\
&= \Big\|\gamma|\beta|^2\,|1\rangle\langle 1| + (1 - \sqrt{1-\gamma})(\alpha\overline{\beta}\,|0\rangle\langle 1| + \overline{\alpha}\beta\,|1\rangle\langle 0|) - \gamma|\beta|^2\,|0\rangle\langle 0|\Big\|_1 \\
&= \mathrm{Tr}\left(\sqrt{\Big(\gamma|\beta|^2\,|1\rangle\langle 1| + (1 - \sqrt{1-\gamma})(\alpha\overline{\beta}\,|0\rangle\langle 1| + \overline{\alpha}\beta\,|1\rangle\langle 0|) - \gamma|\beta|^2\,|0\rangle\langle 0|\Big)^2}\right) \\
&= \mathrm{Tr}\left(\sqrt{\Big((1 - \sqrt{1-\gamma})^2|\alpha|^2|\beta|^2 + \gamma^2|\beta|^4\Big)(|0\rangle\langle 0| + |1\rangle\langle 1|)}\right)
\end{aligned}$$

$$= 2 \cdot \sqrt{(1 - \sqrt{1 - \gamma})^2 |\alpha|^2 |\beta|^2 + \gamma^2 |\beta|^4}$$

$$\leq 2\gamma$$

This final step follows because the expression attains its maximum when $|\beta| = 1$.

Attending to Theorem 5.2.40, it holds that

$$\|\mathrm{id} - A_\gamma\|_\diamond \leq 2\sqrt{\gamma}$$

As a result, we can postulate the following axiom:

$$\mathrm{id} =_{2\sqrt{\gamma}} A_\gamma. \tag{5.13}$$

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**$^\gamma$.

$$\textbf{Id} = qb : \texttt{qbit} \triangleright \mathrm{id}(qb) \tag{5.14}$$

$$\textbf{Id}^\gamma = A_\gamma(q) : \texttt{qbit} \triangleright A_\gamma(qb) \tag{5.15}$$

Consequently the ideal version of teleportation protocol is now defined as follows

$$\textbf{QTP} = qb_0 : \texttt{qbit} \triangleright \texttt{pm } \textbf{EPR} \texttt{ to } qb_1 \otimes qb_2.$$

$$\texttt{pm } \textbf{BellMeasure} \, [qb_0/q_1, qb_1/q_2] \texttt{ to } c_0 \otimes c_1.$$

$$\textbf{Id} \, [\textbf{Correction}/qb] \, [qb_2/q, c_0/x, c_1/y] : \texttt{qbit}$$

Considering the axiom in equation (5.13) and our metric deductive system, it holds that

$$\textbf{Id} =_{2\sqrt{\gamma}} \textbf{Id}^\gamma$$

Similarly to the case of the dephasing channel (Equation 5.11), we observe— as expected— that the upper bound on the distance tends to 0 as the amplitude damping probability γ tends to 0, and reaches its maximum value when $\gamma = 1$. Additionally, for $p = \gamma$, the upper bound on the distance is higher for the amplitude damping channel compared to the dephasing channel. This behavior is expected since the amplitude damping channel not only alters the phase (introducing a dephasing effect) but also the amplitude of the quantum state.

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:

$$T : \ \texttt{qbit} \to \texttt{qbit}$$

$$T = q : \texttt{qbit} \triangleright \mathsf{pm}\, CU(R_{x,\frac{\pi}{2}}(|0\rangle), q) \text{ to } newq \otimes qb.\, disc(newq)$$

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.
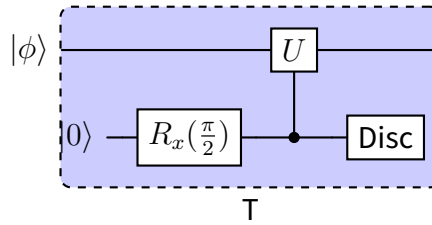
Figure 8: T operation

First, let us verify the result of applying operation T to a quantum state $\rho = |\psi\rangle\langle\psi|$:

$$|\psi\rangle\langle\psi|$$

$$\xmapsto{\ \mathrm{id}\otimes[\![|0\rangle]\!]\ } \quad |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|$$

$$\xmapsto{\ \mathrm{id}\otimes[\![R_{x,\frac{\pi}{2}}]\!]\ } \quad |\psi\rangle\langle\psi| \otimes \frac{1}{2}\left(|0\rangle\langle 0| - i\,|0\rangle\langle 1| + i\,|1\rangle\langle 0| + |1\rangle\langle 1|\right)$$

$$= \frac{1}{2}\left(|\psi\rangle\langle\psi|\,|0\rangle\langle 0| - i\,|\psi\rangle\langle\psi|\,|0\rangle\langle 1| + i\,|\psi\rangle\langle\psi|\,|1\rangle\langle 0| + |\psi\rangle\langle\psi|\,|1\rangle\langle 1|\right)$$

$$\xmapsto{\ [\![CU]\!]\ } \quad \frac{1}{2}\left(|\psi\rangle\langle\psi|\,|0\rangle\langle 0| - i\,|\psi\rangle\langle\psi|\,|0\rangle\langle 1|\,U^\dagger + iU\,|\psi\rangle\langle\psi|\,|1\rangle\langle 0| + U\,|\psi\rangle\langle\psi|\,|1\rangle\langle 1|\,U^\dagger\right)$$

$$\xmapsto{\ \mathrm{id}\otimes[\![disc]\!]\ } \quad \frac{1}{2}\left(|\psi\rangle\langle\psi| + U\,|\psi\rangle\langle\psi|\,U^\dagger\right)$$

Considering X as U, we compute

$$\|\mathrm{id}(\rho) - T(\rho)\|_1$$

$$= \left\| (1/2)\Big((|\alpha|^2 - |\beta|^2)\,|0\rangle\langle 0| + (\alpha\overline{\beta} - \overline{\alpha}\beta)\,|0\rangle\langle 1| + (\overline{\alpha}\beta - \alpha\overline{\beta})\,|0\rangle\langle 1| \right.$$

$$\left. + (|\beta|^2 - |\alpha|^2)\,|1\rangle\langle 1|\Big) \right\|_1$$

$$= (1/2)\mathsf{Tr}\left(\sqrt{\left((|\alpha|^2 - |\beta|^2)\,|0\rangle\langle 0| + (\alpha\overline{\beta} - \overline{\alpha}\beta)\,|0\rangle\langle 1| + (\overline{\alpha}\beta - \alpha\overline{\beta})\,|0\rangle\langle 1| \right.}\right.$$
$$\left.\overline{\left. + (|\beta|^2 - |\alpha|^2)\,|1\rangle\langle 1|\right)^2}\right)$$

$$= (1/2)\mathsf{Tr}\left(\sqrt{\left(((|\alpha|^2 - |\beta|^2)^2 + (\alpha\overline{\beta} - \overline{\alpha}\beta)(\overline{\alpha}\beta - \alpha\overline{\beta}))(|0\rangle\langle 0| + |1\rangle\langle 1|)\right)}\right)$$

$$= (1/2)\mathsf{Tr}\left(\sqrt{\left(((|\alpha|^2 - |\beta|^2)^2 + 2|\alpha|^2|\beta|^2 - (\overline{\alpha})^2\beta^2 - (\overline{\beta})^2\alpha^2)(|0\rangle\langle 0| + |1\rangle\langle 1|)\right)}\right)$$

$$= (1/2)\mathsf{Tr}\left(\sqrt{\left((|\alpha|^4 + |\beta|^4 - 2\mathsf{Re}((\alpha\overline{\beta})^2))(|0\rangle\langle 0| + |1\rangle\langle 1|)\right)}\right)$$

$$= \sqrt{|\alpha|^4 + |\beta|^4 - 2\mathsf{Re}\left((\alpha\overline{\beta})^2\right)}$$

$$\leq 1$$

This last step holds because the expression is maximized when either $|\alpha|$ or $|\beta|$ are equal to one.

Considering Theorem 5.2.40, it holds that

$$\|\mathrm{id} - T\|_\diamond \leq \sqrt{2}$$

Consequently, we postulate the following axiom:

$$\mathrm{id} =_{\sqrt{2}} T. \tag{5.16}$$

In this case we reason about the following λ-terms:

$$\textbf{BellMeasure} = q_1 : \mathtt{qbit}, q_2 : \mathtt{qbit} \rhd \mathsf{pm}\, CNOT(q_1, q_2) \,\text{to}\, x \otimes y.$$
$$meas(H(x)) \otimes meas(y) : (\mathbb{I} \oplus \mathbb{I}) \otimes (\mathbb{I} \oplus \mathbb{I})$$

$$\textbf{BellMeasure}^{\epsilon_1, \epsilon_2, \epsilon_3, T} = q_1 : \mathtt{qbit}, q_2 : \mathtt{qbit} \rhd \mathsf{pm}\, CNOT(q_1, q_2) \,\text{to}\, x \otimes y.$$
$$meas(T(H^{\epsilon_1, \epsilon_2, \epsilon_3}(x))) \otimes meas(T(y)) : (\mathbb{I} \oplus \mathbb{I}) \otimes (\mathbb{I} \oplus \mathbb{I})$$

Attending to the axioms in equations (5.7) and (5.16), via our metric deductive system, we infer that

$$\textbf{BellMeasure} =_{2\sqrt{2} + \sqrt{2}\epsilon_1 + 3\epsilon_2 + 2\epsilon_3 + |\epsilon_2 - \epsilon_3|} \textbf{BellMeasure}^{\epsilon_1, \epsilon_2, \epsilon_3, T} \tag{5.17}$$

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 (5.12), (5.17), (5.10), and (5.14), using our deductive metric system, it follows that

$$\textbf{QTP} =_{3\sqrt{2}\epsilon_1 + 6\epsilon_2 + 4\epsilon_3 + (2 + \sqrt{2})|\epsilon_2 - \epsilon_3| + 2\sqrt{2} + \sqrt{2p} + 2\sqrt{\gamma}} \textbf{QTP}^{\epsilon_1, \epsilon_2, \epsilon_3, T, p, \gamma}$$

This example demonstrates that our metric deductive system enables modular reasoning about approximate equivalence, offering better scalability and would likely be more challenging to achieve through purely semantic methods.

Chapter 6

Future work

In this work, we extend [36] by introducing a metric equation for conditionals. We prove its soundness and completeness, and illustrate its syntactic and semantic applicability across several domains. Nevertheless, much remains to be done.

Generalizing to quantales This work can be generalized to other quantales, such as the Boolean, ultrametric, and Gödel quantales. Indeed, we have already taken initial steps in this direction in [97], where a broader range of quantales is considered; this work was accepted at an international workshop. Readers unfamiliar with quantales may also consult [110] 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 [110]. In this quantale, the associative operation $\otimes$ and the infimum/meet are defined as follows:

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

where $I, J \in \mathcal{P}(\Sigma^*)$ and $\mathbin{+\!\!+}$ 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 $\varepsilon \mathbin{+\!\!+} s = s = s \mathbin{+\!\!+} \varepsilon$). 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.

As an illustrative example of the aforementioned generalisation to quantales, consider, for example, the Boolean quantale, where equations are labelled by elements of $\{0, 1\}$. The

judgement $\Gamma \triangleright v =_1 w : \mathbb{A}$ can be interpreted as an inequation $\Gamma \triangleright v \leq w : \mathbb{A}$, 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, inequational 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 [36]. For the ultrametric quantale, one could investigate ultrametric λ-theories within computational paradigms such as the guarded λ-calculus [12] and functional reactive programming [71]. Finally, the Gödel quantale, which underlies fuzzy logic [40], gives rise to what we refer to as *fuzzy inequations*.

Closing quantum (first-order) categories Another possible direction stems from the fact that, as previously mentioned, the quantum categories discussed in Section 5.4 are not closed. In [36], 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 $\mathcal{K}(\mathrm{CPS})$ and $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$.

The completely bounded norm and the W^* completely bounded norm In this work, our focus is limited to showing that the metric induced by the W^* completely bounded norm makes $(\mathsf{W}^*_{\mathsf{CPSU}})^{\mathsf{op}}$ into a first-order model. However, as previously noted, another norm—the *completely bounded* norm—is widely used in the context of C^*-algebras and comes equipped with established results that could simplify distance computations between programs. A natural next step would be to determine whether the completely bounded norm itself constitutes a suitable metric, *i.e.*, if it satisfies $\|\mathrm{id}\overline{\otimes}\Phi\|_{\mathsf{cb}} \leq \|\Phi\|_{\mathsf{cb}}$, for any completely bounded normal map Φ between W^*-algebras (recall Definition 5.4.13). Of course, we aim as well to establish additional results regarding the W^* completely bounded norm that simplify distance computations between morphisms. That would, for instance, allow us to reason about quantum walks on a line [116].

A metric on Selinger's Q by an embedding into $\mathcal{K}(\mathrm{CPS})$ As previously mentioned in [99], Selinger introduced a first-order functional quantum language, QPL, whose denotational semantics is given by the distributive symmetric monoidal category Q. Here, Selinger works with *vectors* (*i.e.*, direct sums) of square matrices, and extends the standard notions of *positivity* and *trace* to matrix tuples.

Another direction for future work would be to define a functor $F \colon \mathsf{Q} \to \mathcal{K}(\mathrm{CPS})$ and use it to induce a norm on the morphisms Φ in Q via $\|\Phi\| \coloneqq \|F(\Phi)\|_{\diamond}$, taking advantage of the fact

that $\mathcal{K}(\text{CPS})$ is a model of our calculus.

A metric on Q **by** W^***-algebras** In [23], the author established an equivalence of categories $Q \simeq (\text{FdW}^*_{\text{CPSU}})^{\text{op}}$. Using this equivalence, along the same lines as above, we aim to induce an alternative norm on Q by assigning to each superoperator $\Phi \in Q$ the norm of its corresponding map $\Phi^* \in (\text{FdW}^*_{\text{CPSU}})^{\text{op}}$.

Quantum graded λ**-calculus** [35] extends [36] by introducing a sound and complete quantalic equational system —which includes the metric quantale— for a λ-calculus with graded modal types, allowing multiple uses of the same resource. Since the *op. cit.* [35] 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 [1], or estimating an unknown parameter across n copies of a quantum state in quantum metrology [51, 128].

Bibliography

[1] 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. `doi:10.1103/PhysRevA.71.032338`.

[2] Charalambos D. Aliprantis and Kim C. Border. *Infinite Dimensional Analysis*. Springer-Verlag, 2006. `doi:10.1007/3-540-29587-9`.

[3] Huzihiro Araki. *Mathematical Theory of Quantum Fields*. Oxford University Press, 1999.

[4] Alain Aspect, Jean Dalibard, and Gérard Roger. Experimental test of bell's inequalities using time-varying analyzers. *Physical review letters*, 49(25):1804, 1982. `doi:10.1103/PhysRevLett.49.1804`.

[5] Krishna B. Athreya and Soumendra N. Lahiri. *Measure Theory and Probability Theory*. Springer Science & Business Media, 2006.

[6] Steve Awodey and Steve Awodey. *Category Theory*. Oxford Logic Guides. Oxford University Press, second edition, second edition edition, 2010.

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

[8] Stephen Barnett. *Quantum Information*. Oxford University Press, Inc., USA, 2009.

[9] Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva, editors. *Foundations of Probabilistic Programming*. Cambridge University Press, 2020. `doi:10.1017/9781108770750`.

[10] J. S. Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1:195–200, Nov 1964. `doi:10.1103/PhysicsPhysiqueFizika.1.195`.

[11] 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. `doi: 10.1103/PhysRevLett.70.1895`.

[12] Lars Birkedal, Jan Schwinghammer, and Kristian Støvring. A metric model of lambda calculus with guarded recursion. Informal workshop proceedings, Aug 2010.

[13] Vladimir I. Bogachev. *Measure Theory*. Springer, 2007. `doi:10.1007/978-3-540-34514-5`.

[14] Francis Borceux. *Handbook of Categorical Algebra: Volume 1: Basic Category Theory*, volume 1 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 1994. `doi:10.1017/CBO9780511525858`.

[15] Francis Borceux. *Handbook of Categorical Algebra: Volume 2: Categories and Structures*, volume 2 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 1994. `doi:10.1017/CBO9780511525865`.

[16] M. Born and P. Jordan. Zur quantenmechanik. 34(1):858–888, 1925. `doi:10.1007/BF01328531`.

[17] Ola Bratteli and Derek W. Robinson. *Operator Algebras and Quantum Statistical Mechanics 1*. Springer, 1987. `doi:10.1007/978-3-662-02520-8`.

[18] 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. `doi:10.1103/PhysRevLett.81.5932`.

[19] Nathanial Patrick Brown and Narutaka Ozawa. *C*-Algebras and Finite-dimensional Approximations*. American Mathematical Soc., 2008.

[20] 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. `doi:10.1109/TCAD.2020.3032630`.

[21] A Robert Calderbank and Peter W Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098, 1996. `doi:10.1103/PhysRevA.54.1098`.

[22] Bing-yuan Cao, Guo-jun Wang, Shui-li Chen, and Si-zong Guo, editors. *Quantitative Logic and Soft Computing 2010: Volume 2*, volume 82 of *Advances in Intelligent and Soft Computing*. Springer, 2010. `doi:10.1007/978-3-642-15660-1`.

[23] Kenta Cho. Semantics for a Quantum Programming Language by Operator Algebras. *New Generation Computing*, 34(1):25–68, 2016. `doi:10.1007/s00354-016-0204-3`.

[24] Kenta Cho and Abraham Westerbaan. Von Neumann Algebras form a Model for the Quantum Lambda Calculus, 2016. `arXiv:1603.02133`, `doi:10.48550/arXiv.1603.02133`.

[25] Frederic T Chong, Diana Franklin, and Margaret Martonosi. Programming languages and compiler design for realistic quantum hardware. *Nature*, 549(7671):180–187, 2017. `doi:10.1038/nature23459`.

[26] Alonzo Church. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58(2):345–363, 1936. `arXiv:2371045`, `doi:10.2307/2371045`.

[27] Alonzo Church. A Formulation of the Simple Theory of Types. 5(2):56–68, 1940. `doi:10.2307/2266170`.

[28] Alain Connes. *Noncommutative Geometry*. Academic Press, 1995.

[29] John B. Conway. *A Course in Operator Theory*. American Mathematical Society, 2000.

[30] John B. Conway. *A Course in Functional Analysis*, volume 96 of *Graduate Texts in Mathematics*. Springer, 2007. `doi:10.1007/978-1-4757-4383-8`.

[31] Roy L. Crole. *Categories for Types*. Cambridge University Press, 1994. `doi:10.1017/CBO9781139172707`.

[32] 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. `doi:10.1109/LICS.2015.64`.

[33] 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. `doi:10.1007/978-3-662-54434-1_13`.

[34] 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. doi:10.4230/LIPIcs.CSL.2022.16.

[35] 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. doi:10.46298/entics.12299.

[36] 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. doi:10.46298/lmcs-19(4:31)2023.

[37] 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. doi:10.1017/9781108770750.002.

[38] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2 edition, 2002. doi:10.1017/CBO9780511809088.

[39] Karel De Leeuw, Edward F Moore, Claude E Shannon, and Norman Shapiro. Computability by probabilistic machines. automata studies. *The Journal of Symbolic Logic*, 34:183–198, 1956. doi:10.2307/2270759.

[40] K. Denecke, M. Erné, and S. L. Wismath, editors. *Galois Connections and Applications*. Springer Netherlands, 2004. doi:10.1007/978-1-4020-1898-5.

[41] IBM Quantum Documentation. December 2020 product updates. URL: https://quantum.cloud.ibm.com/announcements/en/product-updates/2020-12-01-product-updates.

[42] Edward G. Effros and Zhong-Jin Ruan. *Operator Spaces*. Clarendon Press, 2000.

[43] Samuel Eilenberg and Saunders MacLane. General Theory of Natural Equivalences. *Transactions of the American Mathematical Society*, 58(2):231–294, 1945. doi:10.2307/1990284.

[44] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777, 1935. doi:10.1103/PhysRev.47.777.

[45] Tai-He Fan, Shui-Li Chen, San-Min Wang, and Yong-Ming Li, editors. *Quantitative Logic and Soft Computing 2016: Proceedings of the 4th International Conference on Quantitative Logic and Soft Computing (QLSC2016) Held at Hangzhou, China, 14-17 October, 2016*, volume 510 of *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2017. URL: http://link.springer.com/10.1007/978-3-319-46206-6, doi:10.1007/978-3-319-46206-6.

[46] Fernando Ferreira. O problema da decisão e a máquina universal de turing. URL: https://ebooks.uminho.pt/index.php/uminho/catalog/view/5/8/185.

[47] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7), 1982. doi:10.1007/BF02650179.

[48] 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. doi:10.1103/PhysRevA.86.032324.

[49] Torkel Franzén. *Gödel's Theorem*. A. K. Peters.

[50] 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. doi:10.1017/CBO9780511542725.

[51] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum metrology. *Phys. Rev. Lett.*, 96:010401, Jan 2006. doi:10.1103/PhysRevLett.96.010401.

[52] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Cambridge University Press, 1989.

[53] Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden. Quantum cryptography. *Rev. Mod. Phys.*, 74:145–195, Mar 2002. doi:10.1103/RevModPhys.74.145.

[54] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. doi:10.1016/0022-0000(84)90070-9.

[55] 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.

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

[57] 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.

[58] Kurt Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37(1):349–360, 1930-12-01. doi:10.1007/BF01696781.

[59] Rudolf Haag and Daniel Kastler. An Algebraic Approach to Quantum Field Theory. *Journal of Mathematical Physics*, 5(7):848–861, 1964. doi:10.1063/1.1704187.

[60] Jan Hamhalter. *Quantum Measure Theory*. Springer Science & Business Media, 2003.

[61] Aram W Harrow, Benjamin Recht, and Isaac L Chuang. Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics*, 43(9):4445–4451, 2002. doi:10.1063/1.1495899.

[62] Teiko Heinosaari and Mário Ziman. *The Mathematical Language of Quantum Theory: From Uncertainty to Entanglement*. Cambridge University Press, 2011. doi:10.1017/CBO9781139031103.

[63] W. Heisenberg. Über quantentheoretische umdeutung kinematischer und mechanischer beziehungen. 33(1):879–893, 1925. doi:10.1007/BF01328377.

[64] D. Hilbert and W. Ackermann. *Grundzüge Der Theoretischen Logik*. Springer, 1928.

[65] 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. doi:10.1145/3290344.

[66] Simon Peyton Jones. *The Implementation of Functional Programming Languages*. Prentice Hall, 1987.

[67] Jan Jurka, Stefan Milius, and Henning Urbat. Algebraic reasoning over relational structures. *CoRR*, abs/2401.08445, 2024. `doi:10.48550/ARXIV.2401.08445`.

[68] Michael Keyl. Fundamentals of quantum information theory. *Physics Reports*, 369(5):431–548, 2002. `doi:10.1016/S0370-1573(02)00266-1`.

[69] H Jeff Kimble. The quantum internet. *Nature*, 453(7198):1023–1030, 2008. `doi:10.1038/nature07127`.

[70] Kreyszig. *Introductory Functional Analysis with Applications*. Wiley India Pvt. Limited, 2007-03.

[71] 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. `doi:10.1109/LICS.2011.38`.

[72] 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. `doi:10.4230/LIPICS.FSCD.2022.4`.

[73] J. Lambek and P. J. Scott. *Introduction to Higher-Order Categorical Logic*. Cambridge University Press, 1988.

[74] Joachim Lambek. From lambda-calculus to cartesian closed categories. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, pages 375–402, 1980.

[75] Daniel A Lidar. Lecture notes on the theory of open quantum systems. *arXiv preprint arXiv:1902.00967*, 2019. `doi:10.48550/arXiv.1902.00967`.

[76] Hans Maassen. Quantum probability and quantum information theory. *Lecture Notes in Physics*, 808:65–108, 07 2010. `doi:10.1007/978-3-642-11914-9_3`.

[77] Ian Mackie, Leopoldo Román, and Samson Abramsky. An internal language for autonomous categories. *Applied Categorical Structures*, 1(3):311–343, 1993. `doi:10.1007/BF00873993`.

[78] Christopher Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[79] 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. doi:10.1145/2933575.2934518.

[80] 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. doi:10.1109/LICS.2017.8005102.

[81] Matteo Mio, Ralph Sarkis, and Valeria Vignudelli. Universal quantitative algebra for fuzzy relations and generalised metric spaces. *Log. Methods Comput. Sci.*, 20(4), 2024. doi:10.46298/LMCS-20(4:19)2024.

[82] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[83] 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. Association for Computing Machinery, 2019. doi:10.1145/3297858.3304075.

[84] J von Neumann. Wahrscheinlichkeitstheoretischer aufbau der quantenmechanik. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1927:245–272, 1927.

[85] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010. doi:10.1017/CBO9780511976667.

[86] 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. doi:10.1145/2535838.2535879.

[87] 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. `doi:10.1103/PhysRevLett.80.3891`.

[88] Vern Paulsen. *Completely Bounded Maps and Operator Algebras*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2003. `doi:10.1017/CBO9780511546631`.

[89] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, 2014. `arXiv:mn2jBQAAQBAJ`.

[90] Gert Kjaergård Pedersen. *C*-Algebras and Their Automorphism Groups*. Academic Press, 1979.

[91] Simon Perdrix. Quantum entanglement analysis based on abstract interpretation. In *International Static Analysis Symposium*, pages 270–282. Springer, 2008. `doi:10.1007/978-3-540-69166-2_18`.

[92] Gilles Pisier. *Tensor Products of C*-Algebras and Operator Spaces: The Connes–Kirchberg Problem*. London Mathematical Society Student Texts. Cambridge University Press, 2020. `doi:10.1017/9781108782081`.

[93] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018. `doi:10.22331/q-2018-08-06-79`.

[94] Walter Rudin. *Functional Analysis*. McGraw-Hill, 1991.

[95] Raymond A. Ryan. *Introduction to Tensor Products of Banach Spaces*. Springer Science & Business Media, 2013.

[96] Shôichirô Sakai. *C*-Algebras and W*-Algebras*, volume 60 of *Classics in Mathematics*. Springer, 1998. `doi:10.1007/978-3-642-61993-9`.

[97] Bruna Salgado and Renato Neves. On the additive structure of quantalic λ-calculus. BMQL 2025, 2025. URL: `https://bmql25.cs.aau.dk/wp-content/uploads/2025/08/Salgado-Neves-bmql25.pdf`.

[98] 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. `doi:10.1103/PhysRevA.78.022322`.

[99] Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004. `doi:10.1017/S0960129504004256`.

[100] Peter Selinger. Lecture notes on the lambda calculus, 2013. `arXiv:0804.3434`.

[101] Peter Selinger and Benoît Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 16(3):527–552, 2006. `doi:10.1007/11417170_26`.

[102] 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. `doi:10.1007/978-3-540-78499-9_7`.

[103] 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. `doi:10.1016/j.entcs.2008.04.022`.

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

[105] 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. `doi:10.1145/3548679`.

[106] Yu Shi and Edo Waks. Error metric for non-trace-preserving quantum operations. *Phys. Rev. A*, 108:032609, Sep 2023. `doi:10.1103/PhysRevA.108.032609`.

[107] 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. `doi:10.1109/SFCS.1994.365700`.

[108] Michael Shulman. A practical type theory for symmetric monoidal categories. *Theory and Applications of Categories*, 37(5):863–907, 2021. `doi:10.48550/arXiv.1911.00818`.

[109] Andrew M Steane. Error correcting codes in quantum theory. *Physical Review Letters*, 77(5):793, 1996. `doi:10.1103/PhysRevLett.77.793`.

[110] Isar Stubbe. An introduction to quantaloid-enriched categories. 256:95–116, 2014. `doi:10.1016/j.fss.2013.08.009`.

[111] Masamichi Takesaki, editor. *Theory of Operator Algebras I*. Springer, 1979. `doi:10.1007/978-1-4612-6188-9`.

[112] 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. `doi:10.1145/3453483.3454029`.

[113] Sebastian Thrun et al. Robotic mapping: A survey. 2002.

[114] 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. `doi:10.1103/PhysRevA.57.3229`.

[115] 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. `doi:10.1007/978-3-319-23461-8_36`.

[116] Salvador Elías Venegas-Andraca. Quantum walks: A comprehensive review. *Quantum Information Processing*, 11(5):1015–1106, 2012. `doi:10.1007/s11128-012-0432-5`.

[117] Joel J Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94(5):052325, 2016. `doi:10.1103/PhysRevA.94.052325`.

[118] Jing Wang, Li Jiang, Han Zhang, Hanzhuang Zhang, and Liquan Zhang. Fidelity of structured amplitude-damping channels. 83(4):045008, 2011. `doi:10.1088/0031-8949/83/04/045008`.

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

[120] Abraham A. Westerbaan. The Category of Von Neumann Algebras, 2019. `arXiv:1804.02203`, `doi:10.48550/arXiv.1804.02203`.

[121] Glynn Winskel. *The Formal Semantics of Programming Languages: An Introduction*. MIT Press, 1993.

[122] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982. `doi:10.1038/299802a0`.

[123] Noson S. Yanofsky. Monoidal Category Theory: Unifying Concepts in Mathematics, Physics, and Computing (Lecture Slides), 2024.

[124] 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. `doi:10.1126/science.aan3211`.

[125] 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. `doi:10.1016/B978-044450871-3/50152-2`.

[126] Elias Zakon. *Mathematical Analysis I*. The Saylor Foundation, 2011.

[127] Jianjun Zhao. Quantum software engineering: Landscapes and horizons. *arXiv preprint*, 2020. `doi:10.48550/arXiv.2007.07047`.

[128] Sisi Zhou. Limits of noisy quantum metrology with restricted quantum controls. *Phys. Rev. Lett.*, 133:170801, Oct 2024. `doi:10.1103/PhysRevLett.133.170801`.

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,

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 [43]. 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. [123]

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 $\mathsf{C}(A, B)$, $\mathrm{Hom}_\mathsf{C}(A, B)$, 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^{op}, is defined as follows:

- The objects of C^{op} are the same as those of C.

- For any pair of objects A, B, the hom-set in C^{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^{op} corresponds to a morphism $f : B \to A$ in C.

- Composition in C^{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^{op}, corresponding to morphisms

  $$C \xrightarrow{\ g\ } B \xrightarrow{\ f\ } A$$

  in C, then the composition in C^{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, Coproducts, and Other Properties of Objects and Arrows

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 $A \in \mathsf{C}$, 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 $A \in \mathsf{C}$, 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. An arrow $f : A \to A$ in a category C is *idempotent* if $f \circ f = f$. An idempotent is said to *split* if there is an object B and functions $g : A \to B$ and $h : B \to A$ for which $h \circ g = f$ and $g \circ h = \mathrm{id}_B$. In this case, the pair (g, h) is called a *splitting* of the idempotent f.

Example A.2.19. In Set, consider the function $f : \mathbb{R} \to \mathbb{R}$ defined by

$$f(x) = \begin{cases} x, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0. \end{cases}$$

Then $f \circ f = f$, so f is idempotent.
Define $B = [0, \infty)$, and let

$$g : \mathbb{R} \to B, \quad g(x) = \max(x, 0), \quad \text{and} \quad h : B \to \mathbb{R}, \quad h(x) = x.$$

Then $h \circ g = f$ and $g \circ h = \mathrm{id}_B$, so f splits.

Definition A.2.20. 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.
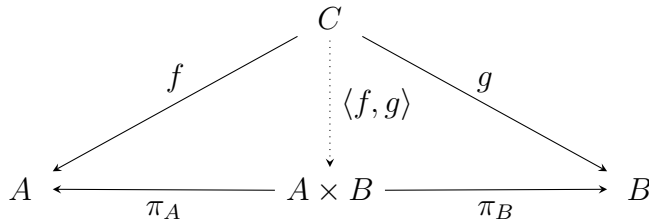
Definition A.2.21. Let $A \times B$ be a product of objects A and B, and let $A' \times B'$ be a product of objects A' and B' in a category C. Suppose we are given morphisms $f : A \to A'$ and $g : B \to B'$. Then there exists a unique morphism

$$f \times g : a \times b \to a' \times b'$$

such that the following diagram commutes.
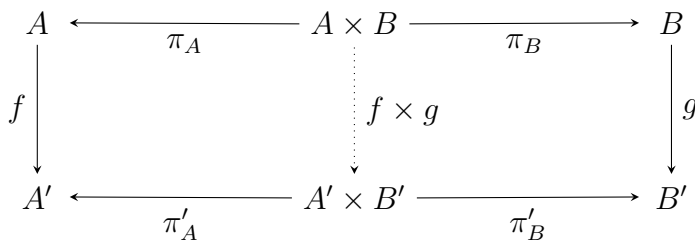
This induced morphism $f \times g$ is called the *product of the morphisms* f and g, and it is given explicitly by

$$f \times g = \langle f \circ \pi_A, \, g \circ \pi_B \rangle.$$

Theorem A.2.22. *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 $\pi_A \circ \langle f, g \rangle = f$ and $\pi_B \circ \langle f, g \rangle = g$. 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.

$\square$
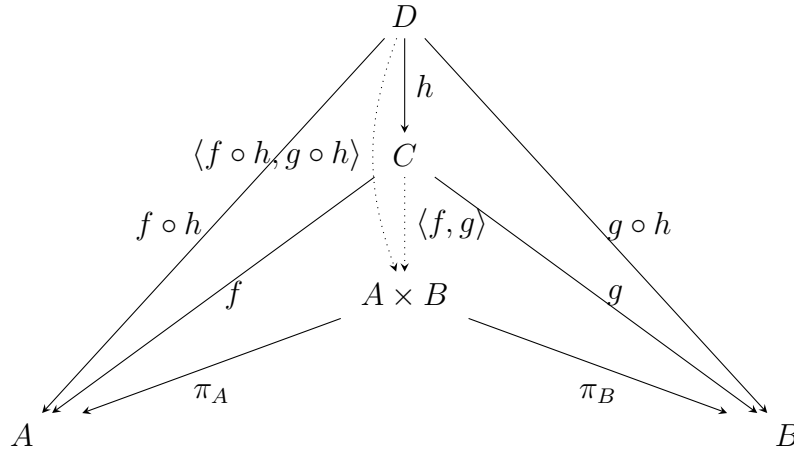
Example A.2.23. In the category Set, the product of two sets A and B is given by their Cartesian product, denoted as

$$A \times B = \{(a, b) \mid a \in A, \ b \in B\}.$$

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.2.24. 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.2.25. 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^{op}. More explicitly,

**Definition A.2.26.** 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 $A + B$ in C with morphisms $\mathrm{inl} : A \to A + B$ and $\mathrm{inr} : B \to A + B$ 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 $[f, g] : A + B \to C$ known as *co-pairing* that makes the diagram below commute.
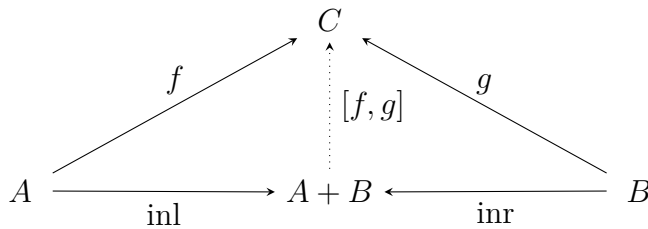
Definition A.2.27. Let $A + B$ be a coproduct of objects A and B, and let $A' + B'$ be a coproduct of objects A' and A' in a category C. Suppose we are given morphisms $f : A \to A'$ and $g : B \to B'$. Then there exists a unique morphism

$$f + g : A + B \to A' + B'$$

such that the following diagram commutes.
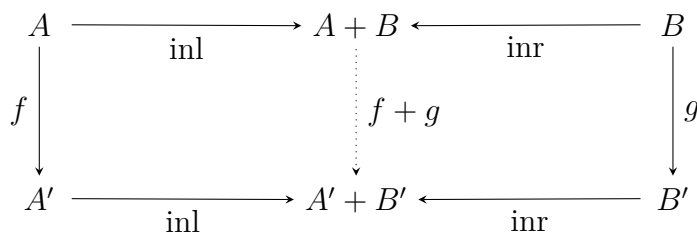
This induced morphism $f + g$ is called the *coproduct of the morphisms* f and g, and it is given explicitly by

$$f + g = [\mathrm{inl} \circ f, \mathrm{inr} \circ g].$$

Theorem A.2.28. *Let $A + B$ 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. $\qquad\square$

Proposition A.2.29. *[6, 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*

$$\varphi \cdot \mathrm{inl} = \mathrm{inl}' \quad \text{and} \quad \varphi \cdot \mathrm{inr} = \mathrm{inr}'.$$

Example A.2.30. In the category Set, the coproduct $A + B$ of two sets is their disjoint union, which can be constructed as

$$A + B = \{(a, 1) \mid a \in A\} \cup \{(b, 2) \mid b \in B\}.$$

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 $[f, g] : A + B \to C$ 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.2.31. Let $(P, \leq)$ be a partial order and P be its associated category. Consider a coproduct of elements $p + q \in P$. Then, by definition, there must exist injections satisfying

$$p \leq p + q \quad \text{and} \quad q \leq p + q.$$

Furthermore, for any element $z \in P$, if

$$p \leq z \quad \text{and} \quad q \leq z,$$

then it follows that

$$p + q \leq z.$$

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.2.32. In CVect the coproduct coincides with the product. In such cases, this structure is called a *biproduct,* and denoted by $\oplus$. 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 $\pi_i \cdot u = f_i$ 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.2.33. 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.2.34. 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.2.35. Let C be a category. Then there exists an *identity functor* $\mathrm{id}_C : C \to C$, which is defined on objects by $\mathrm{id}_C(A) = A$ for every object A in C, and analogously on morphisms, that is, $\mathrm{id}_C(f) = f$ for every morphism f in C.

Example A.2.36. 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.2.37. 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.2.38. 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.2.39. 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, $a \,\square\, b$. When we use this notation, the condition

$$F(f' \circ f, \, g' \circ g) = F(f', g') \circ F(f, g)$$

becomes

$$(f' \circ f) \,\square\, (g' \circ g) = (f' \,\square\, g') \circ (f \,\square\, g).$$

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.2.40. 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.

$$
\begin{array}{ccc}
FA & \xrightarrow{\eta_A} & GA \\
\downarrow{\scriptstyle Ff} & & \downarrow{\scriptstyle Gf} \\
FA' & \xrightarrow{\eta_{A'}} & GA'
\end{array}
$$

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

$$
\mathsf{C} \underset{G}{\overset{F}{\rightrightarrows}} \mathsf{D} \quad \Downarrow \eta
$$

Example A.2.41. 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.2.42. 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.2.43. 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.2.44. 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.2.45. 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.2.46. Given categories C and D, a pair of functors $L : \mathsf{C} \to \mathsf{D}$ and $R : \mathsf{D} \to \mathsf{C}$ 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.2.47. 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:

$$L(R) \le N \text{ if and only if } R \le \mathrm{inc}(N) = N.$$

Take $R = 7.27$ as an example. Then the inequality $R \le N$ holds precisely when N is an integer greater than or equal to 7.27. That is:

$$7.27 \not\le 5, \quad 7.27 \not\le 6, \quad 7.27 \not\le 7, \quad 7.27 \le 8, \quad 7.27 \le 9, \quad \dots$$

By the condition above, we must then have:

$$L(7.27) \not\le 5, \quad L(7.27) \not\le 6, \quad L(7.27) \not\le 7, \quad L(7.27) \le 8, \quad L(7.27) \le 9, \quad \dots$$

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 $r \le \lceil r \rceil$ 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.2.48. Let $F : \mathsf{C} \to \mathsf{D}$ and $G : \mathsf{D} \to \mathsf{E}$ 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.2.49. *[123, Section 4.6] Left adjoints preserve coproducts.*

A.2.7 Monoidal categories

Definition A.2.50. A **monoid** is a triple $(M, \cdot, u)$, where M is a set equipped with a binary operation $\cdot : M \times M \to M$ and a distinguished element $u \in M$ called the *unit*, satisfying the following axioms for all $x, y, z \in M$:

$$\text{(Associativity)} \qquad x \cdot (y \cdot z) = (x \cdot y) \cdot z,$$
$$\text{(Unit laws)} \qquad u \cdot x = x = x \cdot u.$$

Monoidal categories are named so because they are categories equipped with an additional structure that resembles the structure of monoids.

Definition A.2.51. 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.

Definition A.2.52. 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.

$$A \otimes (B \otimes C) \xrightarrow{\alpha} (A \otimes B) \otimes C \xrightarrow{\text{sw}} C \otimes (A \otimes B)$$

$$\downarrow \text{id} \otimes \text{sw} \qquad\qquad\qquad\qquad\qquad \downarrow \alpha$$

$$A \otimes (C \otimes B) \xrightarrow{\alpha} (A \otimes C) \otimes B \xrightarrow{\text{sw} \otimes \text{id}} (C \otimes A) \otimes B$$

Definition A.2.53. 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.2.54. 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,

$$[\text{inl} \otimes \text{id}, \text{inr} \otimes \text{id}] : B \otimes A + C \otimes A \to (B + C) \otimes A$$

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.2.55. 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.2.56 (*Coherence Theorem for Symmetric Monoidal Categories*). *[123, 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.*

A.3 Banach spaces

Definition A.3.1. 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 A.3.2. A vector space together with a norm is called a *normed vector space*.

Every normed space may be regarded as a metric space (Definition 2.3.8), in which the distance, $d(v, w)$, between vectors v and w is $\|v - w\|$.

Definition A.3.3. Let V and W be normed vector spaces, and let $T : V \to W$ be a linear operator. Then $\|T\|_{\text{op}}$ is defined as the nonnegative extended real number

$$\|T\|_{\text{op}} = \sup \left\{ \|T(v)\|_W \mid v \in V, \|v\|_V = 1 \right\}.$$

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*. The function $\|\cdot\|_{\mathsf{op}}$, when restricted to the space of bounded linear operators, is called the *operator norm*.

Lemma A.3.4. *[2, 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 $\|T(v)\| \leq \|T\|_{op} \cdot \|v\|$.*

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

Definition A.3.5. 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 A.3.6. (*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 A.3.7. 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.

A.4 Topology

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 $x \in [0, 1]$ [2]. 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 A.4.1. 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 A.4.2. 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 A.4.3. 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 A.4.4. 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 A.4.5. 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,

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

More generally, continuity may be defined as follows:

Definition A.4.6. 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 A.4.7. 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 A.4.8. Let S_1 and S_2 be two topological spaces, and let s_1 be a point in S_1. A map $f \colon S_1 \to S_2$ 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 A.4.9. *[2, 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 A.4.10. *[2, Theorem 2.28] Let $f : S_1 \to S_2$ 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 A.4.11. 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 A.4.12. 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 A.4.13. *[94, Theorem 4.3] Let V be a normed vector space. For each $v^* \in V^*$, define its norm by*

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

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

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

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

Definition A.4.14. 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 $v \in V$.