



THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e. g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

- This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
- A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
- The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
- When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Unbounded loops in quantum programs: categories and weak while loops

Pablo Andrés-Martínez



Doctor of Philosophy
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2022

Abstract

Control flow of quantum programs is often divided into two different classes: classical and quantum. Quantum programs with classical control flow have their conditional branching determined by the classical outcome of measurements, and these collapse quantum data. Conversely, quantum control flow is coherent, *i.e.* it does not perturb quantum data; quantum walk-based algorithms are practical examples where coherent quantum feedback plays a major role. This dissertation has two main contributions: (i) a categorical study of coherent quantum iteration and (ii) the introduction of weak while loops.

(i) The objective is to endow categories of quantum processes with a traced monoidal structure capable of modelling iterative quantum loops. To this end, the trace of a morphism is calculated via the execution formula, which adds up the contribution of all possible paths of the control flow. Haghverdi's unique decomposition categories are generalised to admit additive inverses and equipped with convergence criteria using basic topology. In this setting, it is possible to prove the validity of the execution formula as a categorical trace on certain categories of quantum processes. Among these there are categories of quantum processes over finite dimensional Hilbert spaces (as previously shown by Bartha), but also certain categories of quantum processes over infinite dimensional Hilbert spaces, such as a category of time-shift invariant quantum processes over discrete time.

(ii) A weak while loop is a classical control flow primitive that offers a trade-off between the collapse caused on each iteration and the amount of information gained. The trade-off may be adjusted by tuning a parameter and, in certain situations, it is possible to set its value so that we may control the algorithm without sacrificing its quantum speed-up. As an example, it is shown that Grover's search problem can be implemented using a weak while loop, maintaining the same time complexity as the standard Grover's algorithm (as previously shown by Mizel). In a more general setting, sufficient conditions are provided that let us determine, with arbitrarily high probability, a worst-case estimate of the number of iterations the loop will run for.

Acknowledgements

These four years of PhD studies have been humbling; I am ultimately content with the final product — both the experience and the result — but it did feel grim at times. Some exceptional people helped me push through without becoming a bitter person, and this is what I am most grateful for. To those individuals, I hope you can recognise the sincerity in the words that I dedicate to you below.

To Chris, you have given me the freedom to explore whatever I found interesting and I always had the certainty you would be willing and interested to talk about it. I am especially grateful for your friendliness, genuine curiosity and the unconditional support you have given me, both in my PhD studies and in discussions of career paths.

To Robin, collaborating with you was a pleasure. I am inspired by your researcher style and I am grateful for the patience you (and Chris) have had with me. Your joyful spirit towards research lifted me up when I was struggling with burn out. That gave me strength to take my time to polish up my results and write a thesis I can be proud of.

To my parents, thank you for understanding and putting up with my tendency to drift away when I am not feeling at my best. Thank you for bringing me up in a way that I felt free (and responsible) of my own choices and in an environment that gave me so many opportunities. It goes without saying I would not be here without you. On that note, I must also thank a few teachers and university professors that, along my parents, made me thirst for knowledge; thank you Enrique Meléndez, Juan María Jurado, Francisco Vico and Inmaculada P. de Guzmán.

To the *Patos Macizones*, each and every one of you. It is rare to have such a long lasting friendship that is still standing strong, it makes me feel special to be part of such an exceptional bunch of people. Spending quality time with you has always felt so genuine, refreshing and liberating — League games do not count as quality time. A special shout out to my brother, Miguel; I have avoided so many obstacles in life by following your trail, you are a great older brother and friend.

To Emil and Natalia, the original rad flat was a great experience and having you around made me feel instantly at home in Edinburgh. Your support during periods of self-doubt was invaluable.

To Ali and Tim, I am really grateful to you for helping me get out and socialise every now and then. The long walks and chats with Ali by the end of the PhD helped a lot in keeping everything into perspective and maintaining a healthy attitude. I have grown as a person by simply being around you both and I have regained confidence in

who I am in great deal thanks to you.

To Hono, by far you are the person I have had the largest number of deep conversations with. Talking with you about almost anything tends to be a wonderful exercise in creative thinking. I derive a great deal of joy from our conversations and they remind me that the thirst for knowledge also asks for things beyond academic matters.

To Vicky, you are a wonderful partner. I have made some selfish choices to get where I am and you have welcomed each and every one of them because you genuinely want me to grow and enjoy my life at its fullest. As cheesy as it is, your support keeps me afloat and reminds me that, ultimately, whoever I end up being or doing will be alright. There is so much to thank you for that literally everything I have said in the previous four paragraphs applies almost verbatim to you too.

I am truly grateful.

Non-technical Summary

If you have ever followed a recipe from a cookbook, it is likely you have encountered an unbounded loop: these have the form “bake the soufflé until it turns golden”. It is a *loop* since you keep doing the same action — baking — for an extended duration of time and it is *unbounded* since the actual amount of time you will bake for has not been specified; it may take longer on less powerful ovens. Writing a program is very similar to telling a computer how to follow a recipe: you specify the steps to solve a problem so that the computer “bakes” the solution for you. Quantum computers are no different, you still need to give them a recipe; however, quantum computer scientists have generally been shy in their use of unbounded loops in quantum recipes. The main motivation behind this thesis is to argue that unbounded loops in quantum computer science may be more interesting than they are usually credited for.

When following the instruction “bake the soufflé until it turns golden” you are repeatedly peeking at the soufflé to decide whether or not it is time to turn off the oven; a classical computer would do the same. However, quantum computers deal with quantum data which is very delicate in the sense that the simple act of measuring it alters it: as if peeking at your quantum soufflé would spoil it. Then, how can you possibly know when the soufflé is ready? It turns out that how “badly” the soufflé is spoiled depends on how “intensively” you peeked at it. One of the contributions of this thesis is the discussion of a general procedure that let us play with this trade-off and come out on top: we can keep an eye on the soufflé so we detect when it is ready without spoiling it badly.

The soufflé in the previous metaphor was meant to represent quantum data, but the nature of our decision making was still classical: we either turned off the oven or let it bake for a little longer. Quantum physics allows us to do something crazier: we may *entangle* the state of the soufflé with the state of the oven. If we engineer the right entanglement, the scenario where the oven is still *on* while the soufflé is burning will simply be unfeasible. Then, we no longer need to control the oven ourselves, avoiding the risk of spoiling the soufflé by peeking at it. Previous works in the literature have attempted to formalise these ideas in mathematical terms that would help us better understand these “fully quantum” unbounded loops. In this thesis I make contributions to this line of research, discussing a mathematical framework where these “fully quantum” unbounded loops are proven to be well-defined and consistent with the theory.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Pablo Andrés-Martínez)

Table of Contents

1	Introduction	1
1.1	Brief introduction to quantum computing	1
1.1.1	Control flow of quantum programs	4
1.1.2	Quantum walks: an example of coherent quantum feedback	7
1.2	Contributions and structure of the thesis	9
1.2.1	Categorical study of iterative quantum loops	10
1.2.2	Weakly measured while loops	11
1.3	Publications and previous work	11
1.4	Preliminaries on category theory (<i>Preamble</i>)	13
1.4.1	Monoidal categories	18
1.4.2	Enriched category theory	21
1.4.3	Adjunctions	23
1.5	Categories in quantum computer science (<i>Preamble</i>)	27
1.5.1	Beyond coherent quantum operations	29
2	Categories for infinitary addition	33
2.1	Σ -monoids	34
2.1.1	Important full subcategories of Σ_w	40
2.1.2	Σ_s is a reflective subcategory of Σ_w	43
2.1.3	Σ_{ft} is a reflective subcategory of Σ_w	46
2.1.4	Σ_g is a reflective subcategory of Σ_{ft}	48
2.1.5	Tensor product of Σ -monoids	50
2.2	Topological monoids	63
2.2.1	Brief introduction to general topology (<i>Preamble</i>)	63
2.2.2	Nets and limit points (<i>Preamble</i>)	67
2.2.3	Hausdorff commutative monoids	71
2.2.4	The left adjoint to $\mathbf{HausCMon} \rightarrow \Sigma_{ft}$	80

2.3	Related work	84
3	Categorical study of quantum iteration	87
3.1	Traced monoidal categories (<i>Preamble</i>)	87
3.1.1	The execution formula	91
3.1.2	The kernel-image trace	93
3.1.3	The canonical trace on compact closed categories	98
3.2	Unique decomposition categories	100
3.2.1	Classical iterative loops	109
3.2.2	Towards an execution formula with additive inverses	116
3.2.3	Σ_g -UDCs and their kernel-image trace	117
3.2.4	A limit condition	120
3.3	Quantum iterative loops and the execution formula	128
3.3.1	Properties of contractions in Hilb (<i>Preamble</i>)	129
3.3.2	The execution formula in FdContraction	134
3.3.3	Comparing the traces in FdContraction and CPTR	144
3.3.4	Open questions in Contraction	147
3.4	Coherent quantum processes over time	148
3.4.1	The discrete-time Fourier transform (<i>Preamble</i>)	152
3.4.2	Linear shift invariant maps	155
3.4.3	The execution formula in LSI _≤	161
3.5	Related work	164
4	Weakly measured while loops	167
4.1	Weak measurements (<i>Preamble</i>)	168
4.2	The κ -while loop	169
4.2.1	Motivation	170
4.2.2	Definition and properties	171
4.3	An example: Grover’s algorithm	173
4.3.1	Standard Grover’s algorithm (<i>Preamble</i>)	175
4.3.2	While loop approach	175
4.4	Discussion	181
4.4.1	Related work	182
4.4.2	Applications beyond quantum search	183

5	Final remarks	185
5.1	A toy language with quantum control flow	185
5.2	The κ -while loop in existing programming languages	188
5.3	Conclusion	190
	Bibliography	193

Chapter 1

Introduction

This thesis studies quantum iteration using multiple technical results on traced monoidal categories, Hausdorff topology, the Fourier transform and weak quantum measurements. For the sake of completeness, a brief introduction to each of these fields of study is provided where relevant throughout the text. Thus, the thesis is self-contained in the sense that no prior knowledge is required from the reader. This stylistic choice results in a rather long text with multiple sections on introductory material. These sections have been labelled as (*Preamble*) in their title (also appearing in the index) so that readers experienced in the subject may skip them without risk of missing out on novel contributions. A glossary has been provided at the end of the text, linking technical nomenclature to the page it is defined in.

This first chapter serves as the introduction to the thesis. Section 1.1 provides a brief introduction to quantum computing and, in particular, a discussion of the state of the art in the field of control flow of quantum programs — the field of study that this thesis contributes to. Sections 1.2 and 1.3 summarise the contributions of this thesis along with its structure and enumerate previous works we build upon. Sections 1.4 and 1.5 provide a brief introduction to category theory and examples of categories relevant to quantum computing.

1.1 Brief introduction to quantum computing

The information held in the memory of a computer at any given time is known as its *state*. A programmer provides a computer with instructions on how it must manipulate its state so that, at the end of the process, the resulting state encodes the solution to the problem the programmer is interested in. Thus, there are three essential aspects that

need to be captured by any mathematical framework used to discuss computation:

- the collection of possible states the computer can be in,
- the collection of transformations that we may apply to such states and
- how information may be retrieved from the computer's state.

The collection of possible states of a *quantum computer* is described as a Hilbert space H . Information may be retrieved from a quantum state by applying a *measurement*; however, unlike in classical computing, measuring perturbs the quantum state being observed. To apply a measurement, we must first choose a decomposition of H into orthogonal subspaces:

$$H \cong \bigoplus_{i \in I} H_i$$

with projections $\pi_i: H \rightarrow H_i$. When the measurement is applied to a state $v \in H$ it becomes one of the states in the collection $\{\pi_i(v)\}_{i \in I}$, up to normalisation; with the probability of the i -th outcome determined by $\|\pi_i(v)\|^2$. Consequently, vectors $v \in H$ describing quantum states are required to be unit vectors $\|v\| = 1$, so that the probabilities of all possible measurement outcomes add up to one. The rest of the vectors in H are understood as ‘unnormalised’ states.

Notation 1.1.1. Given two vectors $v, u \in H$, their inner product is denoted $\langle u | v \rangle$; the widespread convention in quantum computing is to define the inner product to be linear in the second argument (and anti-linear in the first one). In the bra-ket notation, vectors are denoted as ‘kets’ $|x\rangle \in H$ where x may be any arbitrary label, e.g. $|0\rangle, |1\rangle, |\psi\rangle \in H$. For each vector $|\psi\rangle \in H$ we define its ‘bra’ to be the linear map $\langle\psi|: H \rightarrow \mathbb{C}$ defined as follows:

$$\langle\psi|(v) = \langle\psi|v\rangle.$$

For any two vectors $|\psi\rangle, |\varphi\rangle \in H$ we define the linear map $|\psi\rangle\langle\varphi|: H \rightarrow H$ to be:

$$|\psi\rangle\langle\varphi|(v) = \langle\varphi|v\rangle \cdot |\psi\rangle.$$

To manipulate their state, quantum computers may apply *coherent* operations on their state space H . The term ‘coherent’ means that no loss of quantum information occurs when applying the operation; formally, this is captured by the requirement that coherent operations are invertible. The coherent operations that may be applied on a quantum state space H are unitary linear maps $f: H \rightarrow H$, whose inverse is given by their adjoint. In some situations — for instance, after applying a measurement —

we may be certain that the computer's state is in a closed subspace $H_0 \subset H$; then, it is reasonable to consider the restriction of a unitary $f: H \rightarrow H$ to $f|_{H_0}: H_0 \rightarrow H$. Such a linear map $f|_{H_0}$ is no longer surjective and, consequently, it cannot be unitary; however, it still is an isometry (and, hence, it has a left inverse). The requirement that the computer's operations are isometries guarantees that $\|f(v)\| = \|v\|$ for any $v \in H$, so that unit vectors (*i.e.* normalised states) are mapped to unit vectors.

Recall that, for any $n \in \mathbb{N}$, all Hilbert spaces of dimension n are isomorphic to each other, the zero-dimensional space is just $\{0\}$ and the one dimensional space is the field $\mathbb{C}$. Thus, $\mathbb{C}^2$ can be seen as the smallest nontrivial Hilbert space and, as such, it plays a special role in quantum computing: that of representing the state space of a *qubit*, the smallest unit of quantum memory. It is customary to fix an orthonormal basis $|0\rangle, |1\rangle \in \mathbb{C}^2$ and refer to these as the 'classical states' of the qubit, whereas any linear combination:

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

is referred to as a *superposition* of $|0\rangle$ and $|1\rangle$. Given two quantum memories whose state spaces are H and H' , the space of states that the two together can store is the tensor product $H \otimes H'$. For instance, the state space provided by two qubits is $\mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^4$ whose classical states are now denoted by the orthonormal basis

$$\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$$

whose vectors are often denoted using the shorthand $|01\rangle := |0\rangle \otimes |1\rangle$. All unit vectors in $H \otimes H'$ represent physically realisable states, but recall that not all vectors in a tensor product are of the form $|\varphi\rangle \otimes |\phi\rangle$, *i.e.* the canonical bilinear map $p: H \times H' \rightarrow H \otimes H'$ is not surjective. States $|\psi\rangle \in H \otimes H'$ that are not in the image of p are known as *entangled* states; the term is a direct reference to the fact that $|\psi\rangle \notin \text{im}(p)$ implies that the state of the two quantum memories cannot be represented by describing the state of each one in isolation.

Instead of dealing with arbitrary Hilbert spaces H and arbitrary unitary operators $H \rightarrow H$, quantum computer scientists often work at a level of abstraction closer to the real-world quantum computer. The memory of a standard quantum computer is comprised of a finite number n of qubits that it can maintain and manipulate at any given time and, consequently, the state space of a real-world quantum computer is a finite-dimensional Hilbert space $\mathbb{C}^{2^n}$. Quantum computers support the direct application of only a selection of unitary maps often comprised of:

- a finite set of unitary maps acting on single qubits $\mathbb{C}^2 \rightarrow \mathbb{C}^2$ that, under composition, generates a group G_2 that is a dense subset of $SU(2)$ — in the sense that, for any $f \in SU(2)$ and any $\epsilon > 0$ there is an element $g \in G_2$ such that $\|f - g\|_{\text{op}} < \epsilon$ — and
- a finite set (often a singleton set) of unitary maps acting on two qubits $\mathbb{C}^4 \rightarrow \mathbb{C}^4$ that, along with G_2 and under composition and tensor product generates a dense subset of $SU(2^n)$, so that any arbitrary unitary operation on the state space of n qubits may be implemented up to arbitrary precision.

Moreover, quantum computers support measurement of qubits in their $\{|0\rangle, |1\rangle\}$ basis which, paired with the ability to approximate any unitary map, enables us to perform any arbitrary measurement.

1.1.1 Control flow of quantum programs

A program specifies a list of operations that a computer is instructed to perform. The program's *control flow* dictates the order in which such operations ought to be applied; the ordering is often made to depend on the computer's state during the program's execution. For instance, the control flow may branch according to a decision making process within the program — thus, applying a sequence of operations or another — or it may loop and repeat a segment of the program until a condition is met. Programming languages offer specialised instructions, known as control primitives, that let the programmer specify the control flow of the program; common examples are *if-then-else* statements, *while loops*, recursion, *etc.* When designing programming languages for quantum computers, two fundamentally different notions of control flow may be considered.

Classical control flow. All decision making is dictated by the outcome of measurements, *i.e.* classical information. This paradigm is simple to realise on physical devices — it may be realised by a quantum chip controlled by a classical computer. Its mathematical formalisation is well understood, since it shares much in common with formal methods for programming languages in classical computer science (see Section 3.2.1). Its main drawback is that, due to the use of measurements, quantum information is irreversibly lost whenever a control primitive needs to be evaluated.

Quantum control flow. The motivation behind this paradigm comes from the realisation that conditional control primitives may be performed *coherently*, *i.e.* in a reversible manner — and, consequently, without loss of quantum information. The canonical example is the construction of a quantumly controlled unitary $\Lambda U: \mathbb{C}^4 \rightarrow \mathbb{C}^4$ defined by linear extension of:

$$\Lambda U(|0\rangle \otimes |\psi\rangle) = |0\rangle \otimes |\psi\rangle \quad \Lambda U(|1\rangle \otimes |\psi\rangle) = |1\rangle \otimes U|\psi\rangle$$

where $U: \mathbb{C}^2 \rightarrow \mathbb{C}^2$ is an arbitrary unitary and $|\psi\rangle \in \mathbb{C}^2$ an arbitrary state. The unitary ΛU applies U on the second qubit if the first qubit is in state $|1\rangle$ and does nothing if it is in state $|0\rangle$; furthermore, when the first qubit is in a superposition $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, we obtain the following behaviour:

$$\Lambda U(|\phi\rangle \otimes |\psi\rangle) = \alpha(|0\rangle \otimes |\psi\rangle) + \beta(|1\rangle \otimes U|\psi\rangle)$$

so that both options — either applying U or not — coexist in a superposition. Under reasonable assumptions (*e.g.* P describes a unitary process), we may define a conditional control primitive in quantum programming languages:

***if* $q \in K$ *then do* P**

that applies the subprogram P on every state q in the closed subspace K , does nothing on the orthogonal complement of K and, on any other vector, acts accordingly to the linear extension — for instance, ΛU would be described by setting P to apply U on the second qubit and define K as the subspace of vectors of the form $|1\rangle \otimes -$. The next natural step is to consider quantumly controlled iterative loops; unfortunately, the requirement that the computation is reversible creates some major obstacles. Assume a programming language offers a control primitive

***do* P *while* $q \in K$**

whose behaviour is described by the following recursive definition:¹

***do* P
if $q \in K$ *then*
do P *while* $q \in K$**

so that subprogram P will be repeatedly applied on any linear component of the state that lands in K . There are three important issues with this iterative control primitive, which have been noted in multiple previous works.

¹The choice of defining a ‘do-while’ primitive instead of the more common ‘while-do’ is not accidental and will be made clear in Section 5.1 of the concluding chapter.

- *A formal issue.* As pointed out by Bădescu and Panangaden [8], it is not clear whether the map $H \rightarrow H$ induced by the iterative primitive would converge at all. Moreover, for the formalism to be considered physically sound, we would require that the solution is unitary and that the map giving the denotational semantics of the ‘do-while’ primitive be continuous in a topological sense — *i.e.* that small changes to P would not cause abrupt changes in the behaviour of the loop.
- *A physical issue.* Some linear components of the input state will have P applied to them more times than others. Evidently, applying subprogram P twice will take more time than applying it once, which means that different terms in the linear combination may not be ‘synchronised’ in time. However, vectors in the Hilbert space H are understood as the state of the quantum computer *at a particular time-step* and, hence, the physical intuition of this iterative loop is not properly captured in the mathematical formalism.
- *A functionality issue.* To amend the physical issue, Ying, Yu and Feng [56] propose to introduce an auxiliary qubit on each iteration, generating a chain of qubits of unbounded length that records the ‘history’ of each term in the superposition. However, as Linden and Popescu discuss in [40] this causes linear components with different histories to no longer interact with each other, as their distinct histories make them orthogonal, and unitary operations send orthogonal states to orthogonal states. This prevents any interference between different branches of the control flow, making it effectively classical.

The formal issue was partially resolved in [10]: Bartha shows that the category of unitary maps is traced with respect to the execution formula; we may interpret this as saying that, if the body of the do-while loop (subprogram P) describes a unitary $H \rightarrow H$, then the result of aggregating all possible control flow branches of the quantum loop yields a unitary. However, Bartha’s approach makes use of pseudoinverses, which complicates the study of whether the iterative primitive is continuous.² The physical issue may be resolved by adding a notion of time to the formalism, in the manner discussed in Section 3.4; the crucial distinction with the proposal from Ying *et al.* [56] is that their locally generated ‘history’ is replaced with a global notion of ‘time-line’. States at different time-steps may not interfere with each other, but the same time-step

²In this work we do not manage to prove that it is in fact continuous, but our framework is more amenable to conduct such a study, as discussed in Remark 3.3.17.

may be reached via multiple paths in the control flow, and such paths will interact with each other; thus, the functionality issue is prevented.

One of the main objectives of the thesis is to formalise this argument, laying out the groundwork for formal semantics of programming languages with quantum iteration. In particular, Chapters 2 and 3 will introduce a categorical framework that Section 5.1 will argue to be appropriate for this task. The discussion in Chapters 2 and 3 is highly abstract and technical, but the ideas are rooted in physical intuitions drawn from the computational model of quantum walks, which is briefly introduced in the following section.

1.1.2 Quantum walks: an example of coherent quantum feedback

Let $G = (V, E)$ describe a graph on a finite set of vertices V and a set of edges $E \subseteq V \times V$ such that if $(u, v) \in E$ then $(v, u) \in E$, i.e. the graph is undirected. For each $v \in V$ let E_v be the following set of edges:

$$E_v = \{(u, v) \in E \mid u \in V\},$$

let $\Delta: V \rightarrow \mathbb{N}$ be the function that yields the degree of a vertex, $\Delta(v) = |E_v|$, and let H_v be the finite-dimensional Hilbert space spanned by taking E_v as an orthonormal basis. Fix a collection of unitary maps $\{f_v: H_v \rightarrow H_v\}_{v \in V}$ to be the set of *coins* of the walk. A *quantum walk* is determined by the pair $(G, \{f_v\}_V)$ and it describes the evolution of a quantum state in the Hilbert space $H = \oplus_{v \in V} H_v$ as follows:

- let $C: H \rightarrow H$ be the *coin operator* defined as the direct sum

$$C = \bigoplus_{v \in V} f_v;$$

- let $S: H \rightarrow H$ be the *shift operator* defined as the linear extension of

$$S|u, v\rangle = |v, u\rangle$$

for each $(u, v) \in E$, with $|u, v\rangle$ being its corresponding unit vector in H ;

- for any state $|\psi\rangle \in H$, after $t \in \mathbb{N}$ time-steps the state becomes:

$$(SC)^t|\psi\rangle.$$

An illustrative physical interpretation of quantum walks is to view the graph G as the layout of an experiment, where each vertex determines a linear optics device (*e.g.* lenses, mirrors, beamsplitters, *etc.*) and edges describe the path a photon can take from one device to another. For each $(u, v) \in E$, the state $|u, v\rangle \in H$ is interpreted to indicate the presence of a photon located somewhere between the device at vertex u and the one at vertex v . A linear combination

$$|\psi\rangle = \sum_{(u,v) \in E} \alpha_{u,v} |u, v\rangle$$

describes *a single photon* in a superposition of locations. The action of the device at vertex v on an incoming photon is described by the unitary operator f_v and the coin operator C simply arranges all of the coins f_v into a single linear map. The shift operator S conveys the structure of the graph, ensuring that the output that f_v leaves at edge (u, v) is, on the next time-step, an input to f_u . Quantum walks are not limited to describe experiments on linear optics and, in general, may be understood to describe arbitrary quantum processes, similarly to how Markov chains and random walks describe stochastic processes [2]. In the abstract case, instead of a photon we use the term *walker* to refer to the entity traversing the graph.

The control flow of any program may be described as a graph, where vertices correspond to the different instructions in the program and edges indicate the possible control flow paths between them. The state of the control flow during execution may be described as a quantum walk on such a graph, where the position of the walker indicates the branch of the control flow (or superposition of branches) that is currently being computed. Crucially, the graph describing a quantum walk may have cycles, and these make it so that the walker (or a linear term within its superposition) may visit the same vertex any arbitrary number of times, thus representing a loop in the control flow.

Notice that both of the operators C and S describing the quantum walk's evolution are unitary and, therefore, reversible. Consequently, quantum walks are a physically realisable framework that supports coherent feedback of quantum processes and, in particular, it can be understood to model quantum (coherent) control flow of programs. However, there are two caveats: first, the graph describing the control flow of a program is directed while the graph of a quantum walk is undirected; fortunately, the requirement that G is undirected is only a matter of convention (which simplifies the quantum walk's definition) and directed quantum walks may be defined, as long as we guarantee that the in-degree and out-degree of each vertex coincides, so that its coin may be a unitary. The second caveat is that programs are supposed to have inputs and

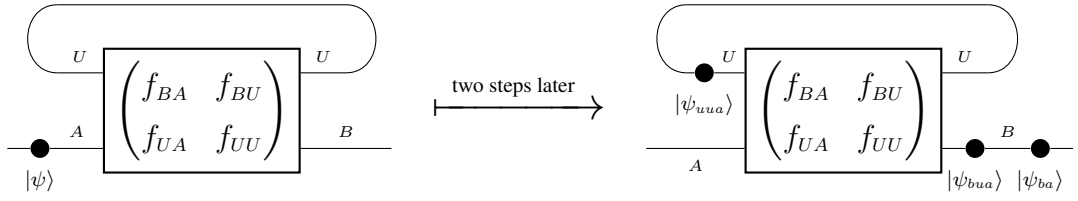


Figure 1.1: A state $|\psi\rangle$ is given as input to a quantum iterative process; two time-steps later, the state is in a superposition $|\psi_{ba}\rangle + |\psi_{bua}\rangle + |\psi_{uua}\rangle$ where $|\psi_{ba}\rangle = f_{BA}|\psi\rangle$, $|\psi_{bua}\rangle = f_{BU}f_{UA}|\psi\rangle$ and $|\psi_{uua}\rangle = f_{UU}f_{UA}|\psi\rangle$. The term $|\psi_{ba}\rangle$ ‘leaves’ the system a step earlier than $|\psi_{bua}\rangle$.

outputs so that they may be composed, but quantum walks are defined on closed graphs and there is no immediate notion of composition for them. A way around this is to define open quantum walks: quantum walks on graphs with special vertices marked as either ‘input’ or ‘output’ so that the walker may enter or leave the graph through them. But then, some linear term of the walker’s state may leave the graph at time-step t , while other terms may not leave until a later time-step $t' > t$ (see Figure 1.1). Thus, to discuss the input-output behaviour of open quantum walks we must extend the Hilbert space H to describe states over time; a way to do so for arbitrary quantum states and processes is described in Section 3.4.

Open quantum walks may also be defined over continuous time [52] and, in order to describe their composition with ease, their input-output behaviour is described in the frequency domain; this insight motivates much of Section 3.4. Both continuous-time and discrete-time open quantum walks have been shown to be universal models of quantum computation [16, 42], in the sense that any unitary operation may be approximated to arbitrary precision by combining a finite variety of resources. Multiple quantum algorithms have been defined in terms of quantum walks (see [3] and the survey from [47]), which may be seen as evidence of the power of coherent feedback in quantum computing. In practice, a quantum algorithm described via a quantum walk is implemented in a quantum computer using simulation techniques such as the ones described in [19].

1.2 Contributions and structure of the thesis

This thesis has two main contributions, both relevant in quantum computing; the first one in the field of quantum control flow and the second one in the field of classical

control flow of quantum programs.

- The first contribution is on the formal study of iterative quantum loops by means of categorical traces on categories of quantum processes. In particular, the thesis focuses on the execution formula, which aggregates all of the possible paths of the quantum control flow. It is shown that both the category of finite-dimensional Hilbert spaces and contractions and the category of linear shift invariant contractions (as introduced in Section 3.4) are totally traced with respect to the execution formula. To do so, Haghverdi’s unique decomposition categories [28] are generalised and equipped with convergence criteria based on basic topology. These contributions are presented in Chapters 2 and 3.
- The second contribution is the proposal of a classical control flow primitive — the weakly measured while loop — that offers a trade-off between decoherence due to measurement and information gained. It is argued that this primitive offers a restricted ability to monitor a quantum state as it evolves; sufficient conditions for such a monitoring to be compatible with quantum speed-up are discussed. These contributions are presented in Chapter 4, which reproduces the contents of the published work [5].

More details on the motivation and structure of these chapters is given in the following subsections. Chapter 5 concludes the thesis and proposes further work in both of these lines of research.

1.2.1 Categorical study of iterative quantum loops

Haghverdi’s unique decomposition categories [28] have been used to model iterative loops in classical computer science. These categories rely on a notion of addition of morphisms that does not admit additive inverses. Unfortunately, additive inverses are a fundamental aspect of categories of quantum processes, as they capture destructive interference, which is one of the key aspects of quantum theory. Chapter 2 proposes a generalisation of the additive structure on Haghverdi’s categories, so that additive inverses are allowed. A connection to topological groups is made, which will be essential to prove the main results in Chapter 3. Multiple categories capturing subtle differences in the notion of infinitary addition are presented, and their relationship is exhibited in terms of adjunctions.

Chapter 3 builds upon the results of the previous chapter to provide a version of Haghverdi’s unique decomposition categories equipped with convergence criteria using basic topology. In this setting, it is possible to prove the validity of the execution formula as a categorical trace on certain categories of quantum processes. Among these there are categories of quantum processes over finite-dimensional Hilbert spaces (as previously shown by Bartha [10] for the case of isometries and unitaries), but also certain categories of quantum processes over infinite-dimensional Hilbert spaces, such as the category of time-shift invariant quantum processes over discrete time (introduced in Section 3.4). The latter result lays the groundwork for the design of categorical semantics for quantum programming languages supporting (coherent) quantum iterative loops. What such a language and its semantics may look like is sketched in Section 5.1 of the concluding chapter; in essence, the discrete time line needs to be formalised in the semantics and managed as transparently as possible.

1.2.2 Weakly measured while loops

Weakly measured while loops are proposed and discussed in Chapter 4. A weakly measured while loop is a classical control flow primitive that offers a trade-off between the collapse of the quantum state caused on each iteration and the amount of information gained. The trade-off may be adjusted by tuning a parameter and, in certain situations, it is possible to set its value so that quantum speed-up is achieved. As an example, it is shown that Grover’s search problem can be implemented using a weak while loop, maintaining the same time complexity as the standard Grover’s algorithm. In a more general setting, sufficient conditions are provided that let us determine, with arbitrarily high probability, a worst-case estimate of the number of iterations a weakly measured while loop will run for, thus allowing us to study the worst-case time (and query) complexity of quantum algorithms that use this primitive.

1.3 Publications and previous work

In this section we enumerate the main works from the literature this thesis builds upon; these will be discussed in more detail when relevant throughout the thesis and, in particular, in the last section of each chapter. In this section we also point out the publications that have come out from this PhD thesis and discuss where novel results presented here may lead to new publications.

Categories for infinitary addition. This topic is explored in Chapter 2 and builds upon the notion of Σ -monoids proposed by Haghverdi [28] and results and proof strategies introduced by Hoshino [33] regarding the category of Σ -monoids. We also consider Hausdorff commutative monoids but only use concepts and results appearing in standard introductory texts on general topology. During the process of connecting Σ -monoids with Hausdorff commutative monoids we took inspiration from Higgs’ work on Σ -groups [32].

Categorical study of quantum iteration. This topic is explored in Chapter 3 and builds upon the unique decomposition categories of Haghverdi [28] and their refinement due to Hoshino [33]. We provide a new proof for a result shown by Bartha [10] — that **FdIsometry** (and, more generally, **FdContraction**) is totally traced using the execution formula — and, to do so, we use some lemmas from Bartha’s work along with the kernel-image trace introduced by Malherbe, Scott and Selinger [44]. The transition from **FdContraction** to the category of time-shift invariant quantum processes over discrete time $\mathbf{LSI}_{\leq}$ is achieved using well-known results on the Fourier transform that can be found in introductory texts on signal processing and engineering.

Our result that the execution formula in $\mathbf{LSI}_{\leq}$ is a valid categorical trace is novel. We believe this result may be used to provide semantics for programming languages with quantum control flow (see Section 5.1 for a toy language). In the coming months we will work on preparing these results in a format suitable for publication.

Weakly measured while loops. This topic is explored in Chapter 4 and builds upon the notion of weak measurement (see [14] for an introduction). Previous work by Mizel [46] proposed an algorithm that is fundamentally the same as the weakly measured Grover algorithm we present in Chapter 4. What distinguishes our work from Mizel’s is that we propose a programming primitive — the κ -while loop — and lay the foundations of weakly controlled quantum iteration for the purpose of quantum algorithms, presenting the case of Grover’s search as a proof of concept. The results discussed in Chapter 4 were accepted as a journal publication [5].

Universal properties of partial quantum maps. This topic is tangential to the thesis and, as such, it is not discussed in this text. However, it is worth mentioning that a collaboration with Kaarsgaard and Heunen took place during the final year of my PhD studies, yielding a manuscript [6] that was accepted at the International Workshop on

Quantum Physics and Logic (QPL 2022). In this project we succeeded in describing how the category of finite-dimensional C^* -algebras and completely positive trace non-increasing maps can be obtained after applying a sequence of constructions characterised by universal properties which start from the category of finite-dimensional Hilbert spaces and unitary maps $\mathbf{FdUnitary}$.

1.4 Preliminaries on category theory (*Preamble*)

This section is included for the sake of completeness and to introduce the notation used. The reader is not assumed to have prior knowledge of category theory; however, the section may be too dense in concepts for an uninitiated reader. There are multiple books providing a pedagogical introduction to category theory and, among them, the book by Heunen and Vicary [31] is a particularly good match for this thesis, considering both its focus on monoidal categories and the discussion of their relevance in quantum computer science. For the concepts of (co)limits and adjunctions, the book by Leinster [38] is recommended.

Composition (of processes, devices, programs, functions, *etc.*) is a fundamental operation often taken for granted. Given two functions $f: A \rightarrow B$ and $g: B \rightarrow C$, their composite $g \circ f$ is obtained by applying f to the input, then g to the result:

$$(g \circ f)(a) = g(f(a)). \quad (1.1)$$

If we see $- \circ -$ as a binary operation over the set of functions, we find that it is not a total function: not all pairs of functions can be composed together. At its most basic level, a category is an axiomatisation of composition.

Definition 1.4.1. A *category* $\mathbf{C}$ is comprised of a collection of *objects* $\text{Ob}(\mathbf{C})$ and, for each pair of objects $A, B \in \text{Ob}(\mathbf{C})$, a collection of *morphisms* $\mathbf{C}(A, B)$, together with a *composition* operation

$$\circ: \mathbf{C}(B, C) \times \mathbf{C}(A, B) \rightarrow \mathbf{C}(A, C)$$

for all $A, B, C \in \text{Ob}(\mathbf{C})$ satisfying the following axioms.

- *Identities:* For every object A , there is a special morphism $\text{id}_A \in \mathbf{C}(A, A)$ so that for all $A, B \in \text{Ob}(\mathbf{C})$ and for all $f \in \mathbf{C}(A, B)$,

$$f \circ \text{id}_A = f = \text{id}_B \circ f. \quad (1.2)$$

- **Associativity:** For all $A, B, C, D \in \text{Ob}(\mathbf{C})$ and all $f \in \mathbf{C}(A, B)$, $g \in \mathbf{C}(B, C)$ and $h \in \mathbf{C}(C, D)$,

$$(h \circ g) \circ f = h \circ (g \circ f). \quad (1.3)$$

For any morphism $f \in \mathbf{C}(A, B)$ we refer to $f: A \rightarrow B$ as its *type*. To reduce clutter, $A \in \mathbf{C}$ is often used to indicate that A is an object of $\mathbf{C}$.

Example 1.4.2. The category **Set** has sets as objects and each $\mathbf{Set}(A, B)$ is the set of all functions of domain A and codomain B . Composition of functions $(g \circ f)(a) = g(f(a))$ is clearly associative and its identities $\text{id}_A: A \rightarrow A$ are the usual identity functions.

Example 1.4.3. The category **Rel** has sets as objects and each $\mathbf{Rel}(A, B)$ is the set of all relations between set A and set B , *i.e.* $\mathbf{Rel}(A, B)$ is the powerset of $A \times B$. The composite of $\mathcal{R}: A \rightarrow B$ and $\mathcal{S}: B \rightarrow C$ is given as follows:

$$\mathcal{S} \circ \mathcal{R} = \{(a, c) \in A \times C \mid \exists b \in B, a\mathcal{R}b \text{ and } b\mathcal{S}c\}.$$

Composition is clearly associative and identities are relations $\text{id}_A = \{(a, a) \mid a \in A\}$.

Example 1.4.4. The category **Mon** has monoids as objects and each $\mathbf{Mon}(A, B)$ is the set of all monoid homomorphisms from A to B . Composition and identities are inherited from **Set**. Similarly, there is a category **CMon** of commutative monoids, a category **Grp** of groups and a category **Ab** of abelian groups, all of which have monoid homomorphisms as their morphisms.

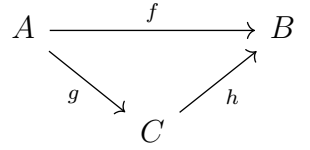
Example 1.4.5. Fix a field K ; the category $\mathbf{Vect}_K$ has vector spaces over the field K as objects and each $\mathbf{Vect}_K(V, U)$ is the set all linear maps from the vector space V to the vector space U . Composition is inherited from **Set**, along with identities. Whenever **Vect** appears in this text without subscript it should be understood that $K = \mathbb{C}$.

Remark 1.4.6. There is some subtlety hidden in the word *collection* in the definition of a category. If we consider the example of the category **Set**, we cannot say that $\text{Ob}(\mathbf{Set})$ is a set itself, as we would run into Russell's paradox.³ The way out of this conundrum is to define some sort of hierarchy of collections, where what we commonly refer to as a set is at the base level, and the collection of all sets is one step

³Loosely speaking, if it were a set, it would be the “set containing all sets”. Thus, it would need to be contained in itself, which creates a great deal of problems.

up the ladder. A category where the collection of objects is actually a set is called a *small category*. Similarly, it can happen that a category $\mathbf{C}$ satisfies that for all objects $A, B \in \mathbf{C}$, the collection $\mathbf{C}(A, B)$ is actually a set; then $\mathbf{C}$ is said to be a *locally small category*. $\mathbf{Set}$, $\mathbf{Rel}$ and $\mathbf{Vect}$ are all locally small categories, but not small. All of the categories discussed in this thesis are locally small. When working with locally small categories, we refer to each set $\mathbf{C}(A, B)$ as a *hom-set*.

Definition 1.4.7. Let $\mathbf{C}$ be a category. A *diagram* is a graph where the vertices are objects in $\mathbf{C}$ and the arrows are morphisms in the direction determined by their type. For instance,



A *commuting diagram* satisfies that composing all morphisms along a path results in the same morphism obtained along any other path between the same endpoints. In particular, the diagram above commutes if and only if $f = h \circ g$.

From the examples above, it is reasonable to consider a notion of mapping between categories so that, for instance, we may say that all monoid homomorphisms are functions, and that all functions are relations.

Definition 1.4.8. Let $\mathbf{C}$ and $\mathbf{D}$ be two categories. A *functor* $F: \mathbf{C} \rightarrow \mathbf{D}$ is comprised of a mapping between objects so that if $A \in \mathbf{C}$ then $F(A) \in \mathbf{D}$, and a mapping between morphisms so that if $f \in \mathbf{C}(A, B)$ then $F(f) \in \mathbf{D}(F(A), F(B))$. For it to be a functor, F must preserve composition and identities:

$$F(g \circ f) = F(g) \circ F(f)$$

$$F(\text{id}_A) = \text{id}_{F(A)}.$$

A *faithful functor* satisfies that the mapping $\mathbf{C}(A, B) \rightarrow \mathbf{D}(F(A), F(B))$ is injective for each pair of objects $A, B \in \mathbf{C}$. A *full functor* satisfies that the mapping $\mathbf{C}(A, B) \rightarrow \mathbf{D}(F(A), F(B))$ is surjective for each pair of objects $A, B \in \mathbf{C}$.

Example 1.4.9. There is a faithful functor $\mathbf{Mon} \rightarrow \mathbf{Set}$ that sends each monoid to its underlying set, and each homomorphism to its underlying function. Because composition in $\mathbf{Mon}$ is defined in the same manner as in $\mathbf{Set}$, proving that this is a functor is trivial.

Example 1.4.10. There is a full and faithful functor $\mathbf{Ab} \rightarrow \mathbf{CMon}$ that sends each abelian group to its underlying monoid — which will automatically be commutative — and acts as the identity on morphisms. The morphisms in $\mathbf{Ab}$ are the same as those in $\mathbf{CMon}$, so it is trivial to show that this is indeed a functor.

Example 1.4.11. There is a faithful functor $\mathbf{Set} \rightarrow \mathbf{Rel}$ that sends each set to itself and each function $f: A \rightarrow B$ to its *graph*, i.e. to the relation:

$$\mathcal{R}_f = \{(a, b) \in A \times B \mid f(a) = b\}.$$

The graph of the identity function matches the identity relation; it is easy to check that composition is preserved.

Definition 1.4.12. Let $\mathbf{C}$ and $\mathbf{D}$ be categories. We say that $\mathbf{C}$ is a *subcategory* of $\mathbf{D}$ if and only if $\text{Ob}(\mathbf{C})$ is a subcollection of $\text{Ob}(\mathbf{D})$ and $\mathbf{C}(A, B)$ is a subcollection of $\mathbf{D}(A, B)$ for each pair of objects $A, B \in \mathbf{C}$ and composition and identities in $\mathbf{C}$ coincide with those in $\mathbf{D}$. If $\mathbf{C}$ is a subcategory of $\mathbf{D}$ then there is a canonical faithful functor $\mathbf{C} \hookrightarrow \mathbf{D}$ that acts as the identity on objects and morphisms. If the functor is also full, we refer to $\mathbf{C}$ as a *full subcategory* of $\mathbf{D}$.

Example 1.4.13. $\mathbf{Set}$ is a subcategory of $\mathbf{Rel}$. Both $\mathbf{CMon}$ and $\mathbf{Grp}$ are full subcategories of $\mathbf{Mon}$. $\mathbf{Ab}$ is a full subcategory both of $\mathbf{CMon}$ and $\mathbf{Grp}$.

The usefulness of functors goes beyond the simple case of embedding a category into a larger one; intuitively, a functor $F: \mathbf{C} \rightarrow \mathbf{D}$ indicates that there is a way to go from $\mathbf{C}$ to $\mathbf{D}$ preserving the structure in the category $\mathbf{C}$. Currently, the structure of a category is only determined by its composition but, soon enough, richer flavours of categories will be discussed and, along them, there will be refined notions of functors preserving the new structure.

It will often be useful to discuss how pairs of functors $F: \mathbf{C} \rightarrow \mathbf{D}$ and $G: \mathbf{C} \rightarrow \mathbf{D}$ are related, this is done via natural transformations.

Definition 1.4.14. Let $\mathbf{C}$ and $\mathbf{D}$ be categories, let $F: \mathbf{C} \rightarrow \mathbf{D}$ and $G: \mathbf{C} \rightarrow \mathbf{D}$ be functors. A *natural transformation* $F \xrightarrow{\alpha} G$ is a collection of morphisms $\alpha_A: F(A) \rightarrow G(A)$ for each $A \in \mathbf{C}$ such that the following diagram commutes for every $f \in \mathbf{C}(A, B)$,

$$\begin{array}{ccc} F(A) & \xrightarrow{\alpha_A} & G(A) \\ \downarrow F(f) & & \downarrow G(f) \\ F(B) & \xrightarrow{\alpha_B} & G(B) \end{array}$$

We say α is a *natural isomorphism* if, additionally, all morphism α_A are invertible.⁴

An important concept in category theory is the notion of limit; their definition is sketched below. For a proper introduction to categorical limits see Chapter 5 of Leinster's book [38].

Definition 1.4.15. Let $\mathbf{C}$ be a category. For any arbitrary diagram in $\mathbf{C}$ (see Definition 1.4.7), let $\{A_i \in \mathbf{C}\}_{i \in I}$ be the set of objects in it, *e.g.*

$$\begin{array}{ccc} & A_0 & \\ & \downarrow f & \\ A_1 & \xrightarrow{f'} & A_2. \end{array}$$

A *cone* is an object $X \in \mathbf{C}$ along with a set of morphisms $\{g_i: X \rightarrow A_i\}_{i \in I}$ such that, when these are included in the diagram, *e.g.*

$$\begin{array}{ccccc} X & \xrightarrow{g_0} & A_0 & & \\ g_1 \downarrow & \searrow g_2 & \downarrow f & & \\ A_1 & \xrightarrow{f'} & A_2 & & \end{array}$$

it is satisfied for each $i \in I$ that all paths from X to A_i yield the same morphism. A *categorical limit* is a cone $\{h_i: L \rightarrow A_i\}_{i \in I}$ such that for any other cone $\{g_i: X \rightarrow A_i\}_{i \in I}$ there is a unique morphism $m: X \rightarrow L$ satisfying $g_i = h_i \circ m$ for all $i \in I$.

The limit of certain diagrams have special names; for instance, for any two objects $A, B \in \mathbf{C}$ the limit of the diagram

$$A \quad B$$

is a *categorical product* and, for any two morphisms $f, g: A \rightarrow B$, the limit of the diagram

$$A \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} B$$

is known as an *equalizer*. In general, not every diagram in a category $\mathbf{C}$ has a limit; when it does, we say that $\mathbf{C}$ is a *complete category*. A *colimit* is the dual notion of a limit, defined by inverting the direction of all morphisms in Definition 1.4.15. The dual of a product is a coproduct and the dual of an equalizer is a coequalizer. An object that is both a product and a coproduct and satisfies certain additional algebraic identities is known as a *biproduct*.

⁴A morphism $f: A \rightarrow B$ is invertible if there is another morphism $g: B \rightarrow A$ in the category such that $g \circ f = \text{id}_A$ and $f \circ g = \text{id}_B$. Such a morphism g is often denoted f^{-1} and referred to as the inverse of f . It is straightforward to check that inverses are unique.

1.4.1 Monoidal categories

We will often find that ‘sequential’ composition of morphisms $- \circ -$ is not the only reasonable notion of composition that may be considered. The notion of ‘parallel’ composition is formalised in monoidal categories.

Definition 1.4.16. Let $\mathbf{C}$ be a category, let $\otimes: \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ be a functor and let I be an object in $\mathbf{C}$. The triple $(\mathbf{C}, \otimes, I)$ is a *monoidal category* if there are natural isomorphisms α , λ and ρ — known respectively as associator, left unitor and right unitor — such that the diagrams below commute for all choices of objects $A, B, C, D \in \mathbf{C}$. We refer to $\otimes$ as the *monoidal product* and I as the *monoidal unit*.

$$\begin{array}{ccc}
 (A \otimes I) \otimes B & \xrightarrow{\alpha_{A,I,C}} & A \otimes (I \otimes B) \\
 \searrow \rho_A \otimes \text{id}_B & & \swarrow \text{id}_A \otimes \lambda_B \\
 & A \otimes B &
 \end{array}$$

$$\begin{array}{ccccc}
 & & (A \otimes B) \otimes (C \otimes D) & & \\
 & \nearrow \alpha_{A \otimes B, C, D} & & \searrow \alpha_{A, B, C \otimes D} & \\
 ((A \otimes B) \otimes C) \otimes D & & & & A \otimes (B \otimes (C \otimes D)) \\
 \downarrow \alpha_{A, B, C} \otimes \text{id}_D & & & & \uparrow \text{id}_A \otimes \alpha_{B, C, D} \\
 (A \otimes (B \otimes C)) \otimes D & \xrightarrow{\alpha_{A, B \otimes C, D}} & & & A \otimes ((B \otimes C) \otimes D)
 \end{array}$$

The requirement that $\otimes: \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ is a functor imposes an interchange law between $\circ$ and $\otimes$:

$$(g \otimes k) \circ (f \otimes h) = (g \circ f) \otimes (k \circ h). \quad (1.4)$$

The intuition behind this equation is provided in Figure 1.2. The first of the commuting diagrams above imposes that the object I acts as the unit of ‘parallel’ composition $- \otimes -$, whereas the second commuting diagram imposes that $\otimes$ is associative.

Remark 1.4.17. By convention, morphisms in monoidal categories are represented pictorially as in Figure 1.2. Morphisms are depicted as labelled boxes connected via wires when composed (using $\circ$) and, when combined using the monoidal product, the morphisms are drawn one on top of the other. Thanks to the coherence theorem of monoidal categories (see Section 1.3.4 from [31]), we may prove facts about morphisms in a monoidal category $\mathbf{C}$ by manipulating the pictorial representation of their

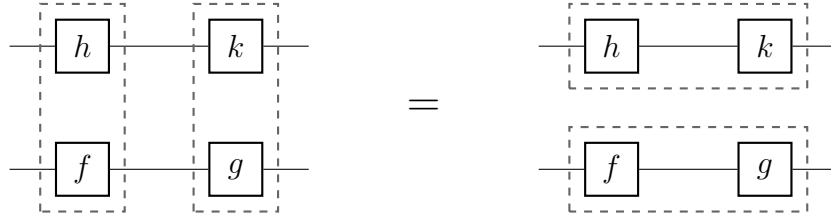


Figure 1.2: Pictorial representation of the interchange law (1.4). It does not matter whether we first compose in parallel or sequentially, the result is the same.

morphisms. This is known as the *graphical calculus*; its main advantage is that associators, unitors and the interchange law (1.4) become trivial when represented pictorially, removing unnecessary verbosity from definitions and proofs.

Example 1.4.18. Let $\times: \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ be the functor acting on objects as the Cartesian product of sets. On morphisms $f: A \rightarrow B$ and $g: C \rightarrow D$, the functor yields a function $f \times g$ that maps each $(a, c) \in A \times C$ to $(f(a), g(c)) \in B \times D$. Let $\{\bullet\}$ be an arbitrary singleton set; then, $(\mathbf{Set}, \times, \{\bullet\})$ is a monoidal category.

Example 1.4.19. Let $+: \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ be the functor acting on objects as the disjoint union of sets. On morphisms $f: A \rightarrow B$ and $g: C \rightarrow D$, the functor yields a function $f + g$ that maps each $a \in A$ to $f(a)$ and each $c \in C$ to $g(c)$. It can be shown that $(\mathbf{Set}, +, \emptyset)$ is a monoidal category.

Definition 1.4.20. Given two vector spaces $A, B \in \mathbf{Vect}_K$, their tensor product $A \otimes B$ is a vector space characterised (up to isomorphism) by the property that there is a bilinear map⁵ $p: A \times B \rightarrow A \otimes B$ such that, for any other bilinear map $f: A \times B \rightarrow C$, there is a unique linear map $g: A \otimes B \rightarrow C$ making the diagram

$$\begin{array}{ccc} A \times B & \xrightarrow{p} & A \otimes B \\ & \searrow f & \downarrow g \\ & & C \end{array}$$

commute in $\mathbf{Set}$. For any two linear maps $f: A \rightarrow B$ and $g: C \rightarrow D$, the map $f \otimes g: A \otimes C \rightarrow B \otimes D$ is defined as the linear extension of the map $a \otimes c \mapsto f(a) \otimes g(c)$.

Example 1.4.21. Let $\otimes: \mathbf{Vect}_K \times \mathbf{Vect}_K \rightarrow \mathbf{Vect}_K$ be the functor acting as the tensor product on vector spaces and linear maps. It can be shown that $(\mathbf{Vect}_K, \otimes, K)$ is a monoidal category.

⁵A bilinear map is a map $f: A \times B \rightarrow C$ such that, for all $a \in A$ and all $b \in B$, both $f(a, -)$ and $f(-, b)$ are linear.

Definition 1.4.22. Given two vector spaces $A, B \in \mathbf{Vect}_K$, their *direct sum* $A \oplus B$ is the vector space on the Cartesian product of their underlying sets, with coordinate-wise addition and scalar multiplication. For any two linear maps $f: A \rightarrow B$ and $g: C \rightarrow D$, the linear map $f \oplus g: A \oplus C \rightarrow B \oplus D$ is defined as follows:

$$(f \oplus g)(a, c) = (f(a), g(c)).$$

Example 1.4.23. Let $\oplus: \mathbf{Vect}_K \times \mathbf{Vect}_K \rightarrow \mathbf{Vect}_K$ be the functor acting as the direct sum of vector spaces and linear maps and let $\{0\}$ be the zero-dimensional vector space. It can be shown that $(\mathbf{Vect}_K, \oplus, \{0\})$ is a monoidal category.

We have seen two distinct monoidal structures both for **Set** and **Vect**. Both monoidal structures in **Vect** are relevant to this thesis, with the $\otimes$ -monoidal structure being the usual focus in the literature of categorical quantum mechanics [31, 18]. However, when discussing the control flow of programs, the monoidal structure induced by the direct sum is the one to take the spotlight and, hence, the one that will be most relevant to this thesis.

Definition 1.4.24. Let $(\mathbf{C}, \otimes, I)$ be a monoidal category and let σ be a natural isomorphism with components

$$\sigma_{A,B}: A \otimes B \rightarrow B \otimes A.$$

We say $\mathbf{C}$ is a *braided monoidal category* if σ satisfies the coherence axioms given below. We refer to σ as braiding and represent it pictorially as the crossing of wires; the coherence axioms are represented pictorially as:

The image contains two equations of string diagrams. The left equation shows the associativity of braiding: a crossing between wires A and B, followed by a crossing between B and C, is equal to a crossing between A and B⊗C, followed by a crossing between B⊗C and C. The right equation shows the naturality of braiding: a crossing between wires C and B, followed by a crossing between B and A, is equal to a crossing between C and A⊗B, followed by a crossing between A⊗B and C.

We say $\mathbf{C}$ is a *symmetric monoidal category* if, additionally, $\sigma_{A,B}^{-1} = \sigma_{B,A}$ for all pairs of objects.

Both **Set** and **Vect** are symmetric monoidal categories with either of the monoidal structures discussed in the examples above. Along each new flavour of categories comes a refined notion of structure-preserving functor.

Definition 1.4.25. Let $(\mathbf{C}, \otimes_{\mathbf{C}}, I_{\mathbf{C}})$ and $(\mathbf{D}, \otimes_{\mathbf{D}}, I_{\mathbf{D}})$ be monoidal categories and let $F: \mathbf{C} \rightarrow \mathbf{D}$ be a functor. Let μ be a natural transformation with components

$$\mu_{A,B}: F(A) \otimes_{\mathbf{D}} F(B) \rightarrow F(A \otimes_{\mathbf{C}} B)$$

and let $\mu_I: I_{\mathbf{D}} \rightarrow F(I_{\mathbf{C}})$ be a morphism in $\mathbf{D}$. We say (F, μ) is a *lax monoidal functor* when μ commutes with the associators and unitors of $\mathbf{C}$ and $\mathbf{D}$ in an appropriate way, made precise in reference texts (see equations (1.29) and (1.30) from [31]) and omitted here for brevity. If μ_I and all $\mu_{A,B}$ are isomorphisms we say that (F, μ) is a *strong monoidal functor*. Similarly, a *braided monoidal functor* is a monoidal functor $(F, \mu): \mathbf{C} \rightarrow \mathbf{D}$ between braided monoidal categories such that if σ is the braiding in $\mathbf{C}$ then $F(\sigma)$ is the braiding in $\mathbf{D}$. A *symmetric monoidal functor* is a braided monoidal functor between symmetric monoidal categories.

There are many other flavours of categories, each of them adding more structure by introducing new operations and axioms. Category theorists define and classify different flavours of categories, they describe how different categories are related to each other and, most importantly, they study how properties arise from the structure that is imposed.

1.4.2 Enriched category theory

In this thesis, it will often be required to combine the contribution of two morphisms $f, g: A \rightarrow B$ to describe another morphism of type $A \rightarrow B$. Abstractly, for some category $\mathbf{C}$, this can be framed as requiring that each collection of morphisms $\mathbf{C}(A, B)$ is endowed with an operation

$$+: \mathbf{C}(A, B) \times \mathbf{C}(A, B) \rightarrow \mathbf{C}(A, B)$$

making each hom-set $(\mathbf{C}(A, B), +)$ a commutative monoid. When hom-sets are endowed with a mathematical structure and composition interacts nicely with it, we say $\mathbf{C}$ is an enriched category, formally defined below.

Definition 1.4.26. Let $(\mathbf{V}, \otimes, I)$ be a monoidal category. A *$\mathbf{V}$ -enriched category* $\mathbf{C}$ is comprised of a collection of objects $\text{Ob}(\mathbf{C})$ and, for each pair of objects $A, B \in \text{Ob}(\mathbf{C})$ a *hom-object* $\mathbf{C}(A, B) \in \text{Ob}(\mathbf{V})$. Composition and identities in $\mathbf{C}$ are determined by morphisms in $\mathbf{V}$,

$$\circ_{A,B,C}: \mathbf{C}(B, C) \otimes \mathbf{C}(A, B) \rightarrow \mathbf{C}(A, C)$$

$$1_A: I \rightarrow \mathbf{C}(A, A)$$

satisfying the following commuting diagrams in $\mathbf{V}$, which ensure that composition in

$\mathbf{C}$ is associative and that the identities are its units.

$$\begin{array}{ccc}
 (\mathbf{C}(C, D) \otimes \mathbf{C}(B, C)) \otimes \mathbf{C}(A, B) & \xrightarrow{\circ_{B,C,D} \otimes \text{id}} & \mathbf{C}(B, D) \otimes \mathbf{C}(A, B) \\
 \downarrow \alpha & & \downarrow \circ_{A,B,D} \\
 & & \mathbf{C}(A, D) \\
 & & \uparrow \circ_{A,C,D} \\
 \mathbf{C}(C, D) \otimes (\mathbf{C}(B, C) \otimes \mathbf{C}(A, B)) & \xrightarrow{\text{id} \otimes \circ_{A,B,C}} & \mathbf{C}(C, D) \otimes \mathbf{C}(A, C)
 \end{array}$$

$$\begin{array}{ccccc}
 \mathbf{C}(A, B) \otimes \mathbf{C}(A, A) & \xrightarrow{\circ_{A,A,B}} & \mathbf{C}(A, B) & \xleftarrow{\circ_{A,B,B}} & \mathbf{C}(B, B) \otimes \mathbf{C}(A, B) \\
 \uparrow \text{id} \otimes 1_A & \nearrow \rho & & \nwarrow \lambda & \uparrow 1_B \otimes \text{id} \\
 \mathbf{C}(A, B) \otimes I & & & & I \otimes \mathbf{C}(A, B)
 \end{array}$$

An immediate example of an $\mathbf{Ab}$ -enriched category is $\mathbf{Vect}$; the proof is sketched in Example 1.4.27 below. The monoidal structure in $\mathbf{Ab}$ must be chosen with care: even though the cartesian product of groups provides a valid monoidal structure, it is *not* the appropriate one to be used for the enrichment.⁶ Instead, the monoidal category to be used is $(\mathbf{Ab}, \otimes, \mathbb{Z})$ where $\otimes$ is the tensor product of abelian groups and $\mathbb{Z}$ is the group of integers with addition. The explicit definition of this tensor product can be found in [53]; for the purposes of this thesis, it suffices to say that $A \otimes B$ is special in that the set of linear maps in $\mathbf{Ab}(A \otimes B, C)$ is in one-to-one correspondence with the set of functions $f: A \times B \rightarrow C$ satisfying

$$\begin{aligned}
 f(a, b + b') &= f(a, b) + f(a, b'), & f(a, 0) &= 0, \\
 f(a + a', b) &= f(a, b) + f(a', b), & f(0, b) &= 0.
 \end{aligned} \tag{1.5}$$

Example 1.4.27. $\mathbf{Vect}$ is an $\mathbf{Ab}$ -enriched category where for any two linear maps $f, g: U \rightarrow V$ their addition is defined pointwise, *i.e.* for all $u \in U$:

$$(f + g)(u) = f(u) + f(v).$$

Indeed, this operation is associative and commutative, there is a linear map sending every vector to the zero vector — thus acting as the addition's neutral element — and every linear map f has an additive inverse $u \mapsto -f(u)$, making the set of all linear maps $U \rightarrow V$ an abelian group. Importantly, asking that composition in $\mathbf{Vect}$ is a

⁶Given two abelian groups A and B , their cartesian product $A \times B$ has its group operation defined as $(a, b) + (a', b') = (a + a', b + b')$. If we used this monoidal structure for enrichment, we would have that

$$h \circ (f + g) = \circ(h, f + g) = \circ((h, f) + (0, g)) = h \circ f + 0$$

which is not the correct way composition and addition of linear maps interact.

morphism in $\mathbf{Ab}$ along with (1.5) implies that composition distributes over addition, as expected:

$$\begin{aligned} h \circ (f + g) &= h \circ f + h \circ g, \\ (f + g) \circ h &= f \circ h + g \circ h. \end{aligned}$$

Chapter 2 will define multiple Σ_* categories capturing the notion of infinitary sum; an appropriate notion of tensor product of objects in these categories will be given in Section 2.1.5. Then, Chapter 3 will define Σ_* -enriched categories and use their notion of infinitary sum of morphisms to formalise iterative loops.

1.4.3 Adjunctions

Example 1.4.9 discussed the existence of a functor $\mathbf{Mon} \rightarrow \mathbf{Set}$ that ‘forgets’ the monoid structure, sending monoids and homomorphisms to their underlying sets and functions. A similar forgetful functor $\mathbf{CMon} \rightarrow \mathbf{Set}$ exists as well; we may entertain the idea of a functor on the opposite direction $\mathbf{Set} \rightarrow \mathbf{CMon}$ as described in the following example.

Example 1.4.28. Let A be an arbitrary set and let $F(A)$ be the collection of all finite multisets of elements in A . There is a functor $F: \mathbf{Set} \rightarrow \mathbf{CMon}$ that maps each set A to the commutative monoid on the set $F(A)$ with disjoint union as its monoid operation: disjoint union is associative and commutative, with $\emptyset \in F(A)$ acting as its neutral element. Each function $f: A \rightarrow B$ is lifted to a monoid homomorphism $F(f): F(A) \rightarrow F(B)$ that maps each multiset $\{a, b, \dots\} \in F(A)$ to $\{f(a), f(b), \dots\} \in F(B)$. It is trivial to check that $F(g \circ f) = F(g) \circ F(f)$ and $F(\text{id}_A) = \text{id}_{F(A)}$ so F is indeed a functor.

This functor $F: \mathbf{Set} \rightarrow \mathbf{CMon}$ satisfies a powerful property: it is left adjoint to the canonical forgetful functor $G: \mathbf{CMon} \rightarrow \mathbf{Set}$. The concept of categorical adjunction is defined below in terms of the initial object of a comma category.

Definition 1.4.29. Let $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ be arbitrary categories and let $F: \mathbf{A} \rightarrow \mathbf{C}$ and $G: \mathbf{B} \rightarrow \mathbf{C}$ be arbitrary functors. The *comma category* $(F \Rightarrow G)$ is defined as follows:

- objects are triples (A, h, B) where A and B are objects in $\mathbf{A}$ and $\mathbf{B}$ respectively and $h: F(A) \rightarrow G(B)$ is a morphism in $\mathbf{C}$;

- morphisms $(A, h, B) \rightarrow (A', h', B')$ are pairs (f, g) where $f: A \rightarrow A'$ is a morphism in $\mathbf{A}$ and $g: B \rightarrow B'$ is a morphism in $\mathbf{B}$ such that the diagram

$$\begin{array}{ccc} F(A) & \xrightarrow{F(f)} & F(A') \\ h \downarrow & & \downarrow h' \\ G(B) & \xrightarrow{G(g)} & G(B') \end{array}$$

commutes.

Let $\mathbf{1}$ be the category with a single object and a single morphism; for any arbitrary object $A \in \mathbf{C}$ we may define a functor $A: \mathbf{1} \rightarrow \mathbf{C}$ that maps the unique object in $\mathbf{1}$ to A and maps its unique morphism to id_A . Let $G: \mathbf{D} \rightarrow \mathbf{C}$ be a functor; the comma category $(A \Rightarrow G)$ is used often throughout the thesis. To reduce clutter, objects of $(A \Rightarrow G)$ are denoted by pairs (h, B) — since the domain of h is guaranteed to be A — and morphisms (id_A, g) are simply denoted g .

Definition 1.4.30. Let $\mathbf{C}$ and $\mathbf{D}$ be categories, let $1_{\mathbf{C}}: \mathbf{C} \rightarrow \mathbf{C}$ be the identity functor and let $F: \mathbf{C} \rightarrow \mathbf{D}$ and $G: \mathbf{D} \rightarrow \mathbf{C}$ be arbitrary functors. We say that F is *left adjoint* to G , denoted $F \dashv G$, if there is a natural transformation $1_{\mathbf{C}} \xrightarrow{\eta} GF$ such that, for every $A \in \mathbf{C}$, the object $(\eta_A, F(A))$ of the comma category $(A \Rightarrow G)$ is initial — i.e. for each object $(h, B) \in (A \Rightarrow G)$ there is a unique morphism $(\eta_A, F(A)) \rightarrow (h, B)$. The triple (F, G, η) is known as an *adjunction*, with η being its *unit*; equivalently, we say that G is *right adjoint* to F .

Unwrapping the definition we find that a unique morphism $(\eta_A, F(A)) \rightarrow (h, B)$ exists if and only if there is a unique morphism $\bar{h}: F(A) \rightarrow B$ in $\mathbf{D}$ that makes the diagram

$$\begin{array}{ccc} A & \xrightarrow{h} & G(B) \\ & \searrow \eta_A & \uparrow G(\bar{h}) \\ & & GF(A) \end{array}$$

commute.

Example 1.4.31. Let $F: \mathbf{Set} \rightarrow \mathbf{CMon}$ be the functor from Example 1.4.28 and let $G: \mathbf{CMon} \rightarrow \mathbf{Set}$ be the canonical forgetful functor; there is an adjunction $F \dashv G$. For every set A , the function $\eta_A: A \rightarrow GF(A)$ maps each $a \in A$ to $\{a\} \in GF(A)$; it is straightforward to check that $1_{\mathbf{Set}} \xrightarrow{\eta} GF$ is a natural transformation. Let $(B, +)$

be an arbitrary commutative monoid; for any function $f: A \rightarrow B$ the requirement that the diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & G(B) \\ & \searrow \eta_A & \uparrow G(\bar{f}) \\ & & GF(A) \end{array}$$

commutes implies that $\bar{f}$ must map each $\eta_A(a) = \{a\}$ to $f(a)$. Moreover, the requirement that $\bar{f}$ is a monoid homomorphism implies that

$$\bar{f}(\{a, a'\}) = \bar{f}(\{a\} \uplus \{a'\}) = \bar{f}(\{a\}) + \bar{f}(\{a'\}) = f(a) + f(a').$$

This may be extended to any element in $F(A)$, *i.e.* any finite multiset of elements in A . Such a monoid homomorphism $\bar{f}$ is well-defined and it is the unique one satisfying the commuting diagram above so, indeed, (F, G, η) is an adjunction.

Example 1.4.32. The canonical embedding functor $G: \mathbf{Ab} \hookrightarrow \mathbf{CMon}$ from Example 1.4.10 has a left adjoint $F: \mathbf{CMon} \rightarrow \mathbf{Ab}$ explicitly defined below.

- Let $(A, +)$ be a commutative monoid and define an equivalence relation $\sim$ such that for all pairs $(a_+, a_-), (b_+, b_-) \in A \times A$:

$$(a_+, a_-) \sim (b_+, b_-) \iff \exists z \in A \text{ s.t. } a_+ + b_- + z = b_+ + a_- + z.$$

- $F(A)$ is the abelian group $((A \times A)/\sim, +')$ where

$$[(a_+, a_-)] +' [(b_+, b_-)] = [(a_+ + b_+, a_- + b_-)].$$

It can be shown that the result of $+'$ is independent of the choice of representatives and, hence, it is well-defined. Then, $F(A)$ is an abelian group whose neutral element is $[(0, 0)]$, while the inverse of any $[(a_+, a_-)]$ is $[(a_-, a_+)]$.

- Let $f: A \rightarrow B$ be a monoid homomorphism, then $F(f)$ is the monoid homomorphism that maps each $[(a_+, a_-)] \in F(A)$ to $[(f(a_+), f(a_-))] \in F(B)$.

For each $A \in \mathbf{CMon}$ the morphism $\eta_A: A \rightarrow GF(A)$ maps each $a \in A$ to $[(a, 0)] \in F(A)$. Let $(B, +)$ be a group; for any monoid homomorphism $f: A \rightarrow G(B)$, the corresponding group homomorphism $\bar{f}: F(A) \rightarrow B$ is

$$\bar{f}([(a_+, a_-)]) = g(a_+) - g(a_-). \quad (1.6)$$

It is straightforward to check that η is a natural transformation and that $\bar{f}$ is the unique group homomorphism such that $f = G(\bar{f}) \circ \eta_A$, implying that $F \dashv G$.

The functor $FG: \mathbf{Ab} \rightarrow \mathbf{Ab}$ obtained by ‘forgetting’ the existence of inverses — via $G: \mathbf{Ab} \hookrightarrow \mathbf{CMon}$ — then recovering them using $F: \mathbf{CMon} \rightarrow \mathbf{Ab}$ yields the original abelian group up to isomorphism. This is the case whenever a full and faithful functor has a left adjoint. In particular, the embedding functor of every full subcategory is full and faithful; thus, full subcategories whose embedding has a left adjoint are of particular interest.

Definition 1.4.33. Let $\mathbf{C}$ be a category and let $\mathbf{D}$ be a subcategory of $\mathbf{C}$ with $G: \mathbf{D} \hookrightarrow \mathbf{C}$ being its corresponding embedding. We say $\mathbf{D}$ is a *reflective subcategory* of $\mathbf{C}$ if G is full and it has a left adjoint.

Thus, $\mathbf{Ab}$ is a reflective subcategory of $\mathbf{CMon}$. Multiple examples of reflective subcategories will be discussed in Chapter 2.

In Examples 1.4.31 and 1.4.32 the existence of a left adjoint has been proven by defining it explicitly. However, it is sometimes convenient to establish the existence of an adjunction via non-constructive means since, when working at a higher level of abstraction, cumbersome details may be avoided. A powerful tool to this end is the general adjoint functor theorem, which is introduced below. The statement of this result (and its proof) make use of the concept of categorical limit and complete categories; these were introduced in Definition 1.4.15 and are covered in depth in most books on basic category theory (for instance, see Chapter 5 from [38]).

Definition 1.4.34. Let $\mathbf{C}$ be a category. A *weakly initial set* is a set S of objects in $\mathbf{C}$ satisfying that for every object $B \in \mathbf{C}$ there is at least one morphism $A \rightarrow B$ such that $A \in S$.⁷

Lemma 1.4.35. *Let $\mathbf{C}$ be a complete locally small category with a weakly initial set. Then $\mathbf{C}$ has an initial object.*

Proof. See Lemma A.1 from the book on basic category theory by Leinster [38]. $\square$

The previous lemma is the main technical result used in the proof of the general adjoint functor theorem. Moreover, the lemma will be relevant on its own in Section 2.1.5.

Theorem 1.4.36. *Let $G: \mathbf{D} \rightarrow \mathbf{C}$ be a functor. Assume that $\mathbf{D}$ is complete and locally small and that for each object $A \in \mathbf{C}$ the comma category $(A \Rightarrow G)$ has a weakly initial set. Then,*

$$G \text{ has a left adjoint} \iff G \text{ preserves limits.}$$

⁷It is essential that it is an actual set, i.e. a *small* collection.

Proof. (Sketch). A detailed proof can be found in the appendix of the book on basic category theory by Leinster [38]. One direction is immediate since every right adjoint functor preserves limits (see Theorem 6.3.1 from [38]). To prove the other direction, we assume that G preserves limits; then, $\mathbf{D}$ being complete implies that the comma category $(A \Rightarrow G)$ is complete (see Lemma A.2 from [38]) and, since $\mathbf{D}$ is locally small, it follows that $(A \Rightarrow G)$ is locally small. Then, for all $A \in \mathbf{C}$ the existence of a weakly initial set in $(A \Rightarrow G)$ implies that $(A \Rightarrow G)$ has an initial object, as established by Lemma 1.4.35. According to Corollary 2.3.7 from [38] if for all $A \in \mathbf{C}$ the comma category $(A \Rightarrow G)$ has an initial object then a functor $F: \mathbf{C} \rightarrow \mathbf{D}$ and a natural transformation $1_{\mathbf{C}} \xrightarrow{\eta} GF$ exist such that (F, G, η) is an adjunction. $\square$

In general, it is preferable to provide constructive proofs when attainable since they tend to be more illustrative. Unfortunately, for some of the left adjoint functors whose existence is established in Chapter 2 an explicit definition could not be achieved and their existence is instead proven via the general adjoint functor theorem.

1.5 Categories in quantum computer science (*Preamble*)

As discussed in Section 1.1, the state of a quantum computer is described by a vector in a Hilbert space and operators are described by linear maps. Therefore, the following categories are of great importance in quantum computing.

Definition 1.5.1. Let **Hilb** be the category whose objects are complex Hilbert spaces and whose morphisms are bounded linear maps. Let **Contraction** be the subcategory of **Hilb** obtained by restricting morphisms to (weak) contractions; a linear map $f: A \rightarrow B$ between normed vector spaces is a (weak) *contraction* if it satisfies $\|f(a)\|_B \leq \|a\|_A$ for all $a \in A$.⁸ Let **Isometry** be the subcategory of **Contraction** obtained by restricting morphisms to isometries; a linear map $f: A \rightarrow B$ between normed vector spaces is an *isometry* if for all $a \in A$ it satisfies $\|f(a)\|_B = \|a\|_A$. Let **Unitary** be the subcategory of **Isometry** obtained by restricting morphisms to unitaries; a linear map is *unitary* if it is a surjective isometry.

Evidently, there is a chain of faithful functors:

$$\mathbf{Unitary} \hookrightarrow \mathbf{Isometry} \hookrightarrow \mathbf{Contraction} \hookrightarrow \mathbf{Hilb} \rightarrow \mathbf{Vect}.$$

⁸A *strict* contraction f satisfies $\|f(a)\|_B < \|a\|_A$ for all nonzero $a \in A$. In this thesis, the term ‘contraction’ refers to weak contractions unless stated otherwise.

Considering that real-world quantum computers have finite memory, quantum computer science tends to deal with categories whose objects are finite-dimensional vector spaces.

Definition 1.5.2. Let $\mathbf{FdHilb}$ be the full subcategory of $\mathbf{Hilb}$ obtained by restricting objects to finite-dimensional Hilbert spaces. Similarly, let $\mathbf{FdContraction}$, $\mathbf{FdIsometry}$ and $\mathbf{FdUnitary}$ be the corresponding full subcategories of $\mathbf{Contraction}$, $\mathbf{Isometry}$ and $\mathbf{Unitary}$, respectively.

All of these categories can be given a monoidal structure in terms of the (orthogonal) direct sum of Hilbert spaces.

Definition 1.5.3. For any two Hilbert spaces $A, B \in \mathbf{Hilb}$, let $A \oplus B$ be their direct sum, made into a Hilbert space by the inner product $\langle (a, b) | (a', b') \rangle = \langle a | a' \rangle + \langle b | b' \rangle$.

Proposition 1.5.4. Let $\mathbf{C}$ be either $\mathbf{Hilb}$, $\mathbf{Contraction}$, $\mathbf{Isometry}$, $\mathbf{Unitary}$ or the finite-dimensional version of any of them. Then, $(\mathbf{C}, \oplus, \{0\})$ is a symmetric monoidal category.

Proof. It is straightforward to check that $\oplus$ is a valid monoidal product and that the zero-dimensional Hilbert space $\{0\}$ is its monoidal unit so that $(\mathbf{Hilb}, \oplus, \{0\})$ is a symmetric monoidal category. Let $f: A \rightarrow B$ and $g: C \rightarrow D$ be morphisms in $\mathbf{Contraction}$; then, for all $a \in A$ and $c \in C$:

$$\|(f \oplus g)(a, c)\|_{B \oplus D}^2 = \|f(a)\|_B^2 + \|g(c)\|_D^2 \leq \|a\|_A^2 + \|c\|_C^2 = \|(a, c)\|_{A \oplus C}^2$$

implying that $f \oplus g$ is a morphism in $\mathbf{Contraction}$. Thus, $(\mathbf{Contraction}, \oplus, \{0\})$ is a symmetric monoidal subcategory of $(\mathbf{Hilb}, \oplus, \{0\})$. Similarly, if f and g are isometries then $f \oplus g$ is an isometry, and if both f and g are surjective (so that they are unitary) then $f \oplus g$ is also surjective and, hence, unitary. Consequently, both $(\mathbf{Isometry}, \oplus, \{0\})$ and $(\mathbf{Unitary}, \oplus, \{0\})$ are symmetric monoidal subcategories of $(\mathbf{Contraction}, \oplus, \{0\})$. Evidently, each of the subcategories where objects are finite-dimensional are symmetric monoidal as well. $\square$

For all objects $A, B \in \mathbf{Hilb}$ the object $A \oplus B \in \mathbf{Hilb}$ is a biproduct. In contrast, $A \oplus B$ is not a biproduct in $\mathbf{Contraction}$ (nor in its subcategories), since the diagonal morphism $a \mapsto (a, a)$ is not a contraction; but $A \oplus B$ is still a coproduct both in $\mathbf{Contraction}$ and $\mathbf{Isometry}$. In $\mathbf{Unitary}$, $A \oplus B$ is not even a coproduct, since injections $A \rightarrow A \oplus B$ are not surjective and, hence, not unitary. Even though $A \oplus B$

is not a product in **Contraction**, the projection $A \oplus B \rightarrow A$ from **Hilb** is still a morphism in **Contraction**; Hagverdi [28] and other authors have noticed that this is sufficient for morphisms in **Contraction** to have a unique matrix decomposition (more on this in Chapter 3). This hints at the importance of **Contraction** in quantum computing: even though strict contractions are not physical operations, every isometry and unitary is a morphism in **Contraction** and the entries in their block matrix decomposition are (possibly strict) contractions.

Apart from the monoidal structure induced by direct sum, **Hilb** can be given a monoidal structure using tensor product of Hilbert spaces.

Proposition 1.5.5. *Let $A, B \in \mathbf{Hilb}$ and let $A \boxtimes B$ be the tensor product of their underlying vector spaces made into an inner product space by defining*

$$\langle a \boxtimes b | a' \boxtimes b' \rangle = \langle a | a' \rangle \cdot \langle b | b' \rangle.$$

The inner product in $A \boxtimes B$ induces a norm, making it a metric space. The metric completion of $A \boxtimes B$ (see Theorem 43.7 from [48]) is a Hilbert space denoted $A \otimes B$ whose inner product is the canonical extension of that of $A \boxtimes B$.⁹

The monoidal category $(\mathbf{Hilb}, \otimes, \mathbb{C})$ captures the notion of combining the state space $A, B \in \mathbf{Hilb}$ of two distinct quantum memories into a single state space $A \otimes B \in \mathbf{Hilb}$ (see Section 1.1). In contrast, $(\mathbf{Hilb}, \oplus, \{0\})$ is the monoidal structure that captures the notion of parallel composition arising in quantum walks (see Section 1.1.2) and the orthogonal decomposition of a state space with respect to a predicate (see Section 1.1). Consequently, $(\mathbf{Hilb}, \oplus, \{0\})$ is the appropriate monoidal category to use when discussing control flow of quantum programs.

1.5.1 Beyond coherent quantum operations

Measurements are not coherent operations — projections are generally not invertible — and the outcome they yield is probabilistic. A general mathematical framework where measurements and coherent operations are described on equal footing needs to capture not only ‘pure’ quantum states $|\psi\rangle \in H$ but, more generally, probability distributions of quantum states.

Let H be a finite-dimensional complex Hilbert space and let $B(H)$ be the vector space of bounded linear maps $H \rightarrow H$; in the general framework, a state is represented

⁹If A and B are finite-dimensional, $A \otimes B = A \boxtimes B$.

by a *density operator* $\rho \in B(H)$: a positive semi-definite map that satisfies $\text{tr}(\rho) = 1$.¹⁰ Let $|\psi\rangle \in H$ be a unit vector and let $\rho \in B(H)$ be a positive density operator; then it is guaranteed that:

$$0 \leq \langle \psi | \rho | \psi \rangle \leq 1.$$

Moreover, for any orthonormal basis of H , say $\{|\psi_i\rangle\}_{i \in I}$, the trace of ρ satisfies:

$$\text{tr}(\rho) = \sum_{i \in I} \langle \psi_i | \rho | \psi_i \rangle = 1.$$

The value $\langle \psi_i | \rho | \psi_i \rangle \in [0, 1]$ for each $i \in I$ describes the probability of obtaining the pure quantum state $|\psi_i\rangle$ as the result of measuring ρ in the chosen basis. In this framework:

- all pure quantum states $|\varphi\rangle \in H$ may be represented as density operators $|\varphi\rangle\langle\varphi| \in B(H)$ and, more generally,
- for any finite set of pure states $\{|\varphi_i\rangle \in H\}_{i \in I}$ and any probability distribution $p: I \rightarrow [0, 1]$ assigned to them,

$$\rho = \sum_{i \in I} p(i) \cdot |\varphi_i\rangle\langle\varphi_i|$$

is a density operator. We refer to these as *mixed* quantum states.

Given two Hilbert spaces H and K , any density operator in the vector space $B(H) \oplus B(K)$ is a pair (ρ, ρ') where $\rho \in B(H)$ and $\rho' \in B(K)$ are positive semi-definite operators satisfying $\text{tr}(\rho) + \text{tr}(\rho') = 1$. Thus, density operators $(\rho, \rho') \in B(H) \oplus B(K)$ describe mixed states that are known to be either in $B(H)$ with probability $\text{tr}(\rho)$ or in $B(K)$ with probability $\text{tr}(\rho')$. Notice that $B(H) \oplus B(K)$ is strictly contained in $B(H \oplus K)$; in particular, pure states $|\phi\rangle\langle\phi| \in B(H \oplus K)$ are not in $B(H) \oplus B(K)$. Thus, in certain situations — such as after applying a measurement that projects a state to either H or K — we will use $B(H) \oplus B(K)$ to indicate that we are certain that the state is a probabilistic mixture of quantum states.

Remark 1.5.6. The state spaces we are interested in are direct sums $\oplus_{i \in I} B(H_i)$ where $\{H_i\}_{i \in I}$ is a finite collection of finite-dimensional Hilbert spaces. Any such space can be made into a C^* -algebra by defining its multiplication to be composition of maps and defining its involution to yield the adjoint of a map. In fact, any finite-dimensional C^* -algebra is isomorphic to the one defined from some $\oplus_{i \in I} B(H_i)$ as above. Thus, when

¹⁰Recall that, if H is a *complex* Hilbert space then positive semi-definiteness implies self-adjointness.

managing non-coherent operations, the convention in quantum computing is to use C^* -algebras as the state spaces. However, the extra structure that a C^* -algebra carries is not relevant to this thesis and, hence, no knowledge of C^* -algebras is required to follow it. For the purposes of this thesis, it suffices to think of a finite-dimensional C^* -algebra as a direct sum $\oplus_{i \in I} B(H_i)$.

Given a pair of finite-dimensional C^* -algebras A and B , the linear maps $A \rightarrow B$ that are physically meaningful are those that map states to states, *i.e.* density operators to density operators. These are the so-called *completely positive trace-preserving* (CPTP) maps:

- a positive map $A \rightarrow B$ sends positive semi-definite operators in A to positive semi-definite operators in B ;
- a positive map $f: A \rightarrow B$ is completely positive if for every finite-dimensional C^* -algebra C it is satisfied that $f \otimes \text{id}_C: A \otimes C \rightarrow B \otimes C$ is a positive map;
- a trace-preserving map $f: A \rightarrow B$ satisfies $\text{tr}(f(\rho)) = \text{tr}(\rho)$ for every $\rho \in A$.

A common example of a positive map that is not completely positive is the map $B(H) \rightarrow B(H)$ sending each operator to its matrix transpose.

Definition 1.5.7. The category **CPTP** has finite-dimensional C^* -algebras as objects and CPTP maps as morphisms. The triple $(\mathbf{CPTP}, \oplus, Z)$ is a symmetric monoidal category, where $\oplus$ is the direct sum of C^* -algebras and CPTP maps and Z is the zero-dimensional C^* -algebra (*i.e.* $Z = B(\{0\}) = \{0\}$).

Similarly to how the category of (weak) contractions will be useful to discuss the entries of ‘block matrices’ describing morphisms in **Isometry** and **Unitary**, it is useful to define a category of completely positive (weak) trace-reducing maps so that **CPTP** embeds in it.

Definition 1.5.8. The category **CPTR** has finite-dimensional C^* -algebras as objects and completely positive (weak) trace-reducing maps as morphisms; a map $f: A \rightarrow B$ is (weak) trace-reducing if it satisfies that $\text{tr}(f(\rho)) \leq \text{tr}(\rho)$ for all $\rho \in A$.¹¹ The triple $(\mathbf{CPTR}, \oplus, \{0\})$ is a symmetric monoidal category.

¹¹A *strict* trace-reducing map f satisfies $\text{tr}(f(\rho)) < \text{tr}(\rho)$ for all $\rho \in A$. In this thesis, the term ‘trace-reducing’ refers to weak trace-reducing maps unless stated otherwise.

Chapter 2

Categories for infinitary addition

Let $(X, +)$ be an arbitrary commutative monoid. By applying $+$ multiple times, we can add up any finite number of elements and, thanks to associativity and commutativity, the result will be unambiguous. However, standard monoids fall short if we are interested in adding up an infinite collection of elements. For instance, not every infinite sequence of real numbers is summable: to assign a result to a series we must verify that its sequence of partial sums converges. The first section of this chapter introduces Σ -monoids, a generalisation of monoids whose monoid operation is partially defined and aggregates (possibly infinite) collections of elements. The second section provides a brief introduction to topological monoids: these are monoids endowed with extra structure that lets us define a general notion of convergence and, consequently, formalise infinite sums. It will be shown that certain class of topological monoids — Hausdorff commutative monoids — provides an abundant number of examples of Σ -monoids.

The notion of *strong* Σ -monoid presented in this chapter was originally proposed by Haghverdi [28]; other flavours of Σ -monoids are proposed in this chapter, and their relation to each other is described in terms of adjunctions between their categories. Moreover, it is shown that the tensor product of Σ -monoids is well-defined, thus allowing us to define categories enriched over Σ -monoids; this will be essential in Chapter 3.

Definition 2.0.1. A *family* of elements in X is an indexed multiset $\{x_i \in X\}_{i \in I}$, where I is an arbitrary countable set.¹ Two families $\{x_i\}_{i \in I}$ and $\{x_j\}_{j \in J}$ are the same if there is a bijection $\phi: I \rightarrow J$ such that for all $i \in I$ it satisfies $x_i = x_{\phi(i)}$. The collection of all families of elements in X is denoted X^* .

¹Being an indexed multiset, there may be distinct indices $i, j \in I$ with $x_i = x_j$.

Proposition 2.0.2. *For any set X , the collection X^* is a (small) set.*

Proof. We can represent a multiset $\{x_i \in X\}_{i \in I}$ as a set $\{(x_i, i) \mid i \in I\} \subseteq X \times I$. Since the indexing set I of a family is countable by definition, there is an injection from X^* to the collection of all countable sets. The collection of all countable sets is small, hence, X^* is small. $\square$

Notation 2.0.3. The shorthand notation $\{x_i\}_I$ identifies indices with the lowercase of the letter denoting the indexing set. When the indexing set is not specified, all elements of the family are explicitly given within curly brackets, for instance, $\{x\}$ is the singleton family and $\{a, b, c\}$ is a finite family containing three elements. When explicit reference to the elements of the family is not necessary, a family may be denoted with a bold font variable $\mathbf{x} \in X^*$. A family $\{x_j\}_J$ is a *subfamily* of another $\{x_i\}_I$ if there is an injection $\phi: J \rightarrow I$ such that $x_j = x_{\phi(j)}$ for each $j \in J$. Hence, there is a partial order $\subseteq$ on X^* so that $\mathbf{x}' \subseteq \mathbf{x}$ iff $\mathbf{x}'$ is a subfamily of $\mathbf{x}$. Union, disjoint union and intersection of families are defined with respect to the corresponding operations on their indexed sets.² Let $f: X \rightarrow Y$ be a function and $\mathbf{x} = \{x_i\}_I$ a family in X^* ; the shorthand $f\mathbf{x}$ refers to the family $\{f(x_i)\}_I$ in Y^* .

Notation 2.0.4. For partial functions $f: A \rightharpoonup B$, $g: A' \rightharpoonup B$, the Kleene equality $f(a) \simeq g(a')$ indicates that $f(a)$ is defined if and only if $g(a')$ is defined and, when they both are, their results agree. In particular, $f(a) \simeq b$ implies that $f(a)$ is defined and is equal to b . The equals sign $f(a) = b$ is only used in the presence of partial functions when the context has already established that $f(a)$ is defined.

2.1 Σ -monoids

The notion of *strong* Σ -monoid that will be presented in this section was originally proposed by Haghverdi [28], although similar mathematical structures predate it: for instance, the partially additive monoids from Manes and Arbib [45] or the Σ -groups of Higgs [32]. Haghverdi [28] and Hoshino [33] have discussed how certain categories enriched in strong Σ -monoids may be used to capture iteration; this line of work is the main focus of Chapter 3. However, strong Σ -monoids are too restrictive for our application: Proposition 2.1.12 shows that these forbid the existence of additive inverses, which are essential in quantum computing since they capture destructive interference.

²Whenever an index i appears in both $\mathbf{x}$ and $\mathbf{y}$, their corresponding elements x_i and y_i must be equal for union and intersection to be well-defined.

In this section, a weaker version of Σ -monoids is proposed which admits additive inverses.

A Σ -monoid is a set X together with a partial function $\Sigma: X^* \rightarrow X$ describing an infinitary operation which satisfies certain axioms to ensure commutativity, associativity and existence of a neutral element.

Definition 2.1.1. Let X be a set and let $\Sigma: X^* \rightarrow X$ be a partial function. We say $\mathbf{x} \in X^*$ is a *summable family* if $\Sigma\mathbf{x}$ is defined. The pair (X, Σ) is a *weak Σ -monoid* if the following axioms are satisfied.

- *Singleton.* For all $x \in X$ the family $\{x\}$ is summable with $\Sigma\{x\} \simeq x$.
- *Neutral element.* (i) There is an element $0 \in X$ such that $\Sigma\emptyset \simeq 0$.
(ii) Let $\mathbf{x} \in X^*$ be a family and let $\mathbf{x}_\emptyset$ be its subfamily of elements other than 0; if $\mathbf{x}$ is summable then $\mathbf{x}_\emptyset$ is summable as well.
- *Bracketing.* Let $\{\mathbf{x}_j\}_J$ be an indexed set of *finite* summable families $\mathbf{x}_j \in X^*$ and let $\mathbf{x} = \uplus_J \mathbf{x}_j$; then:

$$\Sigma\mathbf{x} \simeq x \implies \Sigma\{\Sigma\mathbf{x}_j\}_J \simeq x.$$

- *Flattening.* Let $\{\mathbf{x}_j\}_J$ be an indexed set of summable families $\mathbf{x}_j \in X^*$ whose indexing set J is *finite* and let $\mathbf{x} = \uplus_J \mathbf{x}_j$; then:

$$\Sigma\{\Sigma\mathbf{x}_j\}_J \simeq x \implies \Sigma\mathbf{x} \simeq x.$$

In this definition, the axioms of commutative monoids have been lifted to the infinitary case. Bracketing and flattening together capture associativity and commutativity. Notice that these are not exactly dual to each other: bracketing assumes that each family $\mathbf{x}_j$ is finite, whereas flattening admits infinite families $\mathbf{x}_j$ but assumes there is only a finite number of them. As will be established in Proposition 2.1.12, the flattening axiom's requirement that J is finite is necessary for weak Σ -monoids to admit additive inverses. On the other hand, the bracketing axiom's requirement that each family $\mathbf{x}_j$ is finite is necessary so that every Hausdorff commutative monoid is a weak Σ -monoid whose summable families are precisely those that converge according to the topology (see Proposition 2.2.37). The following proposition establishes that $0 \in X$ acts as the neutral element of the Σ -monoid.

Proposition 2.1.2. *Let (X, Σ) be a weak Σ -monoid with $\Sigma\emptyset = 0$. Let $\mathbf{x} \in X^*$ be a family such that $\Sigma\mathbf{x} \simeq x$ for some $x \in X$. Let $\mathbf{0} \in X^*$ be a (possibly infinite) family whose elements are all 0; then $\Sigma(\mathbf{x} \uplus \mathbf{0}) \simeq x$. Conversely, let $\mathbf{x}_\emptyset \subseteq \mathbf{x}$ be the subfamily of elements other than 0; then $\Sigma\mathbf{x}_\emptyset \simeq x$.*

Proof. Since $\Sigma\emptyset = 0$, the family $\mathbf{0}$ may be equivalently defined as

$$\mathbf{0} = \{\Sigma\emptyset, \Sigma\emptyset, \dots\}$$

and, due to bracketing, $\Sigma\mathbf{0} \simeq 0$. Moreover, for any element $x \in X$, the singleton axiom imposes that $\Sigma\{x\} \simeq x$ and, clearly, $\{x\} = \{x\} \uplus \emptyset$ so according to bracketing:

$$\Sigma\{x\} \simeq x \implies \Sigma\{\Sigma\{x\}, \Sigma\emptyset\} \simeq x.$$

Using the singleton axiom and the definition of the neutral element, the right hand side can be rewritten as $\Sigma\{x, 0\} \simeq x$. Then, it follows that $\Sigma\{\Sigma\mathbf{x}, \Sigma\mathbf{0}\} \simeq x$ and the flattening axiom implies that $\Sigma(\mathbf{x} \uplus \mathbf{0}) \simeq x$, proving the first part of the claim.

On the other hand, we may partition any summable family $\mathbf{x} \in X^*$ into its subfamily $\mathbf{x}_\emptyset$ and its subfamily $\mathbf{0}$ so that $\mathbf{x} = \mathbf{x}_\emptyset \uplus \mathbf{0}$. Since the neutral element axiom imposes that $\mathbf{x}_\emptyset$ is summable, there is some $x' \in X$ such that $\Sigma\mathbf{x}_\emptyset \simeq x'$ and, since $\Sigma\mathbf{0} \simeq 0$ and $\Sigma\{x', 0\} \simeq x'$ have been established above, we may apply flattening to obtain $\Sigma\mathbf{x} \simeq x'$. But $\Sigma\mathbf{x} \simeq x$ by assumption, so it follows that $x = x'$ and $\Sigma\mathbf{x}_\emptyset \simeq x$, as claimed. $\square$

Example 2.1.3. Let $\pm$ be the set $\{0, +, -\}$ and, for each family $\mathbf{x} \in \pm^*$, let $n_+(\mathbf{x})$ (and $n_-(\mathbf{x})$) be the number of occurrences of $+$ (respectively, of $-$). Define a partial function $\Sigma: \pm^* \rightarrow \pm$ as follows:

$$\Sigma\mathbf{x} = \begin{cases} 0 & \text{if } n_+(\mathbf{x}) < \infty \text{ and } n_+(\mathbf{x}) = n_-(\mathbf{x}) \\ + & \text{if } n_+(\mathbf{x}) < \infty \text{ and } n_+(\mathbf{x}) = n_-(\mathbf{x}) + 1 \\ - & \text{if } n_+(\mathbf{x}) < \infty \text{ and } n_+(\mathbf{x}) = n_-(\mathbf{x}) - 1 \\ \text{undefined} & \text{otherwise.} \end{cases} \quad (2.1)$$

Then, $(\pm, \Sigma)$ is a weak Σ -monoid.

Proof. Both neutral element and singleton axioms are trivial to check; flattening is also straightforward: if each $\mathbf{x}_j$ is summable, then each $n_+(\mathbf{x}_j)$ is finite and

$$n_+(\uplus_J \mathbf{x}_j) = \sum_J n_+(\mathbf{x}_j)$$

holds, where the right hand side is a finite sum of natural numbers since J is finite. The same can be said of n_- , so the difference between n_+ and n_- stays the same after flattening. To see how bracketing is satisfied, recall that both the family $\mathbf{x}$ and each $\mathbf{x}_j$ are assumed to be summable. For all $j \in J$, the sum $\Sigma \mathbf{x}_j$ is either 0, + or -, so we may partition J into J_0 , J_+ and J_- depending on this result. Notice that J_+ and J_- must be finite sets, otherwise $\mathbf{x} = \uplus_J \mathbf{x}_j$ would have an infinite number of nonzero elements and would not be summable. Then, a simple counting argument determines that: if $\Sigma \mathbf{x} = 0$ then $|J_+| = |J_-|$, if $\Sigma \mathbf{x} = +$ then $|J_+| = |J_-| + 1$ and if $\Sigma \mathbf{x} = -$ then $|J_+| = |J_-| - 1$. Therefore, it follows that the family $\{\Sigma \mathbf{x}_j\}_J$ is summable, and the result agrees with that of $\Sigma \mathbf{x}$, thus satisfying the bracketing axiom. $\square$

Example 2.1.4. Let X be a set and let $P = \mathcal{P}(X)$ be its power set. For any $x \in X$, define $n_x: P^* \rightarrow \mathbb{N} \cup \{\infty\}$ as follows:

$$n_x(\{A_i\}_I) = |\{i \in I \mid x \in A_i\}|. \quad (2.2)$$

Define a partial function $\Sigma: P^* \rightarrow P$ as follows:

$$\Sigma\{A_i\}_I = \begin{cases} \{x \in X \mid n_x(\{A_i\}_I) \text{ odd}\} & \text{if } \forall x \in X, n_x(\{A_i\}_I) < \infty \\ \text{undefined} & \text{otherwise.} \end{cases} \quad (2.3)$$

Then, (P, Σ) is a weak Σ -monoid.

Proof. Notice that on finite families Σ corresponds to symmetric difference of sets. Singleton and neutral element axioms are trivial and flattening follows from associativity and commutativity of the symmetric difference of sets. Bracketing can be verified by realising that $n_x(\{A_i\}_I)$ is odd if and only if, for every partition $I = \uplus_J I_j$, there is an odd number of odd $n_x(\{A_i\}_{I_j})$. $\square$

Particularly relevant to this thesis is the fact that every Hausdorff commutative monoid is a weak Σ -monoid. This is established in Section 2.2 along with a brief introduction to general topology and Hausdorff commutative monoids. Many examples arise from this result; one of the simplest among them is given below.

Example 2.1.5. Let $(\mathbb{R}, +)$ be the standard Hausdorff commutative monoid of real numbers. According to Proposition 2.2.37, we may define a partial function $\Sigma: \mathbb{R}^* \rightarrow \mathbb{R}$ so that $(\mathbb{R}, \Sigma)$ is a weak Σ -monoid. Indeed, such Σ corresponds to the usual notion of absolute convergence of series.

In the same spirit that monoid homomorphisms are functions preserving the monoid structure, Σ -homomorphisms are functions that preserve summable families.

Definition 2.1.6. Let (X, Σ) and (Y, Σ') be weak Σ -monoids. A function $f: X \rightarrow Y$ is a Σ -homomorphism if for every family $\mathbf{x} \in X^*$ and every $x \in X$:

$$\Sigma \mathbf{x} \simeq x \implies \Sigma' f\mathbf{x} \simeq f(x).$$

Example 2.1.7. Let $(\pm, \Sigma)$ and $(\mathbb{R}, \Sigma)$ be the weak Σ -monoids from Examples 2.1.3 and 2.1.5 respectively. The function $f: \pm \rightarrow \mathbb{R}$ given by:

$$f(0) = 0 \quad f(+) = 1 \quad f(-) = -1$$

is a Σ -homomorphism.

Notice that in the weak Σ -monoid $(\pm, \Sigma)$ from Example 2.1.3 the family $\{+, +, -\}$ is summable whereas its subfamily $\{+, +\}$ is not. Similarly, we may define a weak Σ -monoid $([0, 1], \Sigma)$ whose Σ function is the same as in $(\mathbb{R}, \Sigma)$ when the result is in the interval $[0, 1]$ and it is undefined otherwise. Once again, $\{0.75, 0.5, -0.25\}$ is summable whereas its subfamily $\{0.75, 0.5\}$ is not. This idea of defining a weak Σ -monoid from an existing one by restricting the underlying set is generalised by the following lemma.

Lemma 2.1.8. Let (Y, Σ) be a weak Σ -monoid and let X be a set. Let $f: X \rightarrow Y$ be an injective function such that $0 \in \text{im}(f)$. Define a partial function $\Sigma^f: X^* \rightarrow X$ as follows:

$$\Sigma^f \mathbf{x} = \begin{cases} x & \text{if } \exists x \in X \text{ s.t. } \Sigma f\mathbf{x} \simeq f(x) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then, (X, Σ^f) is a weak Σ -monoid and $f: X \rightarrow Y$ is a Σ -homomorphism.

Proof. The definition of Σ^f is unambiguous thanks to the requirement that f is injective. Neutral element and singleton axioms in (X, Σ^f) are trivially derived from those in (Y, Σ) . Let $\{\mathbf{x}_j\}_J$ be an indexed set where each $\mathbf{x}_j \in X^*$ is a summable family in (X, Σ^f) and let $\mathbf{x} = \uplus_J \mathbf{x}_j$, evidently, $f\mathbf{x} = \uplus_J f\mathbf{x}_j$ (see Notation 2.0.3). Assume that each family $\mathbf{x}_j$ is finite, then the following sequence of implications proves bracketing:

$$\begin{aligned} \Sigma^f \mathbf{x} \simeq x &\iff \Sigma f\mathbf{x} \simeq f(x) && \text{(definition of } \Sigma^f) \\ &\implies \Sigma \{\Sigma f\mathbf{x}_j\}_J \simeq f(x) && \text{(bracketing in } Y) \\ &\iff \Sigma \{f(\Sigma^f \mathbf{x}_j)\}_J \simeq f(x) && (\mathbf{x}_j \text{ summable and def. of } \Sigma^f) \\ &\iff \Sigma^f \{\Sigma^f \mathbf{x}_j\}_J \simeq x && \text{(definition of } \Sigma^f). \end{aligned}$$

Finally, flattening of (X, Σ^f) can be proven following a similar argument, this time using flattening in (Y, Σ) . Thus, it has been shown that (X, Σ^f) is a weak Σ -monoid; to check that $f: X \rightarrow Y$ is a Σ -homomorphism realise that, by definition, $\mathbf{x}$ is summable if and only if $f\mathbf{x}$ is summable, so it is immediate that $\Sigma^f \mathbf{x} \simeq x$ implies $\Sigma f\mathbf{x} \simeq f(x)$. $\square$

The remainder of this section will study the category of weak Σ -monoids and certain important full subcategories of it, including that of the strong Σ -monoids discussed by Haghverdi [28].

Definition 2.1.9. Let Σ_w be the category whose objects are weak Σ -monoids and whose morphisms are Σ -homomorphisms.

Proposition 2.1.10. *The category Σ_w is complete.*

Proof. First, we show that Σ_w has all equalizers. Let X and Y be weak Σ -monoids and let $f, g: X \rightarrow Y$ be Σ -homomorphisms. Define the subset $E = \{x \in X \mid f(x) = g(x)\}$; the inclusion $e: E \rightarrow X$ is injective and $0 \in E$ due to $f(\Sigma\emptyset) = g(\Sigma\emptyset)$. Then, according to Lemma 2.1.8, E can be endowed with a weak Σ -monoid structure:

$$\Sigma^e \mathbf{x} = \begin{cases} x & \text{if } \exists x \in E \text{ s.t. } \Sigma e\mathbf{x} \simeq e(x) \\ \text{undefined} & \text{otherwise} \end{cases} \quad (2.4)$$

so that $e: E \rightarrow X$ is a Σ -homomorphism. Consequently, (E, Σ^e) is a cone in Σ_w and, since E is an equalizer in **Set**, for any other cone (A, Σ') with a Σ -homomorphism $h: A \rightarrow X$ there is a unique function $m: A \rightarrow E$ such that $h = e \circ m$. Therefore, to prove that (E, Σ^e) is an equalizer in Σ_w we only need to show that m is a Σ -homomorphism. For any summable family $\mathbf{a} \in A^*$ we have that $\Sigma h\mathbf{a} \simeq h(\Sigma'\mathbf{a})$ due to h being a Σ -homomorphism and since (A, Σ') is a cone $fh(\Sigma'\mathbf{a}) = gh(\Sigma'\mathbf{a})$ so $h(\Sigma'\mathbf{a}) \in E$ or, more precisely, $m(\Sigma'\mathbf{a}) \in E$. It then follows from $h = em$ that $\Sigma ema \simeq em(\Sigma'\mathbf{a})$ so, by definition of Σ^e , the family $m\mathbf{a}$ is summable in E and m is a Σ -homomorphism.

Next, we show that Σ_w has all small products. Let $\{(X_i, \Sigma^i)\}_I$ be a small collection of objects in Σ_w . Endow the set $\times_I X_i$ with the following partial function:

$$\Sigma^\times (\mathbf{x}_i)_{i \in I} = \begin{cases} (\Sigma^i \mathbf{x}_i)_{i \in I} & \text{if } \forall i \in I, \mathbf{x}_i \text{ is summable} \\ \text{undefined} & \text{otherwise.} \end{cases} \quad (2.5)$$

Then, $(\times_I X_i, \Sigma^\times)$ is a weak Σ -monoid; the proof is straightforward. It is immediate that projections are Σ -homomorphisms, so this is a cone for the discrete diagram $\{(X_i, \Sigma^i)\}_I$. Moreover, $\times_I X_i$ is the categorical product in **Set** so, for any other cone (A, Σ') with Σ -homomorphisms $h_i: A \rightarrow X_i$, there is a unique function $m: A \rightarrow \times_I X_i$ so that each h_i factors through m . To prove that $(\times_I X_i, \Sigma^\times)$ is a categorical product in Σ_w it only remains to show that this unique function m is a Σ -homomorphism. If $\mathbf{a}$ is summable in A , then each $h_i \mathbf{a}$ is summable in X_i and, by definition, $\Sigma^\times(h_i \mathbf{a})_{i \in I}$ is summable in $\times_I X_i$. Each h_i is equal to $\pi_i \circ m$ where π_i is the corresponding projection, so it is immediate that $\Sigma^\times m \mathbf{a} \simeq m(\Sigma' \mathbf{a})$ and m is a Σ -homomorphism.

A category with all equalizers and small products is complete, so Σ_w is complete as claimed. $\square$

2.1.1 Important full subcategories of Σ_w

Different flavours of Σ -monoids are introduced in this subsection. The differences between them are with respect to the partiality of Σ , *i.e.* new axioms are introduced that require more families to be summable. The first to be introduced is the notion of Σ -monoid proposed by Haghverdi [28].

Definition 2.1.11. A *strong Σ -monoid* is a weak Σ -monoid (X, Σ) that satisfies the following extra axioms.

- *Subsummability.* Let $\{x_i\}_I$ be a summable family; for every subset $J \subset I$ the subfamily $\{x_j\}_J$ is also summable.
- *Strong bracketing.* Let $\{\mathbf{x}_j\}_J$ be an indexed set where each $\mathbf{x}_j \in X^*$ is a summable family and let $\mathbf{x} = \uplus_J \mathbf{x}_j$; then:

$$\Sigma \mathbf{x} \simeq x \implies \Sigma\{\Sigma \mathbf{x}_j\}_J \simeq x.$$

- *Strong flattening.* Let $\{\mathbf{x}_j\}_J$ be an indexed set where each $\mathbf{x}_j \in X^*$ is a summable family and let $\mathbf{x} = \uplus_J \mathbf{x}_j$; then:

$$\Sigma\{\Sigma \mathbf{x}_j\}_J \simeq x \implies \Sigma \mathbf{x} \simeq x.$$

Notice that the preconditions of finiteness required in the standard bracketing and flattening axioms are removed, thus making the axioms stronger. It is straightforward

to check that this definition is equivalent to that of Haghverdi [28]. In Haghverdi's definition, strong bracketing, strong flattening and subsummability are condensed into a single axiom, and the neutral element axiom of weak Σ -monoids is omitted since it follows immediately from subsummability — notice that the empty family is a trivial subfamily. Importantly, the following proposition establishes that strong flattening forbids the existence of inverse elements.

Proposition 2.1.12 (Haghverdi [28]). *Let (X, Σ) be a strong Σ -monoid. If a family $\mathbf{x} \in X^*$ satisfies $\Sigma \mathbf{x} \simeq 0$, then every element $a \in \mathbf{x}$ is the neutral element $a = 0$.*

Proof. Let $\mathbf{x}$ be a family such that $\Sigma \mathbf{x} \simeq 0$ and choose an arbitrary element $a \in \mathbf{x}$. Define the following infinite families:

$$\begin{aligned}\mathbf{z} &= \mathbf{x} \uplus \mathbf{x} \uplus \dots \\ \mathbf{a} &= \{a\} \uplus \mathbf{z}\end{aligned}$$

Due to strong flattening, neutral element and singleton axioms:

$$\begin{aligned}\Sigma \mathbf{z} &\simeq \Sigma\{\Sigma \mathbf{x}, \Sigma \mathbf{x}, \dots\} \simeq \Sigma\{0, 0, \dots\} \simeq 0 \\ \Sigma \mathbf{a} &\simeq \Sigma\{\Sigma\{a\}, \Sigma \mathbf{z}\} \simeq \Sigma\{a, 0\} \simeq a\end{aligned}$$

However, both families $\mathbf{z}$ and $\mathbf{a}$ contain the same elements: a countably infinite number of copies of each element in $\mathbf{x}$. Being the same family, it is immediate that $\Sigma \mathbf{a} \simeq \Sigma \mathbf{z}$ which implies $a = 0$. This argument holds for each element $a \in \mathbf{x}$, proving the claim. $\square$

Since weak Σ -monoids do admit inverses, it is relevant to consider the notion of Σ -groups. It is straightforward to check that the axioms of Σ -groups defined below are equivalent to those proposed by Higgs [32]. However, Higgs' Σ -groups are more general since Higgs' notion of families are not restricted to be countable, unlike ours (see Definition 2.0.1).

Definition 2.1.13. A *finitely total Σ -monoid* is a weak Σ -monoid where every finite family is summable. A *Σ -group* is a finitely total Σ -monoid where, for every $x \in X$, there is an element $-x \in X$ satisfying $\Sigma\{x, -x\} = 0$ and where the function that maps each $x \in X$ to $-x$ is a Σ -homomorphism.

Remark 2.1.14. Notice that every finitely total Σ -monoid (X, Σ) is trivially a commutative monoid $(X, +, 0)$ where $x + y$ is defined to be $\Sigma\{x, y\}$ for every $x, y \in X$ and $0 = \Sigma \emptyset$; both associativity and commutativity follow from the combination of

bracketing and flattening. Similarly, every Σ -group is trivially an abelian group and, hence, the inverse of an element in a Σ -group is unique. Moreover, thanks to the inverse mapping being a Σ -homomorphism, any summable family $\mathbf{x} \in X$ (even infinite ones) have an ‘inverse’ $-\mathbf{x}$ obtained by applying the inverse mapping to each of its elements so that:

$$0 = \Sigma\{\Sigma\mathbf{x}, -(\Sigma\mathbf{x})\} = \Sigma\{\Sigma\mathbf{x}, \Sigma-\mathbf{x}\}.$$

For each of these flavours of Σ -monoids we may consider a full subcategory of Σ_w as defined below.

Definition 2.1.15. Let Σ_s , Σ_{ft} and Σ_g be the full subcategories of Σ_w obtained by restricting the class of objects to strong Σ -monoids, finitely total Σ -monoids and Σ -groups, respectively. For brevity, these subcategories together with Σ_w are referred to as Σ_* categories.

Since strong Σ -monoids do not admit inverses (see Proposition 2.1.12) it is clear that Σ -groups and strong Σ -monoids are disjoint subclasses of weak Σ -monoids. The hierarchy of inclusions between Σ_* categories can be summarised by the existence of the following full and faithful embedding functors:

$$\Sigma_s \hookrightarrow \Sigma_w \qquad \Sigma_g \hookrightarrow \Sigma_{ft} \hookrightarrow \Sigma_w. \quad (2.6)$$

Proposition 2.1.16. *All of the Σ_* categories are complete. All of the embedding functors from (2.6) create limits.*

Proof. Proposition 2.1.10 already established the claim for Σ_w . Whenever the objects in the diagram are in Σ_s , it is straightforward to check that the construction from Proposition 2.1.10 — both for equalizers and small products — yields a strong Σ -monoid and, hence, provides a cone in Σ_s . The inclusion functor $\Sigma_s \hookrightarrow \Sigma_w$ reflects limits, by virtue of being full and faithful; therefore, Σ_s is complete. The same argument applies to Σ_{ft} and Σ_g so they are all complete, as claimed. Since all Σ_* categories are complete and the embedding functors reflect limits, it is immediate that they also preserve them and, hence, all of the embedding functors from (2.6) create limits. $\square$

The following subsections establish that Σ_s , Σ_{ft} and Σ_g are not only full subcategories of Σ_w but reflective ones — *i.e.* there is a left adjoint to each of the embedding functors in (2.6).

2.1.2 Σ_s is a reflective subcategory of Σ_w

The left adjoint to the canonical embedding $\Sigma_s \hookrightarrow \Sigma_w$ is explicitly constructed in Proposition 2.1.18. The key insight is that the strong versions of bracketing and flattening can be recovered by defining a congruence $\sim$ and quotienting the set of all families with respect to it. Then, we ought to find the ‘smallest’ strong Σ -monoid defined on this quotient set that satisfies the universal property of the adjunction. The following lemma is instrumental to find such a ‘smallest’ strong Σ -monoid.

Lemma 2.1.17. *Let $\{(X_i, \Sigma^i)\}_I$ be a collection of strong Σ -monoids. Define a partial function $\Sigma^\cap: (\cap_I X_i)^* \rightarrow \cap_I X_i$ as follows:*

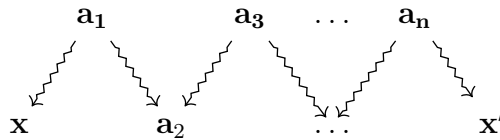
$$\Sigma^\cap_{\mathbf{x}} = \begin{cases} x & \text{if } \forall i \in I, \Sigma^i_{\mathbf{x}} \simeq x \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then, $(\cap_I X_i, \Sigma^\cap)$ is a strong Σ -monoid.

Proof. Whenever a family $\mathbf{x}$ is summable in $\Sigma^\cap$, it must be summable in Σ^i for all $i \in I$ and, considering that each (X_i, Σ^i) is a strong Σ -monoid, it follows that every subfamily of $\mathbf{x}$ is summable in every Σ^i and, hence, it is summable in $\Sigma^\cap$ so that subsummability is satisfied. Similarly, strong bracketing and strong flattening in $\Sigma^\cap$ follow from those in each Σ^i and the neutral element and singleton axioms are trivially satisfied. Consequently, $(\cap_I X_i, \Sigma^\cap)$ is a strong Σ -monoid. $\square$

Proposition 2.1.18. *There is a left adjoint functor to the embedding $\Sigma_s \hookrightarrow \Sigma_w$.*

Proof. Let $G: \Sigma_s \hookrightarrow \Sigma_w$ denote the canonical embedding; its left adjoint $F: \Sigma_w \rightarrow \Sigma_s$ is defined explicitly. Let $X \in \Sigma_w$ be an arbitrary weak Σ -monoid and define a relation $\rightsquigarrow$ on X^* such that for any two families $\mathbf{x}, \mathbf{x}' \in X^*$, we have that $\mathbf{x} \rightsquigarrow \mathbf{x}'$ iff there is a partition of $\mathbf{x}$ into subfamilies $\mathbf{x} = \uplus_J \mathbf{x}_j$ such that each (possibly infinite) subfamily $\mathbf{x}_j$ is summable and such that $\mathbf{x}'$ is the family of their sums $\mathbf{x}' = \{\Sigma \mathbf{x}_j\}_J$. The relation $\rightsquigarrow$ is reflexive since any family may be partitioned into its singleton subfamilies. Let $\sim$ be the equivalence closure of $\rightsquigarrow$; then, $\mathbf{x} \sim \mathbf{x}'$ iff there is a zig-zag chain of $\rightsquigarrow$ relations such as:



Notice that $\sim$ is a congruence with respect to arbitrary disjoint union, *i.e.* it satisfies:

$$\forall j \in J, \mathbf{x}_j \sim \mathbf{x}'_j \implies \uplus_J \mathbf{x}_j \sim \uplus_J \mathbf{x}'_j \quad (2.7)$$

since a zig-zag chain from $\uplus_J \mathbf{x}_j$ to $\uplus_J \mathbf{x}'_j$ may be obtained by composing the chains relating each subfamily $\mathbf{x}_j$ to $\mathbf{x}'_j$. Let A be the quotient set $X^*/\sim$ and let $q: X \rightarrow A$ be the function that maps each $x \in X$ to the equivalence class $[\{x\}] \in A$.

Let $Y \in \Sigma_s$ be an arbitrary strong Σ -monoid and let $f: X \rightarrow G(Y)$ be an arbitrary Σ -homomorphism. Define $\rightsquigarrow$ on Y^* as above; since f is a Σ -homomorphism we have that $\mathbf{x} \rightsquigarrow \mathbf{x}'$ implies $f\mathbf{x} \rightsquigarrow f\mathbf{x}'$ and, consequently,

$$\mathbf{x} \sim \mathbf{x}' \implies f\mathbf{x} \sim f\mathbf{x}'. \quad (2.8)$$

Moreover, in a strong Σ -monoid $\mathbf{y} \rightsquigarrow \mathbf{y}'$ implies $\Sigma\mathbf{y} \simeq \Sigma\mathbf{y}'$, since if $\mathbf{y}$ is summable then $\mathbf{y}'$ is summable due to strong bracketing whereas if $\mathbf{y}'$ is summable then $\mathbf{y}$ is summable due to strong flattening. Consequently, we have that for any two families $\mathbf{x}, \mathbf{x}' \in X^*$:

$$\mathbf{x} \sim \mathbf{x}' \implies \Sigma f\mathbf{x} \simeq \Sigma f\mathbf{x}'. \quad (2.9)$$

Let A_f be the subset of A defined as follows:

$$A_f = \{[\mathbf{x}] \in A \mid f\mathbf{x} \text{ is summable in } Y\}.$$

Notice that, for any equivalence class $[\mathbf{x}] \in A_f$, each of its families $\mathbf{x}' \in [\mathbf{x}]$ satisfies that $f\mathbf{x}'$ is summable, due to the implication (2.9) and the fact that $f\mathbf{x}$ is summable. Let $\Sigma^f: A_f^* \rightarrow A_f$ be the following partial function:

$$\Sigma^f \{[\mathbf{x}_i]\}_I = \begin{cases} [\uplus_I \mathbf{x}_i] & \text{if } [\uplus_I \mathbf{x}_i] \in A_f \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Notice that this definition is independent of the choice of representatives thanks to $\sim$ being a congruence (2.7). We must check that (A_f, Σ^f) is a strong Σ -monoid.

- *Singleton.* Singleton families are trivially summable.
- *Subsummability.* If $\{[\mathbf{x}_i]\}_I \in A_f^*$ is summable then $\uplus_I f\mathbf{x}_i$ is summable in Y and so is $\uplus_J f\mathbf{x}_j$ for any $J \subseteq I$ due to subsummability in Y . Thus, $\{[\mathbf{x}_j]\}_J$ is summable in A_f and Σ^f satisfies subsummability.
- *Neutral element.* Follows immediately from subsummability.
- *Strong flattening.* Let I be an arbitrary set partitioned into $I = \uplus_J I_j$ and, for each $j \in J$, let $\{[\mathbf{x}_i]\}_{I_j}$ be a summable family in A_f , i.e. $\Sigma^f \{[\mathbf{x}_i]\}_{I_j} \simeq [\uplus_{I_j} \mathbf{x}_i]$. Assume that the family $\{[\uplus_{I_j} \mathbf{x}_i]\}_J \in A_f^*$ is summable, i.e. $\Sigma^f \{[\uplus_{I_j} \mathbf{x}_i]\}_J \simeq [\uplus_J (\uplus_{I_j} \mathbf{x}_i)]$; then, due to associativity and commutativity of $\uplus$ we have that $\uplus_I \mathbf{x}_i = \uplus_J (\uplus_{I_j} \mathbf{x}_i)$ and, hence, $[\uplus_I \mathbf{x}_i] \in A_f$ so that $\{[\mathbf{x}_i]\}_I$ is summable in A_f .

- *Strong bracketing.* As in the case of flattening, this axiom follows from associativity and commutativity of disjoint union.

Let S be the set of all such strong Σ -monoids (A_f, Σ^f) .³ Let $F(X)$ be the strong Σ -monoid obtained as the intersection of all the members of S as per Lemma 2.1.17. Let $\eta_X: X \rightarrow GF(X)$ be the function that maps each $x \in X$ to $\{x\}$; such an equivalence class is present in every A_f , so it is present in $F(X)$. Notice that η_X is a Σ -homomorphism: if $\mathbf{x} = \{x_i\}_I \in X^*$ is summable, then $\mathbf{x} \rightsquigarrow \{\Sigma \mathbf{x}\}$ and $[\mathbf{x}] = [\{\Sigma \mathbf{x}\}]$, hence,

$$\Sigma^\cap \{\eta_X(x_i)\}_I \simeq [\uplus_I \{x_i\}] = [\mathbf{x}] = [\{\Sigma \mathbf{x}\}] = \eta_X(\Sigma \mathbf{x}).$$

On morphisms $f: X \rightarrow X'$, let $F(f)$ be the function that maps $[\mathbf{x}] \in F(X)$ to $[f\mathbf{x}] \in F(X')$; this is a Σ -homomorphism thanks to (2.8). It is straightforward to check that this construction yields a functor $F: \Sigma_w \rightarrow \Sigma_s$ and a natural transformation η whose components η_X were defined above.

It remains to check that F is left adjoint to G . Fix some arbitrary $X \in \Sigma_w, Y \in \Sigma_s$ and $f \in \Sigma_w(X, G(Y))$. There is a unique Σ -homomorphism $\bar{f}: F(X) \rightarrow Y$ making the diagram

$$\begin{array}{ccc} X & \xrightarrow{f} & G(Y) \\ & \searrow \eta_X & \uparrow G(\bar{f}) \\ & & GF(X) \end{array}$$

commute: the requirement that $f = G(\bar{f}) \circ \eta_X$ imposes that $\bar{f}$ maps each $\eta_X(x) = \{x\}$ to $f(x)$; the requirement that $\bar{f}$ is a Σ -homomorphism imposes that, for any arbitrary equivalence class $[\{x_i\}_I] \in F(X)$,

$$\bar{f}[\{x_i\}_I] = \bar{f}[\uplus_I \{x_i\}] = \bar{f}(\Sigma^\cap \{\eta_X(x_i)\}_I) = \Sigma \{f(x_i)\}_I.$$

The value of such a Σ -homomorphism $\bar{f}$ is uniquely determined and, thus, it has been shown that F is left adjoint to G , as claimed. $\square$

Since Σ_s is a full subcategory of Σ_w it follows that Σ_s is a reflective subcategory of Σ_w . Let X be a weak Σ -monoid containing a pair of elements $a, b \in X$ such that $\Sigma\{a, b\} \simeq 0$. The family $\mathbf{x} = \{a\}_\mathbb{N} \uplus \{b\}_\mathbb{N}$ that contains an infinite number of copies of a and b satisfies that $\mathbf{x} \rightsquigarrow \{a\}$, $\mathbf{x} \rightsquigarrow \{0\}$ and $\mathbf{x} \rightsquigarrow \{b\}$ depending on how we partition it. Then, $[\{a\}] = [\{0\}] = [\{b\}]$, implying η_X is not injective and, in fact, every element

³Even though $f: X \rightarrow G(Y)$ ranges over all $Y \in \Sigma_s$, each A_f is by definition a subset of A which is small since X^* is small (see Proposition 2.0.2), hence, S is a (small) set.

of X that has an additive inverse is mapped to the neutral element $[\{0\}]$ of $F(X)$. This is to be expected since, according to Proposition 2.1.12, strong Σ -monoids cannot have additive inverses.

2.1.3 Σ_{ft} is a reflective subcategory of Σ_{w}

The canonical embedding $\Sigma_{\text{ft}} \hookrightarrow \Sigma_{\text{w}}$ has a left adjoint. In this case, an explicit construction of the free finitely total Σ -monoid has not been achieved; instead, the proof uses the general adjoint functor theorem.

Proposition 2.1.19. *There is a left adjoint functor to the embedding $\Sigma_{\text{ft}} \hookrightarrow \Sigma_{\text{w}}$.*

Proof. Let $G: \Sigma_{\text{ft}} \hookrightarrow \Sigma_{\text{w}}$ denote the canonical embedding. It is immediate that Σ_{ft} is locally small since there is a faithful forgetful functor $\Sigma_{\text{ft}} \rightarrow \mathbf{Set}$. According to Proposition 2.1.16, the category Σ_{ft} is complete and the embedding functor G preserves limits. It remains to show that, for any $X \in \Sigma_{\text{w}}$, the comma category $(X \Rightarrow G)$ has a weakly initial set; then, the existence of a left adjoint will follow from the general adjoint functor theorem (see Theorem 1.4.36). Let $Y \in \Sigma_{\text{ft}}$ be an arbitrary finitely total Σ -monoid and let $f: X \rightarrow G(Y)$ be an arbitrary Σ -homomorphism. Let Z be the following subset of X^* :

$$Z = \{\mathbf{x} \in X^* \mid f\mathbf{x} \text{ is summable in } Y\}$$

and let $\sim$ be the equivalence relation on Z where:

$$\mathbf{x} \sim \mathbf{x}' \iff \Sigma f\mathbf{x} = \Sigma f\mathbf{x}'.$$

Let A_f be the quotient set $Z/\sim$ and let $\bar{f}: A_f \rightarrow Y$ be the injective function that maps each $[\mathbf{x}] \in A_f$ to $\Sigma f\mathbf{x}$. According to Lemma 2.1.8, there is a partial function $\Sigma^{\bar{f}}: A_f^* \rightharpoonup A_f$ induced by $\bar{f}$:

$$\Sigma^{\bar{f}}\{[\mathbf{x}_j]\}_J = \begin{cases} [\mathbf{x}] & \text{if } \exists [\mathbf{x}] \in A_f \text{ s.t. } \Sigma\{\bar{f}[\mathbf{x}_j]\}_J \simeq \bar{f}[\mathbf{x}] \\ \text{undefined} & \text{otherwise.} \end{cases}$$

which, by virtue of $\bar{f}$ being injective, satisfies that $(A_f, \Sigma^{\bar{f}})$ is a weak Σ -monoid. Using the definition of $\bar{f}$ we obtain the following equivalent definition of $\Sigma^{\bar{f}}$:

$$\Sigma^{\bar{f}}\{[\mathbf{x}_j]\}_J = \begin{cases} [\mathbf{x}] & \text{if } \exists \mathbf{x} \in Z \text{ s.t. } \Sigma\{\Sigma f\mathbf{x}_j\}_J \simeq \Sigma f\mathbf{x} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The definition is independent of the choice of representatives since Z is comprised of families whose image under f is summable and $\sim$ relates those that sum up to the same value in Y . Recall that each $\mathbf{x}_j$ is in Z by definition and, consequently, each $f\mathbf{x}_j$ is summable; then, when J is a finite set the family $\{\Sigma f\mathbf{x}_j\}_J$ is summable thanks to Y being finitely total and $\Sigma\{\Sigma f\mathbf{x}_j\}_J \simeq \Sigma(\oplus_J f\mathbf{x}_j)$ due to flattening. Therefore, whenever J is finite we have that:

$$\Sigma^{\bar{f}}\{[\mathbf{x}_j]\}_J \simeq [\oplus_J \mathbf{x}_j]$$

implying that every finite family is summable in $\Sigma^{\bar{f}}$ so that $(A_f, \Sigma^{\bar{f}}) \in \Sigma_{\text{ft}}$.

Let $q: X \rightarrow G(A_f)$ be the function that maps each $x \in X$ to $[\{x\}]$. Assume that $\mathbf{x} = \{x_i\}_I \in X^*$ is summable, then $f\mathbf{x}$ is summable in Y due to f being a Σ -homomorphism; thus, $\mathbf{x} \in Z$ and q is a Σ -homomorphism:

$$\begin{aligned} q(\Sigma \mathbf{x}) &= [\{\Sigma \mathbf{x}\}] && (\text{def. of } q) \\ &= [\mathbf{x}] && (f(\Sigma \mathbf{x}) = \Sigma f\mathbf{x}) \\ &\simeq \Sigma^{\bar{f}}\{[\{x_i\}]\}_I && (\Sigma f\mathbf{x} = \Sigma\{\Sigma\{f(x_i)\}\}_I) \\ &= \Sigma^{\bar{f}}\{q(x_i)\}_I. && (\text{def. of } q) \end{aligned}$$

According to Lemma 2.1.8, the function $\bar{f}: A_f \rightarrow Y$ is also a Σ -homomorphism and it is straightforward to check that $f = G(\bar{f}) \circ q$. Thus, it has been shown that for every object (f, Y) in the comma category $(X \Rightarrow G)$ there is an object (q, A_f) such that a morphism $\bar{f}: (q, A_f) \rightarrow (f, Y)$ exists. Let S be the collection of all such (q, A_f) objects; notice that S is a (small) set since each A_f is the quotient of a subset of X^* (which is itself small, Proposition 2.0.2), so the collection of all such A_f is small and, for each of them, there is only a small collection of partial functions $A_f^* \rightarrow A_f$. Consequently, S is a weakly initial set in the comma category $(X \Rightarrow G)$ and the claim that G has a left adjoint follows from the general adjoint functor theorem (see Theorem 1.4.36). $\square$

Since Σ_{ft} is a full subcategory of Σ_{w} it follows that Σ_{ft} is a reflective subcategory of Σ_{w} . The proof given above follows the same strategy that Hoshino [33] used to show that the category of totally defined strong Σ -monoids is a reflective subcategory of Σ_{s} . It is reasonable to think that a constructive proof may be achieved using the following strategy: given any weak Σ -monoid (X, Σ) , let $(X, +)$ be the partially commutative monoid obtained by defining $x + x' \simeq \Sigma\{x, x'\}$ and let $(Y, +')$ be its free commutative monoid constructed via the left adjoint to $\mathbf{CMon} \hookrightarrow \mathbf{PCM}$ (see [34]); then, we

may attempt to extend $(Y, +')$ to a Σ -monoid where the only infinite families that are summable are those required for $\eta_X: (X, \Sigma) \rightarrow (Y, \Sigma')$ to be a Σ -homomorphism, plus the ones imposed by bracketing and flattening. Such a strategy was attempted but, unfortunately, it was not clear whether quotienting the underlying set Y would be necessary to guarantee that different bracketings of the same infinite family would agree in their sum and, if quotienting is required, verifying that the definition of Σ' is independent of the choice of representatives becomes exceedingly subtle and it was not pursued further.

2.1.4 Σ_g is a reflective subcategory of Σ_{ft}

The canonical embedding $\Sigma_g \hookrightarrow \Sigma_{ft}$ has a left adjoint. An explicit construction of the free Σ -group has not been achieved; instead, the proof uses the general adjoint functor theorem. The proof is a combination of the Grothendieck group construction (Example 1.4.32) and the proof of Proposition 2.1.19 from the previous subsection.

Proposition 2.1.20. *There is a left adjoint functor to the embedding $\Sigma_g \hookrightarrow \Sigma_{ft}$.*

Proof. Let $G: \Sigma_g \hookrightarrow \Sigma_{ft}$ denote the canonical embedding. It is immediate that Σ_g is locally small, since there is a faithful forgetful functor $\Sigma_g \rightarrow \mathbf{Set}$. According to Proposition 2.1.16, the category Σ_g is complete and the embedding functor G preserves limits. It remains to show that, for any $X \in \Sigma_{ft}$, the comma category $(X \Rightarrow G)$ has a weakly initial set; then, the existence of a left adjoint will follow from the general adjoint functor theorem (see Theorem 1.4.36). Let $Y \in \Sigma_g$ be an arbitrary Σ -group and let $f: X \rightarrow G(Y)$ be an arbitrary Σ -homomorphism. Let Z be the following subset of $X^* \times X^*$:

$$Z = \{(\mathbf{p}, \mathbf{n}) \in X^* \times X^* \mid f\mathbf{p} \text{ and } f\mathbf{n} \text{ are summable in } Y\}$$

and let $\sim$ be the equivalence relation on Z where:

$$(\mathbf{p}, \mathbf{n}) \sim (\mathbf{p}', \mathbf{n}') \iff \Sigma\{\Sigma f\mathbf{p}, -(\Sigma f\mathbf{n})\} = \Sigma\{\Sigma f\mathbf{p}', -(\Sigma f\mathbf{n}')\}.$$

Let A_f be the quotient set $Z/\sim$ and let $\bar{f}: A_f \rightarrow Y$ be the injective function defined as follows for every equivalence class $[(\mathbf{p}, \mathbf{n})] \in A_f$:

$$\bar{f}[(\mathbf{p}, \mathbf{n})] = \Sigma\{\Sigma f\mathbf{p}, -\Sigma f\mathbf{n}\}.$$

According to Lemma 2.1.8, there is a partial function $\Sigma^{\bar{f}}: A_f^* \rightarrow A_f$ induced by $\bar{f}$:

$$\Sigma^{\bar{f}}\{[(\mathbf{p}_j, \mathbf{n}_j)]\}_J = \begin{cases} [(\mathbf{p}, \mathbf{n})] & \text{if } \exists(\mathbf{p}, \mathbf{n}) \in Z \text{ s.t. } \Sigma\{\bar{f}[(\mathbf{p}_j, \mathbf{n}_j)]\}_J \simeq \bar{f}[(\mathbf{p}, \mathbf{n})] \\ \text{undefined} & \text{otherwise.} \end{cases}$$

which, by virtue of $\bar{f}$ being injective, satisfies that $(A_f, \Sigma^{\bar{f}})$ is a weak Σ -monoid. Next, we show that every family $\{(\mathbf{p}_j, \mathbf{n}_j) \in A_f\}_J$ such that J is finite is a summable family. Notice that, by definition of A_f , if $(\mathbf{p}_j, \mathbf{n}_j) \in A_f$ for all $j \in J$ then $f\mathbf{p}_j$ and $f\mathbf{n}_j$ are summable in Y and, due to J being finite and Y finitely total, $\{f\mathbf{p}_j\}_J$ and $\{f\mathbf{n}_j\}_J$ are summable as well. It follows that:

$$\begin{aligned} \Sigma\{\bar{f}[(\mathbf{p}_j, \mathbf{n}_j)]\}_J &= \Sigma\{\Sigma\{\Sigma f\mathbf{p}_j, -\Sigma f\mathbf{n}_j\}\}_J && (\text{def. } \bar{f}) \\ &\simeq \Sigma\{\Sigma\{\Sigma f\mathbf{p}_j\}_J, \Sigma\{-\Sigma f\mathbf{n}_j\}_J\} && (\text{flattening, then bracketing}) \\ &\simeq \Sigma\{\Sigma(\uplus_J f\mathbf{p}_j), -\Sigma(\uplus_J f\mathbf{n}_j)\} && (\text{inversion } \Sigma\text{-hom. and flattening}) \\ &= \bar{f}[(\uplus_J \mathbf{p}_j, \uplus_J \mathbf{n}_j)] && (\text{def. } \bar{f}) \end{aligned}$$

where the first $\simeq$ follows from the fact that J is finite so that $\Sigma\{\Sigma\{a_j, -b_j\}\}_J$ can be first flattened into a sum of a finite family and then bracketed into $\Sigma\{\Sigma\{a_j\}_J, \Sigma\{-b_j\}_J\}$. According to the definition of $\Sigma^{\bar{f}}$, the above implies that

$$\Sigma^{\bar{f}}\{[(\mathbf{p}_j, \mathbf{n}_j)]\}_J \simeq [(\uplus_J \mathbf{p}_j, \uplus_J \mathbf{n}_j)]$$

so that every finite family in A_f^* is summable. Finally, for every element $[(\mathbf{p}, \mathbf{n})] \in A_f$ the element $[(\mathbf{n}, \mathbf{p})] \in A_f$ acts as its inverse; to check this, notice that the neutral element of A_f is $[(\emptyset, \emptyset)]$ which satisfies $\bar{f}[(\emptyset, \emptyset)] = 0$ and let $a = \Sigma f\mathbf{p}$ and $b = \Sigma f\mathbf{n}$ so that:

$$\begin{aligned} \Sigma\{\bar{f}[(\mathbf{p}, \mathbf{n})], \bar{f}[(\mathbf{n}, \mathbf{p})]\} &= \Sigma\{\Sigma\{a, -b\}, \Sigma\{b, -a\}\} && (\text{def. of } \bar{f}) \\ &= \Sigma\{a, -b, b, -a\} && (\text{flattening in } Y) \\ &= \Sigma\{\Sigma\{a, -a\}, \Sigma\{b, -b\}\} && (\text{bracketing in } Y) \\ &= \Sigma\{0, 0\} = 0 && (\text{inverses in } Y) \end{aligned}$$

thus, $\Sigma^{\bar{f}}\{[(\mathbf{p}, \mathbf{n})], [(\mathbf{n}, \mathbf{p})]\} = [(\emptyset, \emptyset)]$. It is straightforward to check that:

$$\Sigma^{\bar{f}}\{[(\mathbf{p}_j, \mathbf{n}_j)]\}_J \simeq [(\mathbf{p}, \mathbf{n})] \iff \Sigma^{\bar{f}}\{[(\mathbf{n}_j, \mathbf{p}_j)]\}_J \simeq [(\mathbf{n}, \mathbf{p})]$$

so that the inverse mapping is a Σ -homomorphism. Thus, it has been established that $(A_f, \Sigma^{\bar{f}})$ is a Σ -group.

Let $q: X \rightarrow G(A_f)$ be the function that maps each $x \in X$ to $[(\{x\}, \emptyset)]$. It is straightforward to check that q is a Σ -homomorphism since for every summable family $\mathbf{x} = \{x_i\}_I \in X^*$ it follows that:

$$q(\Sigma \mathbf{x}) = [(\{\Sigma \mathbf{x}\}, \emptyset)] = [(\mathbf{x}, \emptyset)] \simeq \Sigma^{\bar{f}}\{q(x_i)\}_I.$$

According to Lemma 2.1.8, the function $\bar{f}: A_f \rightarrow Y$ is also a Σ -homomorphism and it is straightforward to check that $f = G(\bar{f}) \circ q$. Thus, it has been shown that for every object (f, Y) in the comma category $(X \Rightarrow G)$ there is an object (q, A_f) such that a morphism $\bar{f}: (q, A_f) \rightarrow (f, Y)$ exists. Let S be the collection of all such (q, A_f) objects; notice that S is a (small) set since each A_f is a quotient of a subset of X^* (which is itself small, Proposition 2.0.2), so the collection of all such A_f is small and, for each of them, there is only a small collection of partial functions $A_f^* \rightarrow A_f$. Consequently, S is a weakly initial set in the comma category $(X \Rightarrow G)$ and the claim that G has a left adjoint follows from the general adjoint functor theorem (see Theorem 1.4.36). $\square$

Since Σ_g is a full subcategory of Σ_{ft} it follows that Σ_g is a reflective subcategory of Σ_{ft} . A constructive proof may perhaps be achieved by adapting the construction of the Grothendieck group (see Example 1.4.32) to infinitary addition Σ . Unfortunately, as in the case of the left adjoint to $\Sigma_{ft} \hookrightarrow \Sigma_w$, it is unclear whether quotienting would be necessary to guarantee that the bracketing axiom is satisfied and, if quotienting is required, proving well-definedness of Σ becomes exceedingly subtle.

2.1.5 Tensor product of Σ -monoids

The results in this section are adapted from those in the appendix of Hoshino's work on the category of strong Σ -monoids [33]. The proofs presented here are more detailed and apply to all of the Σ_* categories but, in essence, the strategy they follow is due to Hoshino.

Proposition 2.1.16 already established that each Σ_* category has small products; denote the terminal object as $(\{0\}, \Sigma) \in \Sigma_*$ whose Σ is the unique total function of its type. Consequently, Σ_* can be given a Cartesian monoidal structure.

Proposition 2.1.21. *For each Σ_* category, let $\times: \Sigma_* \times \Sigma_* \rightarrow \Sigma_*$ be the functor mapping each pair of objects to their categorical product and each pair of morphisms to the universal morphism that makes the following diagram*

$$\begin{array}{ccccc} X & \xleftarrow{\pi_l} & X \times Y & \xrightarrow{\pi_r} & Y \\ f \downarrow & & \downarrow & & \downarrow g \\ Z & \xleftarrow{\pi_l} & Z \times W & \xrightarrow{\pi_r} & W \end{array}$$

commute. Then, each $(\Sigma_*, \times, \{0\})$ is a symmetric monoidal category.

Proof. This construction is well-known for categories with finitary products. Unitors are given by projections and associators are given by the fact that both $(X \times Y) \times Z$ and $X \times (Y \times Z)$ are categorical products of the collection $\{X, Y, Z\}$, so they are isomorphic. The (symmetric) braiding is given by the fact that $X \times Y$ and $Y \times X$ are both products and, therefore, isomorphic. All coherence diagrams follow from the universal properties of categorical products and terminal object. $\square$

Unfortunately such a Cartesian monoidal structure is not suitable for our purposes: we are interested in defining a monoidal structure on each Σ_* category so that categories enriched over Σ_* satisfy that their composition distributes over addition. As discussed in Section 1.4.2, to do so we need that each of the Σ_* categories has tensor products and use these to define their monoidal structure. Such is the goal of this section, which begins by defining the notion of Σ -bilinear function.

Definition 2.1.22. For each Σ_* category, let $X, Y, Z \in \Sigma_*$ be arbitrary objects and let $f: X \times Y \rightarrow Z$ be a function. We say f is a Σ -bilinear function if $f(x, -)$ and $f(-, y)$ are Σ -homomorphisms for all $x \in X$ and all $y \in Y$. Let $\Sigma_*^{X,Y}(Z)$ be the set of all Σ -bilinear functions of type $X \times Y \rightarrow Z$.

Remark 2.1.23. It immediately follows from the definition of a Σ -bilinear function $f: X \times Y \rightarrow Z$ that $f(x, y) = 0$ if $x = 0$ or $y = 0$:

$$f(x, 0) = f(x, \Sigma\emptyset) = \Sigma\emptyset = 0.$$

Furthermore, if $x \in X$ has an inverse then $f(\Sigma\{x, -x\}, y) = f(0, y) = 0$ and, due to f being Σ -bilinear:

$$\Sigma\{f(x, y), f(-x, y)\} \simeq 0.$$

When the codomain Z is a Σ -group the uniqueness of the inverse element implies $-f(x, y) = f(-x, y)$ and similarly $-f(x, y) = f(x, -y)$ if $y \in Y$ has an inverse.

For each Σ_* category and every pair of objects $X, Y \in \Sigma_*$ there is a functor $\Sigma_*^{X,Y}: \Sigma_* \rightarrow \mathbf{Set}$ that maps each object $Z \in \Sigma_*$ to the set of Σ -bilinear functions $\Sigma_*^{X,Y}(Z)$ and maps each Σ -homomorphism $f: Z \rightarrow W$ to a function $\Sigma_*^{X,Y}(Z) \rightarrow \Sigma_*^{X,Y}(W)$ that maps each $h \in \Sigma_*^{X,Y}(Z)$ to the Σ -bilinear function $f \circ h$.

Lemma 2.1.24. For all $X, Y \in \Sigma_w$, the functor $\Sigma_w^{X,Y}: \Sigma_w \rightarrow \mathbf{Set}$ preserves limits.

Proof. Let $Z, W \in \Sigma_w$ be arbitrary objects, let $f, g \in \Sigma_w(Z, W)$ be arbitrary Σ -homomorphisms and let $E \in \Sigma_w$ be their equalizer constructed as in Proposition 2.1.10.

Recall that $E \subseteq Z$ and $z \in E$ implies $f(z) = g(z)$ so it is clear that any $h \in \Sigma_w^{X,Y}(E)$ satisfies $f \circ h = g \circ h$. Conversely, if a Σ -bilinear function $h: X \times Y \rightarrow Z$ satisfies $f \circ h = g \circ h$ then, for every element $z \in \text{im}(h)$, it holds that $f(z) = g(z)$, so $\text{im}(h) \subseteq E$. Therefore, $\Sigma_w^{X,Y}(E)$ is precisely the subset of $\Sigma_w^{X,Y}(Z)$ such that for all $h \in \Sigma_w^{X,Y}(E)$,

$$\Sigma_w^{X,Y}(f)(h) = f \circ h = g \circ h = \Sigma_w^{X,Y}(g)(h)$$

i.e. $\Sigma_w^{X,Y}(E)$ is an equalizer of the diagram in **Set**. Thus, $\Sigma_w^{X,Y}$ preserves equalizers.

Let $Z \times W \in \Sigma_w$ be the categorical product of two weak Σ -monoids and let π_l and π_r be the corresponding projections. Since the categorical product in **Set** is given by the Cartesian product, there is a unique function m making the following diagram

$$\begin{array}{ccccc} & & \Sigma_w^{X,Y}(Z \times W) & & \\ & \swarrow \Sigma_w^{X,Y}(\pi_l) & \downarrow m & \searrow \Sigma_w^{X,Y}(\pi_r) & \\ \Sigma_w^{X,Y}(Z) & \xleftarrow{\pi_l} & \Sigma_w^{X,Y}(Z) \times \Sigma_w^{X,Y}(W) & \xrightarrow{\pi_r} & \Sigma_w^{X,Y}(W) \end{array}$$

commute in **Set**. Such a function m maps every Σ -bilinear function $f \in \Sigma_w^{X,Y}(Z \times W)$ to the pair of Σ -bilinear functions $(\pi_l \circ f, \pi_r \circ f)$. Conversely, there is a function $u: \Sigma_w^{X,Y}(Z) \times \Sigma_w^{X,Y}(W) \rightarrow \Sigma_w^{X,Y}(Z \times W)$ mapping each pair of Σ -bilinear functions (g, h) to the function given below:

$$k(x, y) = (g(x, y), h(x, y))$$

for every $x \in X$ and $y \in Y$. Notice that u is well-defined since k is a Σ -bilinear function:

$$\begin{aligned} k(\Sigma \mathbf{x}, y) &= (g(\Sigma \mathbf{x}, y), h(\Sigma \mathbf{x}, y)) = (\Sigma g(\mathbf{x}, y), \Sigma h(\mathbf{x}, y)) \\ &= \Sigma^\times (g(\mathbf{x}, y), h(\mathbf{x}, y)) = \Sigma^\times k(\mathbf{x}, y) \end{aligned}$$

where $g(\mathbf{x}, y)$ is a shorthand for the family obtained after applying $g(-, y)$ to each element in $\mathbf{x}$. It is straightforward to check that $u \circ m = \text{id}$ and $m \circ u = \text{id}$ so that $\Sigma_w^{X,Y}(Z \times W)$ is isomorphic to $\Sigma_w^{X,Y}(Z) \times \Sigma_w^{X,Y}(W)$. Thus, it follows that $\Sigma_w^{X,Y}(Z \times W)$ is a categorical product and $\Sigma_w^{X,Y}$ preserves binary products. It is straightforward to generalise this argument to small products and, hence, $\Sigma_w^{X,Y}$ preserves small products.

A functor that preserves equalizers and small products and whose domain is a complete category automatically preserves all limits. Therefore, $\Sigma_w^{X,Y}$ preserves limits, as claimed. $\square$

Corollary 2.1.25. *For every Σ_* category and every pair of objects $X, Y \in \Sigma_*$ the functor $\Sigma_*^{X,Y} : \Sigma_* \rightarrow \mathbf{Set}$ preserves limits.*

Proof. The previous lemma establishes the claim for Σ_w . For the rest of the Σ_* categories, it is evident that the corresponding functor $\Sigma_*^{X,Y}$ is equal to the composition $\Sigma_w^{U(X),U(Y)} \circ U$ where U is the embedding functor $\Sigma_* \hookrightarrow \Sigma_w$. Proposition 2.1.16 established that U preserves limits, so it follows that $\Sigma_*^{X,Y}$ preserves limits. $\square$

Lemma 2.1.26. *The comma category $(\{\bullet\} \Rightarrow \Sigma_w^{X,Y})$ has an initial object. Such an initial object will be denoted $(p, X \otimes Y)$.*

Proof. The category Σ_w is complete (see Proposition 2.1.10) and Lemma 2.1.24 establishes that the functor $\Sigma_w^{X,Y}$ preserves limits. Consequently, the comma category $(\{\bullet\} \Rightarrow \Sigma_w^{X,Y})$ is complete (see Lemma A.2 from [38]) and, since Σ_w is locally small, it follows that the comma category is locally small. Then, according to Lemma 1.4.35, it suffices to provide a weakly initial set to prove that the comma category $(\{\bullet\} \Rightarrow \Sigma_w^{X,Y})$ has an initial object. The elements of such a weakly initial set ought to be functions of type $\{\bullet\} \rightarrow \Sigma_w^{X,Y}(A)$ for some $A \in \Sigma_w$; but a function with singleton domain simply selects an element in its codomain, so it is equivalent to think of the weakly initial set as a collection of Σ -bilinear functions $X \times Y \xrightarrow{q} A$. Let S be the collection of all Σ -bilinear functions with domain $X \times Y$ and whose codomain may be any weak Σ -monoid whose underlying set is a quotient of a subset of $(X \times Y)^*$. Notice that S is small since all quantifiers in its definition are with respect to fixed sets X and Y and $(X \times Y)^*$ is small (see Proposition 2.0.2).

Let $W \in \Sigma_w$ and let $f : X \times Y \rightarrow W$ be a Σ -bilinear function. Define a subset $Z \subseteq (X \times Y)^*$ as follows:

$$Z = \{z \in (X \times Y)^* \mid fz \text{ is summable}\}$$

and define an equivalence relation $\sim$ on Z where:

$$z \sim z' \iff \Sigma fz = \Sigma fz'.$$

Define a function $\bar{f} : Z/\sim \rightarrow W$ so that each equivalence class $[z] \in Z/\sim$ is mapped to Σfz ; it is straightforward to check that $\bar{f}$ is injective and, moreover, $[\emptyset] \in Z/\sim$, so the neutral element $0 \in W$ is in the image of $\bar{f}$. Therefore, according to Lemma 2.1.8, $Z/\sim$ may be endowed with a partial function $\Sigma^{\bar{f}}$ defined as follows for every family $\{[z_i]\}_I \in (Z/\sim)^*$:

$$\Sigma^{\bar{f}}\{[z_i]\}_I = \begin{cases} [z] & \text{if } \exists [z] \in Z/\sim \text{ s.t. } \Sigma\{\bar{f}[z_i]\}_I \simeq \bar{f}[z] \\ \text{undefined} & \text{otherwise} \end{cases}$$

so that $(Z/\sim, \Sigma^{\bar{f}})$ is a weak Σ -monoid and $\bar{f}$ is a Σ -homomorphism. Finally, define the function $q: X \times Y \rightarrow Z/\sim$ as the composite

$$q = X \times Y \xrightarrow{\{-\}} Z \xrightarrow{[-]} Z/\sim$$

where $\{-\}$ is the function mapping each $(x, y) \in X \times Y$ to the singleton family $(\{(x, y)\})$ and $[-]$ is the quotient map. It is straightforward to check that $f = \bar{f} \circ q$ and, for any $x \in X$ and any summable family $\{y_i\}_I \in Y^*$,

$$\Sigma\{\bar{f}q(x, y_i)\}_I = \Sigma\{f(x, y_i)\}_I \simeq f(x, \Sigma\{y_i\}_I) = \bar{f}q(x, \Sigma\{y_i\}_I)$$

since f is Σ -bilinear, implying that the family $\{q(x, y_i)\}_I$ is summable in $(Z/\sim, \Sigma^{\bar{f}})$ with:

$$\Sigma^{\bar{f}}\{q(x, y_i)\}_I \simeq q(x, \Sigma\{y_i\}_I).$$

A similar result holds if we fix $y \in Y$ instead and let $\mathbf{x} \in X^*$ be an arbitrary summable family, thus, q is a Σ -bilinear function.

In summary, it has been shown that $Z/\sim$ is an object in Σ_w and that $X \times Y \xrightarrow{q} Z/\sim$ is a Σ -bilinear function, hence, an element of S . Moreover, $\bar{f}$ is a Σ -homomorphism and $f = \bar{f} \circ q$, implying that $\bar{f}$ is a valid morphism from q to f in the comma category. This construction may be reproduced for any Σ -bilinear function $f: X \times Y \rightarrow W$, so it follows that S is a weakly initial set. Thus, Lemma 1.4.35 establishes that the comma category $(\{\bullet\} \Rightarrow \Sigma_w^{X,Y})$ has an initial object, as claimed. $\square$

Lemma 2.1.27. *For each Σ_* category, the comma category $(\{\bullet\} \Rightarrow \Sigma_*^{X,Y})$ has an initial object. Such an initial object will be denoted $(p, X \otimes Y)$.*

Proof. We must show that whenever X, Y and W are objects in certain Σ_* category, the weak Σ -monoid $(Z/\sim, \Sigma^{\bar{f}})$ defined in the previous lemma is in fact an object in the same Σ_* category. Such a result has already been established for Σ_w in the previous lemma.

Assume $W \in \Sigma_s$; then, $\Sigma^{\bar{f}}$ satisfies strong bracketing and strong flattening, the argument being the same as that for (weak) bracketing and flattening (see Lemma 2.1.8): these follow directly from their counterpart in W since $\Sigma^{\bar{f}}$ is defined if and only if the sum in W is defined. Let $\{[z_i]\}_I$ be an arbitrary summable family in $(Z/\sim)^*$; by definition of $\Sigma^{\bar{f}}$, the fact that it is summable implies that the family $\{\bar{f}[z_i]\}_I \in W^*$ is summable or, equivalently, the family $\{\Sigma f z_i\}_I \in W^*$ is summable. Then, due to strong flattening in W , we know that $\biguplus_I f z_i$ is summable and, thanks to subsummability in W , the subfamily $\biguplus_J f z_j$ is summable for every $J \subseteq I$. It then follows from

strong bracketing in W that the subfamily $\{\Sigma f \mathbf{z}_j\}_J$ is summable or, equivalently, $\{\bar{f}[\mathbf{z}_j]\}_J$ is summable. Thus, every subfamily of $\{[\mathbf{z}_i]\}_I$ is summable, implying that $\Sigma \bar{f}$ satisfies subsummability. Consequently, if $W \in \Sigma_s$ then $Z/\sim \in \Sigma_s$.

Assume $W \in \Sigma_{ft}$ and let $\{[\mathbf{z}_i]\}_I$ be an arbitrary finite family in $(Z/\sim)^*$. Notice that $\{\Sigma f \mathbf{z}_i\}_I$ is a summable family due to W being finitely total. Since I is finite and each $f \mathbf{z}_i$ is summable (due to the definition of Z), the flattening axiom in W implies that $\uplus_I f \mathbf{z}_i$ is summable:

$$\Sigma(\uplus_I f \mathbf{z}_i) \simeq \Sigma\{\Sigma f \mathbf{z}_i\}_I.$$

Consequently, $\uplus_I f \mathbf{z}_i \in Z$ and

$$\Sigma \bar{f}\{[\mathbf{z}_i]\}_I \simeq [\uplus_I f \mathbf{z}_i]$$

implying that every finite family in $Z/\sim$ is summable. Consequently, if $W \in \Sigma_{ft}$ then $Z/\sim \in \Sigma_{ft}$.

Assume that both X and W are Σ -groups and recall that $-f(x, y) = f(-x, y)$ (see Remark 2.1.23). For every element $\mathbf{z} = \{(x_i, y_i)\}_I$ of Z let $-\mathbf{z}$ be the family $\{(-x_i, y_i)\}_I$; notice that $-\mathbf{z}$ is in Z as well since its image under f is summable in W :

$$\Sigma\{f(-x_i, y_i)\}_I = \Sigma\{-f(x_i, y_i)\}_I \simeq -\Sigma\{f(x_i, y_i)\}_I = -\Sigma f \mathbf{z} \quad (2.10)$$

where the $\simeq$ step follows from the inversion map being a Σ -homomorphism as required for W to be a Σ -group. For every $[\mathbf{z}] \in Z/\sim$ there is an inverse element $[-\mathbf{z}] \in Z/\sim$ since:

$$\Sigma\{\bar{f}[\mathbf{z}], \bar{f}[-\mathbf{z}]\} = \Sigma\{\Sigma f \mathbf{z}, -\Sigma f \mathbf{z}\} = 0.$$

Such an inversion map $[\mathbf{z}] \mapsto [-\mathbf{z}]$ is well-defined since it is immediate from (2.10) that $\mathbf{z} \sim \mathbf{z}'$ implies $-\mathbf{z} \sim -\mathbf{z}'$. It remains to show that the inversion map in $Z/\sim$ is a Σ -homomorphism. Assume $\{[\mathbf{z}_i]\}_I$ is a summable family in $(Z/\sim)^*$; this implies the existence of a family $\mathbf{z} \in Z$ such that:

$$\Sigma\{\bar{f}[\mathbf{z}_i]\}_I \simeq \bar{f}[\mathbf{z}].$$

But it is immediate from (2.10) that $\bar{f}[-\mathbf{z}'] = -\bar{f}[\mathbf{z}']$ for all $\mathbf{z}' \in Z$, thus:

$$\Sigma\{\bar{f}[-\mathbf{z}_i]\}_I = \Sigma\{-\bar{f}[\mathbf{z}_i]\}_I \simeq -\Sigma\{\bar{f}[\mathbf{z}_i]\}_I \simeq -\bar{f}[\mathbf{z}] = \bar{f}[-\mathbf{z}]$$

and, hence, $\Sigma \bar{f}\{[-\mathbf{z}_i]\}_I \simeq [-\mathbf{z}]$. Consequently, the inversion map in $Z/\sim$ is a Σ -homomorphism and we conclude that, if $W \in \Sigma_g$ and $X \in \Sigma_g$ (or $Y \in \Sigma_g$) then $Z/\sim \in \Sigma_g$.

In summary, it has been shown that if X, Y and W are all objects from the same Σ_* category and $f: X \times Y \rightarrow W$ is a Σ -bilinear function, then $(Z/\sim, \Sigma^{\bar{f}})$ as constructed in Lemma 2.1.26 is an object in the same Σ_* category. Then, thanks to Proposition 2.1.16 and Corollary 2.1.25 it is straightforward to check that the argument from the previous lemma holds for any Σ_* category, concluding the proof. $\square$

The previous lemma establishes the existence of an object $X \otimes Y$ known as the tensor product and a Σ -bilinear function $p: X \times Y \rightarrow X \otimes Y$ that let us uniquely represent any Σ -bilinear function $f: X \times Y \rightarrow Z$ as a Σ -homomorphism $\bar{f}: X \otimes Y \rightarrow Z$ so that $f = \bar{f} \circ p$. In the case of Σ -groups, a constructive proof may perhaps be achieved by adapting the construction of the tensor product in abelian groups (see [53]) to infinitary addition Σ . Unfortunately, as in the case of the left adjoint to $\Sigma_{\text{ft}} \hookrightarrow \Sigma_{\text{w}}$ (see Section 2.1.3), it is unclear whether quotienting the underlying set would be necessary to guarantee that the bracketing axiom is satisfied and, if quotienting is required, proving well-definedness of Σ becomes exceedingly subtle.

The goal of the rest of this section is to show that every Σ_* category has a (closed symmetric) monoidal structure given by this tensor product. However, the monoidal structure is somewhat different for each of these categories; this is illustrated by the differences in the definition of their tensor unit, given below.

Definition 2.1.28. Let $S: \mathbb{N}^* \rightarrow \mathbb{N} \cup \{\infty\}$ be the standard sum of families of natural numbers. For each category Σ_* , define the *tensor unit* $I \in \Sigma_*$ as follows:

- in Σ_{w} , let $I = \{0, 1\}$ along with $\Sigma \mathbf{n} = S \mathbf{n}$ if $S \mathbf{n} \leq 1$ and otherwise undefined;
- in Σ_{s} , let I and its Σ be the same as in Σ_{w} ;
- in Σ_{ft} , let $I = \mathbb{N}$ along with $\Sigma \mathbf{n} = S \mathbf{n}$ if $S \mathbf{n}$ is finite and otherwise undefined;
- in Σ_{g} , let $I = \mathbb{Z}$ along with $\Sigma \mathbf{n} = (S m_+ \mathbf{n}) + (S m_- \mathbf{n})$ if $S m_+ \mathbf{n}$ and $S m_- \mathbf{n}$ are finite and otherwise undefined; where m_+ maps all negative integers to 0 and m_- maps all positive integers to 0.

It is immediate to check that each of these tensor units are well-defined Σ -monoids in their respective categories. These definitions are accompanied by functions $l_X: I \times X \rightarrow X$ and $r_X: X \times I \rightarrow X$ for every $X \in \Sigma_*$, defined as follows:

$$\begin{aligned} l_X(n, x) &= \Sigma \{ \overbrace{x, x, \dots}^{n \text{ times}} \} \\ r_X(x, n) &= \Sigma \{ \overbrace{x, x, \dots}^{n \text{ times}} \} \end{aligned} \tag{2.11}$$

except for Σ_g where, if $n < 0$, then the corresponding family is of $|n|$ copies of $-x$ instead. It is immediate that $l_X(-, x)$ is a Σ -homomorphism. To show that $l_X(n, -)$ is also a Σ -homomorphism, let $\mathbf{x} = \{x_i\}_I \in X^*$ be an arbitrary summable family; then, since n is finite we may apply flattening followed by bracketing to obtain:

$$l(n, \Sigma \mathbf{x}) = \Sigma \{\overbrace{\Sigma \mathbf{x}, \dots}^{n \text{ times}}\} \simeq \Sigma \{\overbrace{\Sigma \{x_i, x_i, \dots\}}^{n \text{ times}}\}_I = \Sigma \{l(n, x_i)\}_I.$$

Therefore, l_X is a Σ -bilinear function and a similar argument holds for r_X . Moreover, it is straightforward to check that l and r are natural transformations since for any Σ -homomorphism $f: X \rightarrow Y$,

$$l(\text{id} \times f)(n, x) = l(n, f(x)) = \Sigma \{\overbrace{f(x), \dots}^{n \text{ times}}\} = f(\Sigma \{\overbrace{x, \dots}^{n \text{ times}}\}) = fl(n, x).$$

Corollary 2.1.29. *For any Σ_* category, let l_X and r_X be the Σ -bilinear functions defined in (2.11) and let a be the associator from the Cartesian monoidal structure on each Σ_* . Then, the following diagram commutes:*

$$\begin{array}{ccc} (X \times I) \times Y & \xrightarrow{a} & X \times (I \times Y) \\ \downarrow r \times \text{id} & & \text{id} \times l \downarrow \\ X \times Y & \xrightarrow{p} & X \otimes Y \\ & \nwarrow p & \nearrow p \\ & X \otimes Y & \end{array}$$

Proof. For all Σ_* categories, the claim follows from the following:

$$p(\text{id} \times l_Y)(x, n, y) = p(x, \Sigma \{\overbrace{y, \dots}^{n \text{ times}}\}) = \Sigma \{\overbrace{p(x, y), \dots}^{n \text{ times}}\} = p(r_X \times \text{id})(x, n, y)$$

due to p being a Σ -bilinear function. $\square$

These Σ -bilinear functions l and r will be used to define the unitors of the monoidal structure given by $\otimes$. The associator is trickier since the Σ -homomorphism representation of a ‘ Σ -trilinear’ function is not yet clear. The following two lemmas deal with this, proving that the object $(X \otimes Y) \otimes Z$ along with the ‘ Σ -trilinear’ function $p \circ (p \times \text{id})$ can take up the role of the ternary tensor product. To do so, we must first introduce a Σ -monoid structure on sets of Σ -homomorphisms.

Lemma 2.1.30. *For every Σ_* category and objects $X, Y \in \Sigma_*$, the hom-set $[X, Y] = \Sigma_*(X, Y)$ may be endowed with a partial function $\Sigma^\rightarrow: [X, Y]^* \rightharpoonup [X, Y]$ so that $[X, Y] \in \Sigma_*$.*

Proof. The partial function $\Sigma^\rightarrow$ is defined in the same manner for all of the Σ_* categories; it is defined pointwise, using the Σ from Y . For every family $\mathbf{f} = \{f_i\}_I \in [X, Y]^*$ and every $x \in X$ let $\mathbf{f}(x)$ be a shorthand for the family $\{f_i(x)\}_I \in Y^*$ and define a partial function $s_{\mathbf{f}}: X \rightarrow Y$ as follow for every $x \in X$:

$$s_{\mathbf{f}}(x) = \begin{cases} y & \text{if } \exists y \in Y \text{ s.t. } \Sigma \mathbf{f}(x) \simeq y \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Define the partial function $\Sigma^\rightarrow: [X, Y]^* \rightarrow [X, Y]$ as follows for every $\mathbf{f} \in [X, Y]^*$:

$$\Sigma^\rightarrow \mathbf{f} = \begin{cases} s_{\mathbf{f}} & \text{if } s_{\mathbf{f}} \in [X, Y] \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Recall that all Σ -homomorphisms are total functions, so the condition $s_{\mathbf{f}} \in [X, Y]$ imposes that $\Sigma \mathbf{f}(x)$ is defined for all $x \in X$. We now show that if Y is an object in a Σ_* category then $([X, Y], \Sigma^\rightarrow)$ is an object in the same category.

Assume $Y \in \Sigma_w$; it is immediate that for every singleton family $\{f\} \in [X, Y]^*$ the corresponding function $s_{\{f\}}$ maps every $x \in X$ to $f(x)$, so $\Sigma^\rightarrow \{f\} \simeq f$ as required by the singleton axiom. Similarly, it is immediate that the function $s_{\emptyset}$ maps every $x \in X$ to the neutral element $0 \in Y$, so the empty family is summable in $([X, Y], \Sigma^\rightarrow)$, with the neutral element being $s_{\emptyset}$. Moreover, let $\mathbf{f} \in [X, Y]^*$ be a summable family and let $\mathbf{f}_{\emptyset}$ be the subfamily where all occurrences of $s_{\emptyset}$ have been removed; it is immediate that $\mathbf{f}_{\emptyset}$ is summable since $s_{\emptyset}$ only contributes to the sum of $\mathbf{f}(x) \in Y^*$ by adding a 0, which may be disregarded thanks to the neutral element axiom in Y . Consequently, the neutral element axiom is satisfied by $\Sigma^\rightarrow$; it remains to prove the bracketing and flattening axioms. Let $\{\mathbf{f}_j \in [X, Y]^*\}_J$ be a collection of summable families and let $\mathbf{f} = \uplus_J \mathbf{f}_j$. Assume that $\mathbf{f}_j$ is a finite family for every $j \in J$ and assume that $\mathbf{f}$ is summable; to prove bracketing we must show that the family $\mathbf{g} = \{\Sigma^\rightarrow \mathbf{f}_j\}_J \in [X, Y]^*$ is summable and $s_{\mathbf{g}} = s_{\mathbf{f}}$. This is straightforward since for every $x \in X$ we have that:

$$s_{\mathbf{g}}(x) = \Sigma\{(\Sigma^\rightarrow \mathbf{f}_j)(x)\}_J = \Sigma\{\Sigma \mathbf{f}_j(x)\}_J \simeq \Sigma \mathbf{f}(x) = s_{\mathbf{f}}(x)$$

where the $\simeq$ step corresponds to bracketing in Y . Therefore, $s_{\mathbf{g}} \in [X, Y]$ is implied by $s_{\mathbf{f}} \in [X, Y]$ — which holds due to $\mathbf{f}$ being summable — and, hence, $\mathbf{g} = \{\Sigma^\rightarrow \mathbf{f}_j\}_J$ is summable with its sum matching that of $\mathbf{f}$. Flattening is proven via the same argument, this time using flattening in Y instead of bracketing. Consequently, if $Y \in \Sigma_w$ then $[X, Y] \in \Sigma_w$.

Assume $Y \in \Sigma_s$; then, the same argument that established bracketing and flattening in $[X, Y]$ can be used to prove strong bracketing and strong flattening in $[X, Y]$ via the corresponding axioms from Y . Moreover, if $\mathbf{f}$ is summable and $\mathbf{h}$ is a subfamily of $\mathbf{f}$, it follows from subsummability in Y that $s_{\mathbf{h}}$ is a total function. To prove that $\Sigma^\rightarrow$ satisfies subsummability, we must show that $s_{\mathbf{h}}$ is a Σ -homomorphism; this follows from strong flattening and strong bracketing in Y , as described below. Let $\mathbf{h} = \{h_j\}_J$ and let $\{x_i\}_I \in X^*$ be an arbitrary summable family; then:

$$\begin{aligned}
 s_{\mathbf{h}}(\Sigma\{x_i\}_I) &= \Sigma\{h_j(\Sigma\{x_i\}_I)\}_J && \text{(definition of } s_{\mathbf{h}}) \\
 &\simeq \Sigma\{\Sigma\{h_j(x_i)\}_I\}_J && (h_j \in [X, Y]) \\
 &\simeq \Sigma\{h_j(x_i)\}_{I \times J} && \text{(strong flattening in } Y \text{ and } \uplus_J I \cong I \times J) \\
 &\simeq \Sigma\{\Sigma\{h_j(x_i)\}_J\}_I && \text{(strong bracketing in } Y \text{ and } s_{\mathbf{h}} \text{ total)} \\
 &= \Sigma\{s_{\mathbf{h}}(x_i)\}_I && \text{(definition of } s_{\mathbf{h}})
 \end{aligned}$$

so $s_{\mathbf{h}} \in [X, Y]$ and $\mathbf{h}$ is summable, implying that subsummability is satisfied. Consequently, if $Y \in \Sigma_s$ then $[X, Y] \in \Sigma_s$.

Assume $Y \in \Sigma_{\text{ft}}$; it is immediate from Y being finitely total that, for every finite family $\mathbf{h} \in [X, Y]^*$, the function $s_{\mathbf{h}}$ is total. Once again, to prove that $\mathbf{h}$ is summable in $[X, Y]$ we must show that $s_{\mathbf{h}}$ is a Σ -homomorphism; this can be achieved via the same argument used above to prove subsummability but, in this case, only (weak) flattening and bracketing are required since $\mathbf{h}$ is assumed to be a finite family. Consequently, if $Y \in \Sigma_{\text{ft}}$ then $[X, Y] \in \Sigma_{\text{ft}}$.

Assume $Y \in \Sigma_g$ and let $f \in [X, Y]$. Due to Y having inverses, we can define a function that maps every x to $-f(x)$, call it $-f$. Since the inversion map in Y is a Σ -homomorphism and so is f , it is trivial to check that $-f \in [X, Y]$. Thanks to $\Sigma^\rightarrow$ being defined pointwise, it is straightforward to check that $\Sigma^\rightarrow\{f, -f\}$ is the function sending every x to $0 \in Y$ — *i.e.* the neutral element $s_\emptyset$ in $[X, Y]$ — and the inversion map $f \mapsto -f$ is a Σ -homomorphism. Consequently, if $Y \in \Sigma_g$ then $[X, Y] \in \Sigma_g$.

In conclusion, it has been shown that $([X, Y], \Sigma^\rightarrow)$ is an object in the same Σ_* category as Y , proving the claim. $\square$

We may now make use of the Cartesian closed structure in **Set** together with the definition of tensor product in Σ_* categories to lift the universal property of the binary tensor product to the ternary case.

Definition 2.1.31. For each Σ_* category, let $X, Y, Z, W \in \Sigma_*$ be arbitrary objects and let $f: X \times Y \times Z \rightarrow W$ be a function. We say f is a Σ -trilinear function if $f(x, y, -)$,

$f(x, -, z)$ and $f(-, y, z)$ are Σ -homomorphisms for every choice of $x \in X$, $y \in Y$ and $z \in Z$.

Lemma 2.1.32. *For every Σ_* category and objects $X, Y, Z, W \in \Sigma_*$, let $f: (X \times Y) \times Z \rightarrow W$ be an arbitrary Σ -trilinear function. Then, there is a unique Σ -homomorphism $(X \otimes Y) \otimes Z \rightarrow W$ making the diagram*

$$\begin{array}{ccc} (X \times Y) \times Z & \xrightarrow{f} & W \\ & \searrow p \circ (p \times \text{id}) & \uparrow \text{---} \\ & & (X \otimes Y) \otimes Z \end{array}$$

commute in **Set**.

Proof. For every Σ_* category and objects $Z, W \in \Sigma_*$, define a function $\text{ev}: [Z, W] \times Z \rightarrow W$ that maps each pair (h, z) to $h(z)$. Let $h \in [Z, W]$ be an arbitrary Σ -homomorphism and $\mathbf{z} = \{z_i\}_I \in Z$ an arbitrary summable family, then:

$$\text{ev}(h, \Sigma \mathbf{z}) = h(\Sigma \mathbf{z}) \simeq \Sigma h \mathbf{z} = \Sigma \{\text{ev}(h, z_i)\}_I.$$

Similarly, let $\mathbf{h} = \{h_i\}_I \in [Z, W]^*$ be an arbitrary summable family and fix an arbitrary $z \in Z$, then:

$$\text{ev}(\Sigma^{\rightarrow} \mathbf{h}, z) = \text{ev}(s_{\mathbf{h}}, z) = s_{\mathbf{h}}(z) = \Sigma \{h_i(z)\}_I = \Sigma \{\text{ev}(h_i, z)\}_I.$$

Therefore, ev is a Σ -bilinear function. Let $f: (X \times Y) \times Z \rightarrow W$ be a Σ -trilinear function. Since **Set** is Cartesian closed, there is a unique function $\Lambda f: X \times Y \rightarrow [Z, W]$ making the diagram

$$\begin{array}{ccc} [Z, W] \times Z & & \\ \Lambda f \times \text{id} \uparrow & \searrow \text{ev} & \\ (X \times Y) \times Z & \xrightarrow{f} & W \end{array} \quad (2.12)$$

commute in **Set**. The function Λf maps each pair $(x, y) \in X \times Y$ to the function $f(x, y, -)$, which is a Σ -homomorphism since f is Σ -trilinear; we now show that Λf is Σ -bilinear. Fix $x \in X$ and let $\mathbf{y} = \{y_i\}_I \in Y^*$ be an arbitrary summable family; then, for every $z \in Z$:

$$\begin{aligned} \Sigma \{\Lambda f(x, y_i)(z)\}_I &= \Sigma \{f(x, y_i, z)\}_I && \text{(definition of } \Lambda f) \\ &\simeq f(x, \Sigma \mathbf{y}, z) && (f \text{ is } \Sigma\text{-trilinear}) \\ &= \Lambda f(x, \Sigma \mathbf{y})(z) && \text{(definition of } \Lambda f) \end{aligned}$$

with $\Lambda f(x, \Sigma y)$ a Σ -homomorphism, as previously discussed. Therefore,

$$\Sigma \rightarrow \{\Lambda f(x, y_i)\}_I \simeq \Lambda f(x, \Sigma y).$$

The same can be said for summable families $\mathbf{x} \in X^*$ if $y \in Y$ is fixed instead, hence, Λf is a Σ -bilinear function.

Since both Λf and ev are a Σ -bilinear functions, the diagram

$$\begin{array}{ccccc} & & [Z, W] \times Z & & \\ & \nearrow \overline{\Lambda f} \times \text{id} & \uparrow \Lambda f \times \text{id} & \searrow \text{ev} & \\ (X \otimes Y) \times Z & \xleftarrow{p \times \text{id}} & (X \times Y) \times Z & \xrightarrow{f} & W \\ & \searrow p & & \nearrow & \\ & & (X \otimes Y) \otimes Z & & \end{array}$$

commutes in **Set** with each dashed arrow denoting uniqueness. The top right triangle is diagram (2.12) which was already established to commute. Because Λf is a Σ -bilinear function, Lemma 2.1.27 implies that the top left triangle commutes, where $\overline{\Lambda f}$ is a Σ -homomorphism. Considering that ev is a Σ -bilinear function, it is straightforward to check that $\text{ev} \circ (\overline{\Lambda f} \times \text{id})$ is also a Σ -bilinear function; then Lemma 2.1.27 implies that the outer edges of the diagram commute, with the corresponding unique function $(X \otimes Y) \otimes Z \rightarrow W$ being a Σ -homomorphism. Consequently, the triangle at the bottom of the diagram commutes. Notice that if we were to find another Σ -homomorphism $g: (X \otimes Y) \otimes Z \rightarrow W$ making the bottom triangle commute, such g would also make the outer edges of the diagram commute but, according to Lemma 2.1.27, there is only one such Σ -homomorphism. Therefore, the Σ -homomorphism $(X \otimes Y) \otimes Z \rightarrow W$ from the claim has been shown to exist and be unique, completing the proof. $\square$

It is straightforward to check that the same argument allows us to factor any Σ -trilinear function $f: X \times (Y \times Z) \rightarrow W$ through $p \circ (\text{id} \times p)$ so that there is a unique Σ -homomorphism characterising f , *i.e.* both $(X \otimes Y) \otimes Z$ and $X \otimes (Y \otimes Z)$ are ternary tensor products. Finally, it is time to define the monoidal structure on the tensor product.

Definition 2.1.33. For every Σ_* category and Σ -homomorphism $X \xrightarrow{f} Z$ and $Y \xrightarrow{g} W$, let $f \otimes g$, λ , ρ and α be defined as the unique Σ -homomorphisms making the following diagrams commute:

$$\begin{array}{ccc} X \times Y & \xrightarrow{f \times g} & Z \times W \\ p \downarrow & & \downarrow p \\ X \otimes Y & \xrightarrow{f \otimes g} & Z \otimes W \end{array} \quad \begin{array}{ccc} I \times X & \xrightarrow{l} & X \\ p \downarrow & \nearrow \lambda & \\ I \otimes X & & \end{array} \quad \begin{array}{ccc} X \times I & \xrightarrow{r} & X \\ p \downarrow & \nearrow \rho & \\ X \otimes I & & \end{array}$$

$$\begin{array}{ccc}
(X \times Y) \times Z & \xrightarrow{a} & X \times (Y \times Z) \\
p \circ (p \times \text{id}) \downarrow & & \downarrow p \circ (\text{id} \times p) \\
(X \otimes Y) \otimes Z & \xrightarrow[\alpha]{} & X \otimes (Y \otimes Z).
\end{array}$$

Theorem 2.1.34. *Every Σ_* category is a monoidal category with monoidal product $\otimes$ and monoidal unit I .*

Proof. On objects $X, Y \in \Sigma_*$, the functor $- \otimes -$ yields $X \otimes Y$ and, on morphisms f and g , it yields the unique Σ -homomorphism $f \otimes g$ given in the definition above; it is straightforward to check that this is indeed a functor, thanks to functoriality of the Cartesian product and the universal property of $X \otimes Y$. Unitors λ and ρ and associators α are given in the definition above. The triangle equation follows from Corollary 2.1.29 after lifting all occurrences of $\times$ to $\otimes$ using p , whereas the pentagon equation follows from the pentagon equation of the Cartesian monoidal structure on Σ_* along with naturality of a . Naturality of λ , ρ and α follows from naturality of l , r and a . The inverse of α is induced from the inverse of a and the universal property of $X \otimes (Y \otimes Z)$. The inverse of λ is somewhat trickier: let $i: X \rightarrow I \times X$ be the function that maps each $x \in X$ to $(1, x)$; it is immediate that $l \circ i = \text{id}$, and $\lambda \circ p \circ i = \text{id}$ follows from the definition of λ . On the other hand:

$$p(1, l(n, x)) = p(1, \Sigma \{\overbrace{x, \dots}^{n \text{ times}}\}) \simeq \Sigma \{\overbrace{p(1, x), \dots}^{n \text{ times}}\} \simeq p(n, x)$$

so, $p \circ i \circ l = p$ and, by definition of λ , we have $p \circ i \circ \lambda \circ p = p$; this means that the Σ -bilinear function p may be characterised by the Σ -homomorphism $\bar{p} = p \circ i \circ \lambda$ but, trivially, $\bar{p} = \text{id}$ and due to uniqueness of $\bar{p}$ we conclude that $(p \circ i) \circ \lambda = \text{id}$. Therefore, the Σ -homomorphism $p \circ i$ is the inverse of λ and we may define the inverse of ρ by similar means. This completes the proof that $(\Sigma_*, \otimes, I)$ is a monoidal category for every Σ_* category. $\square$

Corollary 2.1.35. *For every Σ_* category, $(\Sigma_*, \otimes, I, [-, -])$ is a closed symmetric monoidal category.*

Proof. The symmetry of the monoidal structure is derived from that of the Cartesian monoidal structure, lifted using p . The closure follows from a similar argument to Lemma 2.1.32, but replacing each appearance of $\times$ with $\otimes$ and lifting through p so that all Σ -bilinear functions may be represented as Σ -homomorphisms and, hence, the diagrams can be made to commute in Σ_* . $\square$

2.2 Topological monoids

This section is meant to provide a brief introduction to the notions of topology that will be relevant to the thesis, namely, Hausdorff monoids, nets, limit points, continuous functions and their interaction with limits. For an in-depth introduction to topology see the book by John L. Kelley [37].

2.2.1 Brief introduction to general topology (*Preamble*)

Topological monoids are monoids whose underlying set has some additional structure, known as a *topology*. Sets with a topology are known as topological spaces; these are formally defined below.

Definition 2.2.1. Let X be a set and let τ be a subset of the power set $\mathcal{P}(X)$. The pair (X, τ) is said to be a *topological space* and τ its topology if the following conditions are satisfied:

- the sets $\emptyset$ and X are in τ ;
- for any $U, V \in \tau$, the intersection $U \cap V$ is in τ ;
- for any arbitrary (possibly infinite) indexed set $\{U_i \in \tau\}_{i \in I}$,

$$\bigcup_{i \in I} U_i \in \tau.$$

Let (X, τ) be a topological space; an element $U \in \tau$ is known as an *open set* and, if $x \in U$, we say that U is an *open neighbourhood* of x . Elements in X are known as *points*.

Although highly abstract, this definition has a down to earth interpretation which is helpful to keep in mind. An open neighbourhood of a point $x \in X$ is, as the name suggests, a subset of points in X that are perceived as being ‘near’ x . Let U and V be two open neighbourhoods of a point x ; if $U \subseteq V$, we say that no point in U is farther away from x than x is from the farthest point in V . Hence, a topology — being a collection of open neighbourhoods — provides a loose hierarchy of vicinity between points. A common way to define a topology on a set X is to provide a *base* for it: a collection of subsets of X from which the topology is unambiguously generated.

Definition 2.2.2. Let X be set. A *base for a topology* on X is a subset $\mathcal{B} \subseteq \mathcal{P}(X)$ such that

- for each element $x \in X$, there is at least one $B \in \mathcal{B}$ such that $x \in B$, and
- for all $B_1, B_2 \in \mathcal{B}$ and all $x \in X$,

$$x \in B_1 \cap B_2 \implies \exists B' \in \mathcal{B} \text{ s.t. } x \in B' \text{ and } B' \subseteq B_1 \cap B_2.$$

Each base $\mathcal{B}$ generates a topology on X , defined as the set of all unions (including infinite ones) of elements in $\mathcal{B}$.

Let τ and τ' be two topologies on the same set X . We say that τ is a *coarser topology* than τ' when $\tau \subseteq \tau'$, and *strictly coarser* when the containment is strict. We may show that a topology is coarser than another by studying their bases, as established below.

Proposition 2.2.3. *Let τ and τ' be topologies on X and let $\mathcal{B}$ and $\mathcal{B}'$ be bases for them, respectively. Then, $\tau \subseteq \tau'$ iff for each $x \in X$ and each $B \in \mathcal{B}$:*

$$x \in B \implies \exists B' \in \mathcal{B}' \text{ s.t. } B' \subseteq B \text{ and } x \in B'.$$

Proof. Assume $\tau \subseteq \tau'$; then automatically $B \in \tau'$, but $\mathcal{B}'$ generates τ' , so it must be that B can be obtained as the union of elements in $\mathcal{B}'$ and, in particular, if $x \in B$ then there must be a $B' \in \mathcal{B}'$ such that $B' \subseteq B$ and $x \in B'$. To prove the other direction, let $U \in \tau$ be an arbitrary open set and recall that, because $\mathcal{B}$ generates τ , there must be a family $\{B_x \in \mathcal{B}\}_{x \in U}$ such that each B_x satisfies $x \in B_x$ and $B_x \subseteq U$; consequently:

$$U = \bigcup_{x \in U} B_x.$$

Then, by assumption, for each B_x there is a $B'_x \in \mathcal{B}'$ such that $B'_x \subseteq B_x$ and $x \in B'_x$. Then, it follows that $B'_x \subseteq U$ and all elements $x \in U$ are accounted for, so that:

$$U = \bigcup_{x \in U} B'_x$$

which implies that $U \in \tau'$. Since this argument holds for every $U \in \tau$ we conclude that $\tau \subseteq \tau'$, completing the proof. $\square$

A Hausdorff space is a topological space where we may discern any two distinct points by studying their neighbourhoods.

Definition 2.2.4. A *Hausdorff space* is a topological space (X, τ) satisfying the Hausdorff separation condition; namely, for any two points $x, y \in X$:

$$x \neq y \implies \exists U, V \in \tau \text{ s.t. } x \in U, y \in V \text{ and } U \cap V = \emptyset. \quad (2.13)$$

Example 2.2.5. The collection of all open intervals (a, b) of real numbers is a base for the standard topology on $\mathbb{R}$. This topological space is Hausdorff: let $x, y \in \mathbb{R}$ and assume $x < y$; for $\epsilon = \frac{y-x}{2}$ the open neighbourhoods $(x - \epsilon, x + \epsilon)$ and $(y - \epsilon, y + \epsilon)$ are disjoint and, hence, the Hausdorff separation condition is satisfied.

Example 2.2.6. Every metric space is a Hausdorff space. Let (X, d) be a metric space. A ball centered at $x \in X$ with radius $\epsilon \in \mathbb{R}$ is a subset of X defined as follows:

$$B_{x,\epsilon} = \{y \in X \mid d(x, y) < \epsilon\}.$$

The collection of all balls for every choice of center and radius forms a base. This is a Hausdorff space: if $x \neq y$ choose $\epsilon = \frac{d(x,y)}{2}$, then the open neighbourhoods $B_{x,\epsilon}$ and $B_{y,\epsilon}$ are disjoint. Moreover, every normed vector space is a metric space with its distance function induced by the norm

$$d(u, v) = \|u - v\|$$

and, hence, every normed vector space is a Hausdorff space.

As it is usual whenever a mathematical structure is defined, it is useful to provide a notion of mapping between topological spaces that, in some sense, preserves the topological structure. Such is the role of continuous functions, defined below.

Definition 2.2.7. Let (X, τ_X) and (Y, τ_Y) be two topological spaces. A function $f: X \rightarrow Y$ is said to be continuous at point $x \in X$ if for every open neighbourhood V of $f(x)$ there is an open neighbourhood U of x such that $f(U) \subseteq V$.⁴ Furthermore, $f: X \rightarrow Y$ is said to be a *continuous function* if it is continuous at every point $x \in X$.

Example 2.2.8. Endow $\mathbb{R}$ with the standard topology, as in Example 2.2.5. The function $f: \mathbb{R} \rightarrow \mathbb{R}$ defined as $f(x) = 2x$ is continuous: for any x and any open set $V = (f(x) - a, f(x) + b)$ with $a, b > 0$, the open set $U = (x - \frac{a}{2}, x + \frac{b}{2})$ satisfies $f(U) \subseteq V$. In fact, the standard notion of a continuous function studied in analysis coincides with the topological notion when $\mathbb{R}$ is endowed with the standard topology.

Example 2.2.9. Let τ and τ' be two topologies on the same set X . If $\tau \subseteq \tau'$ then the identity function $(X, \tau') \rightarrow (X, \tau)$ is continuous. If the containment is strict $\tau \subset \tau'$ then the identity function on the opposite direction $(X, \tau) \rightarrow (X, \tau')$ cannot be continuous.

⁴The shorthand $f(U)$ refers to the set $\{f(u) \mid u \in U\}$.

Although this is not the usual definition of a continuous function given in most introductory texts on topology, it is equivalent to the standard one, as shown in Theorem 3.1 from [37]. The definition given in this text is a better fit for the arguments that will appear later in this chapter. Thanks to the following proposition, we may prove that a function is continuous by studying the open sets in the base of the codomain.

Proposition 2.2.10. *Let (X, τ_X) and (Y, τ_Y) be two topological spaces and let $\mathcal{B}$ be a base for τ_Y . A function $f: X \rightarrow Y$ is continuous at point $x \in X$ iff for all $B \in \mathcal{B}$ containing $f(x)$ there is an open neighbourhood U of x such that $f(U) \subseteq B$.*

Proof. If f is continuous at x and $B \in \mathcal{B}$ is an open neighbourhood of $f(x)$, the existence of the required open neighbourhood U is immediate. To prove the other direction, assume U exists for every B as stated in the claim. Recall that every open set in τ_Y is the union of a collection of elements in $\mathcal{B}$, hence, for every open neighbourhood V of $f(x)$ there is at least one element $B \in \mathcal{B}$ such that $B \subseteq V$ and $f(x) \in B$. By assumption, there is an open neighbourhood U of x such that $f(U) \subseteq B$ so we may conclude that $f(U) \subseteq V$, proving that f is continuous at x . $\square$

Proposition 2.2.11. *There is a category **Top** whose objects are topological spaces and whose morphisms are continuous functions. There is a full subcategory **Top**_{Haus} of **Top**, obtained by restricting objects to Hausdorff spaces.*

Proof. It is straightforward to prove that if $f: A \rightarrow B$ and $g: B \rightarrow C$ are continuous functions then $g \circ f$ is a continuous function as well. Identity functions between the same topological spaces are continuous (see Example 2.2.9). Associativity and identity axioms follow immediately from those of **Set**. These arguments are independent of whether objects are Hausdorff spaces or general topological spaces, thus, both **Top** and **Top**_{Haus} are categories, and it is straightforward to check that the embedding **Top**_{Haus} $\hookrightarrow$ **Top** is a full functor. $\square$

Whenever a set can be injectively mapped into a topological space, we may endow it with the subspace topology: the coarsest topology that makes the injection continuous.

Definition 2.2.12. Let (Y, τ) be a topological space, let X be an arbitrary set and let $f: X \rightarrow Y$ be an injective function. The *subspace topology* on X induced by f is defined as follows:

$$\tau_f = \{f^{-1}(V) \mid V \in \tau\}$$

where $f^{-1}(V) = \{x \in X \mid f(x) \in V\}$. It is trivial to check that $f: (X, \tau_f) \rightarrow (Y, \tau)$ is continuous.

It is straightforward to show that, if Y is Hausdorff, the subspace topology τ_f is also Hausdorff.

2.2.2 Nets and limit points (*Preamble*)

In this thesis, topology is used to discuss convergence so that, eventually, we may say that an infinite sum is defined if and only if it converges. The notion of convergence is often linked to sequences but, in the study of general topology, it is more advantageous to discuss it in terms of nets. Nets are a generalisation of sequences that are better behaved in the context of topological spaces, in the sense that there are important results (for instance, Proposition 2.2.19) which hold for nets but are not necessarily true for sequences.

Definition 2.2.13. Let D be a nonempty set and let $\leq$ be a partial order on it. The pair $(D, \leq)$ is said to be a *directed set* if for every pair of elements $a, b \in D$, there is an upper bound, *i.e.* there is an element $c \in D$ such that $a \leq c$ and $b \leq c$. A *net* is a function whose domain is a directed set. A net in X is a net whose codomain is X .

Example 2.2.14. Let $\{x_i\}_{i \in \mathbb{N}}$ be a family indexed by the natural numbers, *i.e.* a sequence. The order $\leq$ of natural numbers makes $(\mathbb{N}, \leq)$ a directed set and, hence, the function mapping each $i \in \mathbb{N}$ to x_i is a net.

Example 2.2.15. Let $(A, \leq)$ and $(B, \preceq)$ be two directed sets. Define $(A \times B, \sqsubseteq)$ to be the partially ordered set where $(a, b) \sqsubseteq (a', b')$ iff both $a \leq a'$ and $b \preceq b'$. It is straightforward to check that $(A \times B, \sqsubseteq)$ is a directed set, as its upper bounds can be defined with respect to those from $(A, \leq)$ and $(B, \preceq)$.

Example 2.2.16. Let X be a set and let $\mathbf{x} \in X^*$ be an arbitrary family. Let $\mathcal{F}(\mathbf{x})$ be the collection of all finite subfamilies of $\mathbf{x}$. Evidently, $(\mathcal{F}(\mathbf{x}), \subseteq)$ is a partially ordered set and, for any two $\mathbf{x}', \mathbf{x}'' \in \mathcal{F}(\mathbf{x})$, it is clear that $\mathbf{x}' \cup \mathbf{x}'' \in \mathcal{F}(\mathbf{x})$ so their union acts as their upper bound.

Let $\mathbf{x} \in \mathbb{R}^*$ be a family of real numbers. Let $(\mathcal{F}(\mathbf{x}), \subseteq)$ be the directed set defined in Example 2.2.16. There are multiple examples of nets in $\mathbb{R}$ with domain $\mathcal{F}(\mathbf{x})$, for instance, $\sigma(\mathbf{x}') = \sum \mathbf{x}'$ and $\mu(\mathbf{x}') = \prod \mathbf{x}'$. Importantly, every element $\mathbf{x}' \in \mathcal{F}(\mathbf{x})$ is finite by definition, so these sums and products are guaranteed to be well-defined.

It is time to define limit points, which formalise the notion of convergence.

Definition 2.2.17. Let $(D, \leq)$ be a directed set and let $d \in D$ be an arbitrary element. Let $(D|_d, \leq)$ be the directed set where:

$$D|_d = \{d' \in D \mid d \leq d'\}.$$

Let X be a topological space; a point $x \in X$ is a *limit point* of a net $\alpha: D \rightarrow X$, written

$$\alpha \rightarrow x$$

if and only if for each open neighbourhood U of x there is an element $d \in D$ such that $\alpha(D|_d) \subseteq U$.

If we are given a base for the topology, we are only required to check the open sets in the base to prove a net has a limit point.

Proposition 2.2.18. *Let $\alpha: D \rightarrow X$ be a net and let $\mathcal{B}$ be a base for a topology in X . A point $x \in X$ is a limit point of α iff for each $B \in \mathcal{B}$ such that $x \in B$ there is an element $d \in D$ satisfying that for all $\alpha(D|_d) \subseteq B$.*

Proof. If x is a limit point of α , the existence of the required $d \in D$ is guaranteed by definition for every open neighbourhood of x , including those in $\mathcal{B}$. To prove the other direction, recall that any open set in the topology is the union of a collection of elements in $\mathcal{B}$. Therefore, for any open neighbourhood U of x , there must be at least one element $B \in \mathcal{B}$ such that $x \in B$ and $B \subseteq U$. Take this B and assume we can obtain a $d \in D$ so that $\alpha(D|_d) \subseteq B$; then, trivially $\alpha(D|_d) \subseteq U$. We may do this for every open neighbourhood U of x so, indeed, $\alpha \rightarrow x$ as claimed. $\square$

The following proposition establishes that continuous functions are precisely the functions that preserve convergence.

Proposition 2.2.19. *Let X and Y be topological spaces and let $f: X \rightarrow Y$ be a function. The function f is continuous iff for all $x \in X$ and every net $\alpha: D \rightarrow X$:*

$$\alpha \rightarrow x \implies f \circ \alpha \rightarrow f(x).$$

Proof. Assume f is continuous. Let V be an arbitrary open neighbourhood of $f(x)$; because f is continuous, there is an open neighbourhood U of x such that $f(U) \subseteq V$. Let $\alpha \rightarrow x$ and take $d \in D$ so that $\alpha(D|_d) \subseteq U$; it follows that $f(\alpha(D|_d)) \subseteq V$ and so $f \circ \alpha \rightarrow f(x)$.

To prove the other direction, we show that if f is not continuous then there is a net α such that $\alpha \rightarrow x$ but $f(x)$ is not a limit point of $f \circ \alpha$. To define such a net,

let D_x be the set of open neighbourhoods of x ; then, $(D_x, \supseteq)$ is a directed set whose upper bounds are given by intersection. By assumption, f is not continuous at some $x \in X$, hence, for some open neighbourhood V of $f(x)$ all open neighbourhoods $U \in D_x$ contain a point $x_U \in U$ such that $f(x_U) \notin V$. Define a net $\alpha: D_x \rightarrow X$ assigning to each $U \in D_x$ such a point x_U . This net satisfies $\alpha \rightarrow x$ since, for any open neighbourhood U of x , it holds that $\alpha(D_x|_U) \subseteq U$ since every $U' \in D_x|_U$ is, by definition of $D_x|_U$, contained in U . However, for all $U \in D_x$ the definition of x_U makes it so that $(f \circ \alpha)(U) = f(x_U)$ is not in V ; thus, the open neighbourhood V prevents $f(x)$ from being a limit point of $f \circ \alpha$. Therefore, it is not satisfied that $\alpha \rightarrow x \implies f \circ \alpha \rightarrow f(x)$ and the proof by contrapositive is complete. $\square$

Finally, it is useful to define the notion of subnet.

Definition 2.2.20. Let $(A, \leq)$ and $(B, \preceq)$ be directed sets. A net $\beta: B \rightarrow X$ is a *subnet* of $\alpha: A \rightarrow X$ if there is a function $m: B \rightarrow A$ such that the following conditions are satisfied:

- the net β factors through α , i.e. $\beta = \alpha \circ m$;
- m is *isotone*, i.e. for all $b, b' \in B$, if $b \preceq b'$ then $m(b) \leq m(b')$;
- $m(B)$ is *cofinal* in A , i.e. for every $a \in A$ there is a $b \in B$ satisfying $a \leq m(b)$.

The first two conditions are straightforward: we are embedding the partial order of B into that of A while making sure that α and β match. The third condition prevents B from being a truncated version of A that omits all of A 's larger elements; this precaution is necessary so that subnets enjoy the following property.⁵

Proposition 2.2.21. Let $\alpha: A \rightarrow X$ and $\beta: B \rightarrow X$ be nets in a topological space X . If β is a subnet of α then:

$$\alpha \rightarrow x \implies \beta \rightarrow x.$$

Proof. Let $m: B \rightarrow A$ be a function witnessing that β is a subnet of α . Assume $\alpha \rightarrow x$, i.e. for every open neighbourhood U of x , there is an element $a \in A$ such that $\alpha(A|_a) \subseteq U$. Choose some $b \in B$ that satisfies $a \leq m(b)$; such an element is guaranteed to exist thanks to $m(B)$ being cofinal in A . Due to m being isotone we

⁵In Section 2.3 of [37] Kelley does not require the mediating function m to be isotone. However, this is a common requirement in many reference texts on topology (for instance, see Exercise 8 of the supplementary material from Chapter 3 of Munkres's book [48]) since it simplifies many results on limit points of subnets.

have that $m(B|_b) \subseteq A|_a$ and, hence, $\alpha(m(B|_b)) \subseteq U$. But $\beta = \alpha \circ m$ so $\beta(B|_b) \subseteq U$, implying $\beta \rightarrow x$ as claimed. $\square$

The proposition below characterises the limits in the subspace topology.

Proposition 2.2.22. *Let (Y, τ) be a topological space, let X be an arbitrary set and $f: X \rightarrow Y$ an injective function. Assign to X the subspace topology induced by f (see Definition 2.2.12); then, for any net $\alpha: D \rightarrow X$:*

$$\alpha \rightarrow x \iff f \circ \alpha \rightarrow f(x).$$

Proof. One direction is immediate from f being continuous by definition of the subspace topology. The other direction follows from the fact that, for every open neighbourhood $U \in \tau_f$ of x , there is an open set $V \in \tau$ such that $U = f^{-1}(V)$ by definition of the subspace topology. Then, if $f \circ \alpha \rightarrow f(x)$, there is an element $d \in D$ such that $f(\alpha(D|_d)) \subseteq V$; thus, $\alpha(D|_d) \subseteq f^{-1}(V)$, implying $\alpha \rightarrow x$. $\square$

In general, limit points need not be unique, but they are if the net is defined in a Hausdorff space, as established below.

Proposition 2.2.23. *Let α be a net in a Hausdorff space. Then, α has at most one limit point.*

Proof. Let $(D, \leq)$ be a directed set and X a Hausdorff space. Assume both $x, y \in X$ are limit points of $\alpha: D \rightarrow X$. Let U_x and U_y be open neighbourhoods of x and y respectively. Take $d_x \in D$ such that $\alpha(D|_{d_x}) \subseteq U_x$; do the same for U_y , obtaining $d_y \in D$. Because D is a directed set, we may take an upper bound of d_x and d_y , call it d . It follows that $\alpha(d)$ is both in U_x and U_y and, hence, $U_x \cap U_y \neq \emptyset$. This holds for any two open neighbourhoods of x and y so, according to the Hausdorff condition, they must be equal. $\square$

When working with a Hausdorff space X , the notation

$$\lim_{d \in D} \alpha(d) \simeq x$$

will be used to indicate that the net $\alpha: D \rightarrow X$ has x as its (unique) limit point, and we may use $\lim \alpha$ to refer to such a point.

2.2.3 Hausdorff commutative monoids

A Hausdorff monoid is a monoid whose underlying set comes equipped with a Hausdorff topology and whose monoid operation is continuous. Let $(X, +)$ be a monoid; since the monoid operation has the type $+: X \times X \rightarrow X$, to discuss continuity we must first assign a topology to $X \times X$.

Definition 2.2.24. Let $\{(X_i, \tau_i)\}_I$ be a collection of topological spaces. Let $\times_I X_i$ be the Cartesian product of the underlying sets and let $\mathcal{B}_\times$ be the collection of sets of the form $\times_I U_i$ such that $U_i \in \tau_i$ for each $i \in I$ and the subset $\{i \in I \mid U_i \neq X_i\}$ is finite. The *product topology* $\tau_\times$ is the one generated from the base $\mathcal{B}_\times$.

This definition makes the topological space $(\times_I X_i, \tau_\times)$ a categorical product in **Top** and in **Top_{Haus}**. In essence, this means that $\tau_\times$ is the coarsest topology that can be assigned to the Cartesian product while satisfying that its projections are continuous functions. Since $\times_I X_i$ is a categorical product in **Set**, every net $\alpha: D \rightarrow \times_I X_i$ is uniquely determined by a collection of nets $\{\alpha_i: D \rightarrow X_i\}_I$ obtained by composing α with the appropriate projection. Let $(\alpha_i)_I$ be a shorthand for a net α in $\times_I X_i$ and, similarly, let $(x_i)_I \in \times_I X_i$ denote a tuple of dimension $|I|$ where $x_i \in X_i$ for each $i \in I$.

Proposition 2.2.25. Let $\{(X_i, \tau_i)\}_I$ be an indexed set of topological spaces and let $(\alpha_i)_I: D \rightarrow \times_I X_i$ be an arbitrary net. Then,

$$(\alpha_i)_I \rightarrow (x_i)_I \iff \forall i \in I, \alpha_i \rightarrow x_i.$$

Proof. One direction follows from Proposition 2.2.19 since projections $\pi_i: \times_I X_i \rightarrow X_i$ are continuous. For the other direction assume that $\alpha_i \rightarrow x_i$ for all $i \in I$; this implies that for every open neighbourhood $U \in \tau_i$ of x_i there is an element $d_i \in D$ such that $\alpha_i(D|_{d_i}) \subseteq U$. Let $\times_I V_i$ be an open neighbourhood of $(x_i)_I$ in the base $\mathcal{B}_\times$ (see Definition 2.2.24) and define the following subset of I :

$$J = \{i \in I \mid V_i \neq X_i\}$$

which is guaranteed to be finite due to the definition of $\mathcal{B}_\times$. For each $j \in J$ obtain an element $d_j \in D$ such that $\alpha_j(D|_{d_j}) \subseteq V_j$ and let $\hat{d}$ be an upper bound for the collection $\{d_j\}_J$ — such an upper bound exists because J is finite. Clearly, it holds that $\alpha_i(D|_{\hat{d}}) \subseteq V_i$ for all $i \in I$, either because $i \in J$ or because $V_i = X_i$; hence, $(\alpha_i(D|_{\hat{d}}))_I \subseteq \times_I V_i$. This applies to every open neighbourhood of $(x_i)_I$ in the base $\mathcal{B}_\times$ so that Proposition 2.2.18 implies that $(\alpha_i)_I \rightarrow (x_i)_I$, concluding the proof. $\square$

In particular, this allows us to characterise continuous function $X \times Y \rightarrow Z$ as limit-preserving functions on both coordinates.

Proposition 2.2.26. *Let X, Y and Z be topological spaces and let $f: X \times Y \rightarrow Z$ be a function; then, f is continuous iff for every net $(\alpha, \beta): D \rightarrow X \times Y$ the following is satisfied:*

$$\alpha \rightarrow x \text{ and } \beta \rightarrow y \implies f(\alpha, \beta) \rightarrow f(x, y). \quad (2.14)$$

Proof. According to Proposition 2.2.19, f is continuous if and only if for every net $(\alpha, \beta): D \rightarrow X \times Y$ the following is satisfied:

$$(\alpha, \beta) \rightarrow (x, y) \implies f \circ (\alpha, \beta) \rightarrow f(x, y). \quad (2.15)$$

Assume $\alpha \rightarrow x$ and $\beta \rightarrow y$; then, Proposition 2.2.25 implies that $(\alpha, \beta) \rightarrow (x, y)$ so, if f is continuous, $f \circ (\alpha, \beta) \rightarrow f(x, y)$ and the implication (2.14) is satisfied. Conversely, assume $(\alpha, \beta) \rightarrow (x, y)$ and recall that Proposition 2.2.25 works in both directions, so it implies $\alpha \rightarrow x$ and $\beta \rightarrow y$. Then, if (2.14) is satisfied, it imposes $f \circ (\alpha, \beta) \rightarrow f(x, y)$; therefore, (2.15) holds for every net (α, β) in $X \times Y$, implying that f is continuous. $\square$

Finally, we have all the ingredients to define topological monoids and, in particular, Hausdorff commutative monoids.

Definition 2.2.27. Let (X, τ) be a topological space and let $(X, +)$ be a monoid. We say $(X, \tau, +)$ is a *topological monoid* if $+: X \times X \rightarrow X$ is continuous. A *Hausdorff commutative monoid* is a topological monoid where (X, τ) is Hausdorff and $(X, +)$ is commutative. A *Hausdorff abelian group* is a Hausdorff commutative monoid $(X, \tau, +)$ where $(X, +)$ is an abelian group and the inversion map $x \mapsto -x$ is continuous.

Example 2.2.28. The set $\mathbb{R}$ of real numbers with the standard topology (see Example 2.2.5) and the standard addition forms a Hausdorff commutative monoid — in fact, a Hausdorff abelian group.

Proposition 2.2.29. *Every normed vector space is a Hausdorff abelian group.*

Proof. As discussed in Example 2.2.6, every normed vector space V is a Hausdorff space and, by virtue of being a vector space, it is an abelian group with respect to addition. For every pair of vectors $v, u \in V$ and every open neighbourhood U of $v + u$ there is some $\epsilon > 0$ such that $B_{v+u, \epsilon} \subseteq U$ where $B_{v+u, \epsilon}$ is the ball of radius ϵ

and center $v + u$. It follows from the triangle inequality of the norm that every pair of vectors $v' \in B_{v, \frac{\epsilon}{2}}$ and $u' \in B_{u, \frac{\epsilon}{2}}$ satisfies $v' + u' \in B_{v+u, \epsilon}$ and, consequently, $+(B_{v, \frac{\epsilon}{2}}, B_{u, \frac{\epsilon}{2}}) \subseteq U$, implying that $+$ is continuous. The fact that the inversion map is continuous can be shown by similar means, completing the proof. $\square$

Definition 2.2.30. Let **HausCMon** be the category whose objects are Hausdorff commutative monoids and whose morphisms are continuous monoid homomorphisms. Let **HausAb** be the full subcategory of **HausCMon** whose objects are restricted to Hausdorff abelian groups.

It is well known that **CMon**, **Ab** and **Top_{Haus}** are complete categories and the forgetful functor from each of these to **Set** preserves limits. This can be used to show that **HausCMon** and **HausAb** are complete categories, as established below.

Proposition 2.2.31. *Both **HausCMon** and **HausAb** are complete categories.*

Proof. First, we show that **HausCMon** has equalizers. For every pair of Hausdorff commutative monoids $(X, \tau_X, +_X)$ and $(Y, \tau_Y, +_Y)$ and every pair of morphisms $f, g \in \mathbf{HausCMon}(X, Y)$, define the subset $E = \{x \in X \mid f(x) = g(x)\}$ with inclusion $e: E \hookrightarrow X$ and let τ_e be the subspace topology:

$$\tau_e = \{e^{-1}(U) \mid U \in \tau_X\} = \{U \cap E \mid U \in \tau_X\}.$$

It is generally known — and a simple exercise to show — that (E, τ_e) along with the continuous function e is an equalizer of f and g in **Top_{Haus}**. Let $+_E: E \times E \rightarrow E$ be the restriction of $+_X$ to inputs in E ; again, it is general knowledge that $(E, +_E)$ along with the monoid homomorphism e is an equalizer of f and g in **CMon**. Consequently, $e: E \rightarrow X$ is a continuous monoid homomorphism and $(E, \tau_e, +_E)$ is both a Hausdorff space and a commutative monoid; to show that this E is a Hausdorff commutative monoid we must prove that $+_E$ is continuous. Recall that $+_X$ is continuous by definition, so for any $x, x' \in E$ and any open neighbourhood $V \in \tau_X$ of $x + x'$, we may find $U, W \in \tau_X$ such that $+_X(U, W) \subseteq V$; notice that $+_E(U \cap E, W \cap E) \subseteq V \cap E$ and every open set in τ_e is of the form $- \cap E$, so $+_E$ is continuous and $E \in \mathbf{HausCMon}$. Since E is the equalizer of f and g in **Set**, it follows that E is a cone in **HausCMon**. Moreover, for any other cone $A \in \mathbf{HausCMon}$, the morphism $h: A \rightarrow X$ will factor through the unique function $m: A \rightarrow E$ due to the fact that E is an equalizer in **Set**. It only remains to show that such a unique function m is a morphism in **HausCMon**; this is immediate since m is the mediating morphism both in **Top_{Haus}** and **CMon**, implying that m is a continuous monoid homomorphism. In conclusion, we have shown

that that **HausCMon** has equalizers. To prove that **HausAb** has equalizers as well we simply need to show that the inversion map in $(E, \tau_e, +_E)$ is continuous; this follows immediately from the corresponding property in the Hausdorff abelian group X .

Next, we show that **HausCMon** has small products. Let $\{(X_i, \tau_i, +_i)\}_I$ be a small collection of objects in **HausCMon**. Define $+_\times$ by applying the corresponding $+_i$ to each coordinate; $(\times_I X_i, +_i)$ is a small product in **CMon**. Define the product topology $\tau_\times$ as in Definition 2.2.24; it is well-known that $(\times_I X_i, \tau_\times)$ is a small product in **Top_{Haus}**. It follows that projections are continuous monoid homomorphisms and we must show that $+_\times$ is continuous. Every open neighbourhood in the base $\mathcal{B}_\times$ for the product topology is of the form $\times_I V_i$ where $V_i \in \tau_i$ for each $i \in I$; consequently, we may find $U_i, U'_i \in \tau_i$ so that $+_i(U_i, U'_i) \subseteq V_i$ for each $i \in I$ due to each $+_i$ being continuous. Then, it is immediate that the condition from Proposition 2.2.10 is satisfied, implying that $+_\times$ is continuous. Thus, it has been shown that $(\times_I X_i, \tau_\times, +_\times)$ is a Hausdorff commutative monoid and, evidently, a cone of the discrete diagram in **HausCMon**; it remains to show that it is a limit. For any other cone $A \in \mathbf{HausCMon}$, each morphism $h_i: A \rightarrow X_i$ will factor through the unique function $m: A \rightarrow \times_I X_i$ due to the fact that $\times_I X_i$ is a categorical product in **Set**. It only remains to show that such a unique function m is a morphism in **HausCMon**; this is immediate since m is the mediating morphism both in **Top_{Haus}** and **CMon**, implying that m is a continuous monoid homomorphism. In conclusion, we have shown **HausCMon** has small products. To prove that **HausAb** has small products as well we simply need to show that the inversion map in $(\times_I X_i, \tau_\times, +_\times)$ is continuous; this follows immediately from the corresponding property in each Hausdorff abelian group X_i .

A category with equalizers and small products is complete, so both **HausCMon** and **HausAb** are complete categories, as claimed. $\square$

As promised in Section 2.1, every Hausdorff commutative monoid $(X, \tau, +)$ can be given a partial function $\Sigma: X^* \rightarrow X$ such that (X, Σ) is a finitely total Σ -monoid whose summable families are precisely those that converge according to the topology.

Definition 2.2.32. Let $(X, \tau, +)$ be a Hausdorff commutative monoid. For any family $\mathbf{x} \in X^*$, let $(\mathcal{F}(\mathbf{x}), \subseteq)$ be the directed set of finite subfamilies of $\mathbf{x}$ (see Example 2.2.16) and let $\sigma: \mathcal{F}(\mathbf{x}) \rightarrow X$ be the net that maps each finite subfamily $\mathbf{x}' \in \mathcal{F}(\mathbf{x})$ to the sum of its elements, obtained after a finite number of applications of $+$. The ex-

tended monoid operation $\Sigma: X^* \rightarrow X$ is defined as follows:

$$\Sigma \mathbf{x} = \begin{cases} x & \text{if } \lim \sigma \simeq x \\ \text{undefined} & \text{otherwise.} \end{cases}$$

We require $(X, +)$ to be a commutative monoid so that associativity and commutativity guarantee that the net σ is well-defined: otherwise the result would depend on the arbitrary order in which the elements from the finite family are combined. The space is required to be Hausdorff so that when σ converges the result is unique.⁶ The following lemmas prove the axioms of finitely total Σ -monoids for the extended monoid operation.

Lemma 2.2.33. *Let $(X, \tau, +)$ be a Hausdorff commutative monoid and let Σ be its extended monoid operation. If $\mathbf{x} \in X^*$ is a finite family, then $\Sigma \mathbf{x}$ is defined and it equals the sum of its elements, obtained after a finite number of applications of $+$.*

Proof. Let $\sigma: \mathcal{F}(\mathbf{x}) \rightarrow X$ be the net that maps each finite subfamily $\mathbf{x}' \in \mathcal{F}(\mathbf{x})$ to the sum of its elements, obtained after a finite number of applications of $+$. When $\mathbf{x}$ is finite it is the largest element in $\mathcal{F}(\mathbf{x})$. Thus, $\mathcal{F}(\mathbf{x})|_{\mathbf{x}} = \{\mathbf{x}\}$ and $\sigma(\mathbf{x})$ trivially satisfies being the limit point of σ , regardless of the topology. By definition, $\Sigma \mathbf{x} \simeq \lim \sigma$ so we conclude that $\Sigma \mathbf{x} \simeq \sigma(\mathbf{x})$, as claimed. $\square$

Lemma 2.2.34. *Let $(X, \tau, +)$ be a Hausdorff commutative monoid and let Σ be its extended monoid operation. Let $\mathbf{x} \in X^*$ be an arbitrary family and let $\mathbf{x}_\emptyset$ be its subfamily of elements other than 0; if $\mathbf{x}$ is summable then $\mathbf{x}_\emptyset$ is summable as well.*

Proof. Let $\sigma: \mathcal{F}(\mathbf{x}) \rightarrow X$ be the net of finite partial sums of $\mathbf{x}$ and let $\sigma_\emptyset: \mathcal{F}(\mathbf{x}_\emptyset) \rightarrow X$ be the corresponding net for the family $\mathbf{x}_\emptyset$. If $\Sigma \mathbf{x} \simeq x$ it follows from the definition of Σ that $\lim \sigma \simeq x$, i.e. for every open neighbourhood U of x there is a finite subfamily $\mathbf{x}' \in \mathcal{F}(\mathbf{x})$ such that $\sigma(\mathcal{F}(\mathbf{x})|_{\mathbf{x}'}) \subseteq U$. Then, it is straightforward to check that $\sigma_\emptyset(\mathcal{F}(\mathbf{x}_\emptyset)|_{\mathbf{x}'_\emptyset}) \subseteq U$ as well since for all $\mathbf{x}' \in \mathcal{F}(\mathbf{x})$ we have that $\sigma(\mathbf{x}') = \sigma_\emptyset(\mathbf{x}'_\emptyset)$ due to 0 being the neutral element of $+$. $\square$

Lemma 2.2.35. *Let $(X, \tau, +)$ be a Hausdorff commutative monoid and let Σ be its extended monoid operation. Let $\{\mathbf{x}_j\}_J$ be a collection of summable families where $|J| < \infty$ and let $\mathbf{x} = \uplus_J \mathbf{x}_j$; then:*

$$\Sigma\{\Sigma \mathbf{x}_j\}_J \simeq x \implies \Sigma \mathbf{x} \simeq x. \quad (2.16)$$

⁶A topological space is Hausdorff iff each net in the space converges to at most one point; see Chapter 2, Theorem 3 from Kelley's book [37].

Proof. Let $\sigma: \mathcal{F}(\mathbf{x}) \rightarrow X$ be the net that maps each finite family $\mathbf{x}' \in \mathcal{F}(\mathbf{x})$ to the sum of its elements. Similarly, let $\sigma_j: \mathcal{F}(\mathbf{x}_j) \rightarrow X$ be the corresponding net for each $j \in J$; the claim assumes that each $\mathbf{x}_j$ is summable, hence, the limit point $\lim \sigma_j$ exists for each $j \in J$. Assume that $J = \{0, 1\}$; the general case will follow straightforwardly. In this case, the implication (2.16) simplifies to:

$$\lim \sigma_0 + \lim \sigma_1 \simeq x \implies \lim \sigma \simeq x.$$

Assume that, indeed, $\lim \sigma_0 + \lim \sigma_1 \simeq x$; unfortunately, we cannot apply continuity of $+$ since σ_0 and σ_1 are defined with respect to different directed sets. Define a function $m: \mathcal{F}(\mathbf{x}) \rightarrow \mathcal{F}(\mathbf{x}_0)$ given by $m(\mathbf{x}') = \mathbf{x}' \cap \mathbf{x}_0$; clearly, m is isotone and, for each $\mathbf{y} \in \mathcal{F}(\mathbf{x}_0)$ it is trivially satisfied that $\mathbf{y} \subseteq m(\mathbf{y})$. Thus, the net $\hat{\sigma}_0: \mathcal{F}(\mathbf{x}) \rightarrow X$ given by $\hat{\sigma}_0 = \sigma_0 \circ m_0$ is a subnet of σ_0 and Proposition 2.2.21 implies that $\lim \hat{\sigma}_0 \simeq \lim \sigma_0$. We may define $\hat{\sigma}_1$ in the same manner and it is immediate that:

$$\lim \hat{\sigma}_0 + \lim \hat{\sigma}_1 \simeq x.$$

Since both $\hat{\sigma}_0$ and $\hat{\sigma}_1$ are now defined on the same directed set $\mathcal{F}(\mathbf{x})$, Proposition 2.2.26 implies that $\lim (\hat{\sigma}_0 + \hat{\sigma}_1) \simeq x$. It is straightforward to check that, due to associativity and commutativity of $+$, the nets σ and $\hat{\sigma}_0 + \hat{\sigma}_1$ are, in fact, the same function. Consequently, it has been shown that $\lim \sigma \simeq x$; this establishes the claim in the case that $|J| = 2$.

If J is empty or it is a singleton set, the claim follows immediately. If $|J| > 2$ we may repeat the argument above as many times as necessary to aggregate all subfamilies; this process will eventually be completed since the claim assumes J is finite. $\square$

Lemma 2.2.36. *Let $(X, \tau, +)$ be a Hausdorff commutative monoid and let Σ be its extended monoid operation. Let $\{\mathbf{x}_j\}_J$ be a collection of finite families and let $\mathbf{x} = \uplus_J \mathbf{x}_j$; then:*

$$\Sigma \mathbf{x} \simeq x \implies \Sigma \{\Sigma \mathbf{x}_j\}_J \simeq x. \quad (2.17)$$

Proof. Since each family $\mathbf{x}_j$ is finite, Lemma 2.2.33 establishes that $\Sigma \mathbf{x}_j \simeq x_j$ for some $x_j \in X$. Let $\hat{\sigma}: \mathcal{F}(J) \rightarrow X$ be the net that maps each finite subset $J' \in \mathcal{F}(J)$ to the sum of the finite family $\{x_j\}_{J'}$. Notice that, by definition, $\Sigma \{\Sigma \mathbf{x}_j\}_J \simeq \lim \hat{\sigma}$ so our goal is to show that $\Sigma \mathbf{x} \simeq x$ implies $\lim \hat{\sigma} \simeq x$.

Assume that $\Sigma \mathbf{x} \simeq x$ and let $\sigma: \mathcal{F}(\mathbf{x}) \rightarrow X$ be the net that maps each finite subfamily $\mathbf{x}' \in \mathcal{F}(\mathbf{x})$ to the sum of its elements; then, $\lim \sigma \simeq x$. Define a function $m: \mathcal{F}(J) \rightarrow \mathcal{F}(\mathbf{x})$ as follows:

$$m(J') = \uplus_{J'} \mathbf{x}_j.$$

The result of $m(J')$ is a finite family since J' is finite and the claim assumes that each $\mathbf{x}_j$ is a finite family. It is evident that m is isotone and, for each finite subfamily $\mathbf{x}' \in \mathcal{F}(\mathbf{x})$ we may define a finite subset $\hat{J} \in \mathcal{F}(J)$ as follows:

$$\hat{J} = \{j \in J \mid \mathbf{x}' \cap \mathbf{x}_j \neq \emptyset\}$$

so that $\mathbf{x}' \subseteq m(\hat{J})$ is trivially satisfied. Moreover, notice that $\hat{\sigma} = \sigma \circ m$; since all families involved are finite the identity follows from associativity and commutativity of $+$. Consequently, $\hat{\sigma}$ is a subnet of σ and Proposition 2.2.21 implies that $\lim \hat{\sigma} \simeq x$. As discussed above, this means that $\Sigma\{\Sigma\mathbf{x}_j\}_J \simeq x$, completing the proof. $\square$

Proposition 2.2.37. *Let $(X, \tau, +)$ be a Hausdorff commutative monoid and let Σ be its extended monoid operation; then, (X, Σ) is a finitely total Σ -monoid.*

Proof. Every finite family is summable according to Lemma 2.2.33. In particular, the empty family is summable and we have that $\Sigma\{x, \Sigma\emptyset\} \simeq x + \Sigma\emptyset$. Moreover, according to Proposition 2.1.2, we have $\Sigma\{x, \Sigma\emptyset\} \simeq x$ for all $x \in X$. If we combine these two results and let $x = 0$ we obtain $0 + \Sigma\emptyset = 0$ which, due to $(X, +, 0)$ being a monoid, results in $\Sigma\emptyset = 0$. Thus, the neutral element of the monoid is the neutral element of the Σ -monoid and Lemma 2.2.34 completes the proof of the neutral element axiom. The singleton axiom follows trivially from Lemma 2.2.33, flattening is proven in Lemma 2.2.35 and bracketing is proven in Lemma 2.2.36. $\square$

Remark 2.2.38. Since Hausdorff abelian groups are a subclass of Hausdorff commutative monoids that have additive inverses, it is immediate from Proposition 2.1.12 that Hausdorff commutative monoids need not satisfy strong flattening. The question of whether or not Hausdorff commutative monoids satisfy strong bracketing is more subtle. The main obstacle is that, at some point, we must prove that for some net $\sigma: \mathcal{F}(J) \times \mathcal{F}(\mathbf{x}) \rightarrow X$ the following implication holds:

$$\lim_{\mathcal{F}(J) \times \mathcal{F}(\mathbf{x})} \sigma \simeq x \implies \lim_{J' \in \mathcal{F}(J)} \left(\lim_{\mathcal{F}(\mathbf{x})} \sigma(J', -) \right) \simeq x.$$

In essence, we need to obtain the limit of a net $\hat{\sigma}: \mathcal{F}(J) \rightarrow X$ that maps each finite subset $J' \in \mathcal{F}(J)$ to the limit point $\lim \sigma(J', -)$. Unfortunately, since in the case of strong bracketing the families $\mathbf{x}_j$ are allowed to be infinite, the argument via the construction of a subnet from the proof of Lemma 2.2.36 cannot be used — the function $m(J') = \uplus_{j \in J'} \mathbf{x}_j$ need not yield families in $\mathcal{F}(\mathbf{x})$. On the other hand, even though $\lim_{\mathcal{F}(J) \times \mathcal{F}(\mathbf{x})} \sigma \simeq x$ guarantees that for every open neighbourhood U of x we

may find $(J', \mathbf{x}')$ such that $\sigma(\mathcal{F}(J) \times \mathcal{F}(\mathbf{x})|_{(J', \mathbf{x}')} \subseteq U$, this does not imply that $\lim \sigma(J', -) \in U$ since U need not be closed. In fact, proving that an iterated limit $\lim_A \lim_B \alpha$ exists and coincides with the double limit $\lim_{A \times B} \alpha$ is generally a subtle matter. Theorem 4 from Chapter 2 in [37] shows that if $\lim_A \lim_B \alpha$ exists we may define a directed set $A \times B^A$ such that

$$\lim_{(a,f) \in A \times B^A} \alpha(a, f(a)) \simeq \lim_{a \in A} \lim_{b \in B} \alpha(a, b)$$

but, in order to get from the left hand side to $\lim_{A \times B} \alpha$, it appears we would need that for each open neighbourhood U of $\lim_A \lim_B \alpha$ the (possibly infinite) set $\{f_U(a') \mid a' \geq a_U\}$ had an upper bound — where $(a_U, f_U) \in A \times B^A$ is the point such that all $(a', f') \geq (a_U, f_U)$ satisfy $\alpha(a', f'(a')) \in U$.

Continuous monoid homomorphisms between Hausdorff commutative monoids become Σ -homomorphisms between the Σ -monoids induced by their extended monoid operation.

Lemma 2.2.39. *Let X and Y be Hausdorff commutative monoids and let $f: X \rightarrow Y$ be a continuous monoid homomorphism. Then, for every $\mathbf{x} \in X^*$ and $x \in X$:*

$$\Sigma \mathbf{x} \simeq x \implies \Sigma f \mathbf{x} \simeq f(x).$$

Proof. Let $\sigma: \mathcal{F}(\mathbf{x}) \rightarrow X$ be the net of finite partial sums of $\mathbf{x}$ and assume that $\Sigma \mathbf{x} \simeq x$. Then $\lim \sigma \simeq x$ and, due to f being continuous, Proposition 2.2.19 implies that $\lim (f \circ \sigma) \simeq f(x)$. Considering that f is a monoid homomorphism, it is straightforward to check that $f \circ \sigma$ is equivalent to the net assigned to $f \mathbf{x}$. It follows that $\Sigma f \mathbf{x} \simeq f(x)$, proving the claim. $\square$

Definition 2.2.40. There is a faithful functor $G: \mathbf{HausCMon} \rightarrow \Sigma_{\text{ft}}$ that equips each Hausdorff commutative monoid with the extended monoid operation from Definition 2.2.32 and acts as the identity on morphisms.

Clearly, G is faithful: f and $G(f)$ are the same function, so it is obvious that $f \neq g$ implies $G(f) \neq G(g)$. However, G is not full as illustrated by the following example.

Example 2.2.41. Let $A \in \mathbf{HausCMon}$ with underlying set $\{0\} \cup [1, \infty)$, discrete topology and the standard addition over the real numbers. Let $B \in \mathbf{HausCMon}$ with the same underlying set $\{0\} \cup [1, \infty)$ along with standard addition, but whose topology is given by the base of open sets comprised of the singleton $\{0\}$ along with every interval $[x, x + \epsilon)$ for each $x \in [1, \infty]$ and each $\epsilon > 0$.

Notice that Σ on both $G(A)$ and $G(B)$ is only defined on families with a finite number of nonzero elements; the reason being that all nonzero elements are ≥ 1 so infinite sums of these always diverge. Since both $G(A)$ and $G(B)$ have the same summable families, it is immediate that $G(A) = G(B)$. Then, if we consider the identity function on the underlying sets $\text{id}: B \rightarrow A$ we get that:

- $\text{id}: G(B) \rightarrow G(A)$ is a valid morphism in Σ_{ft} (indeed, it is the identity morphism);
- since G acts as the identity on morphisms, if the morphism $\text{id}: G(B) \rightarrow G(A)$ were to be in the image of the map $\mathbf{HausCMon}(B, A) \rightarrow \Sigma_{\text{ft}}(G(B), G(A))$, we would need $\text{id}: B \rightarrow A$ to be continuous;
- however, $\text{id}: B \rightarrow A$ is not continuous since the topology of B is strictly coarser than that of A .

Therefore, $G: \mathbf{HausCMon} \rightarrow \Sigma_{\text{ft}}$ is not full.

Similarly, there is a faithful functor $\mathbf{HausAb} \rightarrow \Sigma_{\text{g}}$ since the requirement that the inversion map is continuous implies that it is a Σ -homomorphism; we expect this functor not to be full either. The following subsection establishes that both this functor and $\mathbf{HausCMon} \rightarrow \Sigma_{\text{ft}}$ have a left adjoint. But first, the following remark briefly discusses whether $\mathbf{HausAb}$ -enriched categories are well-defined and what their relevance is with respect to categories in quantum quantum computer science.

Remark 2.2.42. Every Hilbert space is a Hausdorff abelian group and $(\mathbf{FdHilb}, \otimes, I)$ is a closed symmetric monoidal category, so its hom-sets are Hilbert spaces and, consequently, $\mathbf{FdHilb}$ could potentially be perceived as a $\mathbf{HausAb}$ -enriched category. This is possible because the standard tensor product of vector spaces — when applied to *finite*-dimensional Hilbert spaces — is already complete. However, the matter becomes more subtle when we consider the hom-sets of $\mathbf{Hilb}$, since these are no longer Hilbert spaces, even though they are Hausdorff abelian groups using the strong operator topology from Definition 3.2.22. However, tensor products in $\mathbf{HausAb}$ are a very subtle matter [22] and it is well-known [24] that there is no tensor product $\otimes$ for which $(\mathbf{HausAb}, \otimes, I)$ is a monoidal category, implying that $\mathbf{HausAb}$ -enriched categories are not well-defined.

Furthermore, in the case of $\mathbf{FdContraction}$, hom-sets are not even (total) monoids, so $\mathbf{HausCMon}$ -enriched categories (even if they were well-defined) would be too

restrictive for our purposes. Fortunately, the fact that every $(\Sigma_*, \otimes, I)$ category is monoidal (see Section 2.1.5) along with the functor $\mathbf{HausAb} \rightarrow \Sigma_g$ can be used to realise $\mathbf{Hilb}$ as a Σ_g -enriched category. Then, due to Lemma 2.1.8, the hom-sets of $\mathbf{Contraction}$ — which are restrictions of the hom-sets of $\mathbf{Hilb}$ — are weak Σ -monoids, realising $\mathbf{Contraction}$ as a Σ_w -enriched category. Consequently, Chapter 3 will deal with Σ_* -enriched categories whose Σ function on hom-sets is defined in terms of the extended monoid operation of a Hausdorff abelian group.

2.2.4 The left adjoint to $\mathbf{HausCMon} \rightarrow \Sigma_{ft}$

The functor $G: \mathbf{HausCMon} \rightarrow \Sigma_{ft}$ described in the previous section has a left adjoint. An explicit construction has not been achieved; instead, the proof uses the general adjoint functor theorem.

Lemma 2.2.43. *The functor $G: \mathbf{HausCMon} \rightarrow \Sigma_{ft}$ preserves limits.*

Proof. First, we show that G preserves equalizers. Let X and Y be two Hausdorff commutative monoids and let $f, g \in \mathbf{HausCMon}(X, Y)$; the equalizer of f and g in $\mathbf{HausCMon}$ is defined in Proposition 2.2.31. Its underlying set $E = \{x \in X \mid f(x) = g(x)\}$ is the same as that of the equalizer of $G(f)$ and $G(g)$ in Σ_{ft} (see Proposition 2.1.10). Moreover, the extended monoid operation induced by the topology assigned to $E \in \mathbf{HausCMon}$ coincides with the Σ function from the equalizer in Σ_{ft} . To show this, recall that E is endowed with the subspace topology induced by $e: E \hookrightarrow X$ so, according to Proposition 2.2.22, every family $\mathbf{x} \in E^*$ whose net of finite partial sums is denoted $\sigma: \mathcal{F}(\mathbf{x}) \rightarrow E$ satisfies:

$$\lim \sigma \simeq x \iff \lim e \circ \sigma \simeq e(x)$$

Since e is a monoid homomorphism it follows from the definition of the extended monoid operation that the partial function $\Sigma': E^* \rightarrow E$ in $G(E)$ satisfies

$$\Sigma' \mathbf{x} \simeq x \iff \Sigma e \mathbf{x} = e(x)$$

where Σ is the extended monoid operation assigned to $G(X)$. The latter two-way implication uniquely characterises the Σ function (2.4) of the equalizer in Σ_{ft} . Thus, both constructions yield the same Σ -monoid, implying G preserves equalizers.

Similarly, the categorical product in $\mathbf{HausCMon}$ of a small collection of objects $\{X_i \in \mathbf{HausCMon}\}_I$ and the categorical product of $\{G(X_i)\}_I$ in Σ_{ft} coincide in

their underlying sets and projections. To prove that G preserves small products, it suffices to show that the extended monoid operation assigned by G to the product in **HausCMon** coincides with the Σ function of the product in Σ_{ft} . Let $\{x_i \in X_i^*\}$ be a collection of families and let $\sigma_i: \mathcal{F}(\mathbf{x}_i) \rightarrow X_i$ be their corresponding net of finite partial sums; Proposition 2.2.25 establishes that:

$$\lim (\sigma_i)_I \simeq (x_i)_I \iff \forall i \in I, \lim \sigma_i \simeq x_i.$$

Since addition in $\times_I X_i$ is defined coordinate-wise, it follows from the definition of the extended monoid operation that the partial function Σ' in $G(\times_I X_i)$ satisfies

$$\Sigma'(\mathbf{x}_i)_I \simeq (x_i)_I \iff \forall i \in I, \Sigma^i \mathbf{x}_i \simeq x_i$$

where Σ^i is the extended monoid operation assigned to each $G(X_i)$. The latter two-way implication uniquely characterises the Σ function (2.5) of the categorical product in Σ_{ft} . Thus, both constructions yield the same Σ -monoid, implying G preserves small products.

A functor that preserves equalizers and small products and whose domain is a complete category automatically preserves all limits. Therefore, G preserves limits. $\square$

Proposition 2.2.44. *There is a left adjoint to the functor $G: \mathbf{HausCMon} \rightarrow \Sigma_{\text{ft}}$.*

Proof. It is immediate that **HausCMon** is locally small since there is a faithful forgetful functor $\mathbf{HausCMon} \rightarrow \mathbf{Set}$. According to Proposition 2.2.31, the category **HausCMon** is complete and Lemma 2.2.43 establishes that the functor G preserves limits. It remains to show that, for any $X \in \Sigma_{\text{ft}}$, the comma category $(X \Rightarrow G)$ has a weakly initial set; then, the existence of a left adjoint will follow from the general adjoint functor theorem (see Theorem 1.4.36). Let S be the collection of all Σ -homomorphisms $X \xrightarrow{q} G(A)$ where A may be any Hausdorff commutative monoid on a quotient of X . Notice that S is small since all quantifiers in its definition are with respect to a fixed set X .

Let Y be an arbitrary Hausdorff commutative monoid and let $f: X \rightarrow G(Y)$ be a Σ -homomorphism. Recall that every function can be factored through its image as follows:

$$f = X \xrightarrow{q} X/\sim \xrightarrow{\bar{f}} \text{im}(f) \xrightarrow{u} G(Y)$$

where $x \sim x'$ iff $f(x) = f(x')$; moreover, q is surjective, $\bar{f}$ is bijective and u is injective. Recall that $G(Y)$ has the same underlying set as Y , so $\text{im}(f) \subseteq Y$ and, thanks to u and $\bar{f}$ being injective, we may define a subspace topology τ_Σ on $X/\sim$ induced by

$u\bar{f}$. The composite $u\bar{f}: X/\sim \rightarrow Y$ is continuous and $(X/\sim, \tau_\Sigma)$ is Hausdorff. Notice that $\sim$ satisfies the following for every $a, a', b, b' \in X$:

$$a \sim a' \text{ and } b \sim b' \implies \Sigma\{a, b\} \sim \Sigma\{a', b'\}.$$

due to f being a Σ -homomorphism and X being finitely total:

$$f(\Sigma\{a, b\}) = \Sigma\{f(a), f(b)\} = \Sigma\{f(a'), f(b')\} = f(\Sigma\{a', b'\}).$$

Therefore, the operation $+_\Sigma: X/\sim \times X/\sim \rightarrow X/\sim$ defined as follows:

$$[a] +_\Sigma [b] = [\Sigma\{a, b\}]$$

is well-defined and it is trivial to show that it forms a commutative monoid, with the equivalence class $[0]$ being its neutral element. Moreover, the composite $u\bar{f}: X/\sim \rightarrow Y$ is a monoid homomorphism:

$$u\bar{f}([a] +_\Sigma [b]) = f(\Sigma\{a, b\}) = \Sigma\{f(a), f(b)\} = f(a) + f(b) = u\bar{f}[a] + u\bar{f}[b]$$

where $+$ is the monoid operation from Y and $\Sigma\{f(a), f(b)\} = f(a) + f(b)$ by definition of the extended monoid operation from Y . To prove that $+_\Sigma$ is continuous first notice that for every two nets $\alpha, \beta: D \rightarrow X$ the following is satisfied:

$$\begin{aligned} \lim q\alpha &\simeq [a] \text{ and } \lim q\beta \simeq [b] \\ \implies \lim f\alpha &\simeq f(a) \text{ and } \lim f\beta \simeq f(b) && (u\bar{f} \text{ continuous, } f = u\bar{f}q) \\ \implies \lim (f\alpha + f\beta) &\simeq f(a) + f(b) && (+ \text{ is continuous}) \\ \implies \lim f(\Sigma\{\alpha, \beta\}) &\simeq f(\Sigma\{a, b\}) && (\Sigma\text{-homomorphism, } X \in \Sigma_{\text{ft}}) \\ \implies \lim u\bar{f}[\Sigma\{\alpha, \beta\}] &\simeq u\bar{f}[\Sigma\{a, b\}] && (f = u\bar{f}q) \\ \implies \lim [\Sigma\{\alpha, \beta\}] &\simeq [\Sigma\{a, b\}]. && (\text{def. } \tau_\Sigma, \text{ Proposition 2.2.22}) \end{aligned}$$

Notice that every net $\alpha': D \rightarrow X/\sim$ may be (non-uniquely) characterised by a net $\alpha: D \rightarrow X$ such that $\alpha' = q\alpha$ since q is surjective. Thus, according to Proposition 2.2.26 and the chain of implications above, the function $+_\Sigma = [\Sigma\{-, -\}]$ is continuous and it follows that $(X/\sim, \tau_\Sigma, +_\Sigma)$ is a Hausdorff commutative monoid. Finally, we must show that the quotient map $q: X \rightarrow G(X/\sim)$ is a Σ -homomorphism. Let $\mathbf{x} \in X^*$ be a summable family and let $\sigma_f: \mathcal{F}(f\mathbf{x}) \rightarrow Y$ and $\sigma_q: \mathcal{F}(q\mathbf{x}) \rightarrow X/\sim$ be the nets of finite partial sums of $f\mathbf{x} \in Y^*$ and $q\mathbf{x} \in (X/\sim)^*$ respectively. Let $m: \mathcal{F}(q\mathbf{x}) \rightarrow \mathcal{F}(f\mathbf{x})$ be the function that maps each subfamily $q\mathbf{x}' \in \mathcal{F}(q\mathbf{x})$ to the subfamily $f\mathbf{x}' = u\bar{f}q\mathbf{x}'$; it is evident that m is isotone. For each $f\mathbf{x}' \in \mathcal{F}(f\mathbf{x})$ it

is immediate that $f\mathbf{x}' \subseteq m(q\mathbf{x}')$ so that $m(\mathcal{F}(q\mathbf{x}))$ is cofinal in $\mathcal{F}(f\mathbf{x})$. Moreover, $u\bar{f}\sigma_q = \sigma_f \circ m$ so that $u\bar{f}\sigma_q$ is a subnet of σ_f . The following chain of implications establishes that q is a Σ -homomorphism:

$$\begin{aligned}
\Sigma\mathbf{x} \simeq x &\implies \Sigma f\mathbf{x} \simeq f(x) && (\Sigma\text{-homomorphism } f) \\
&\implies \lim \sigma_f \simeq f(x) && (\text{def. } \Sigma \text{ in } G(Y)) \\
&\implies \lim (u\bar{f} \circ \sigma_q) \simeq u\bar{f}q(x) && (u\bar{f}\sigma_q \text{ subnet, } f = u\bar{f}q) \\
&\implies \lim \sigma_q \simeq q(x) && (\text{def. } \tau_\Sigma, \text{ Proposition 2.2.22}) \\
&\implies \Sigma q\mathbf{x} \simeq q(x). && (\text{def. } \Sigma \text{ in } G(X/\sim))
\end{aligned}$$

In summary, it has been shown that $X/\sim$ is a Hausdorff commutative monoid and that $X \xrightarrow{q} G(X/\sim)$ is an element in S . Moreover, $f = u\bar{f} \circ q$ and $u\bar{f}$ is a continuous monoid homomorphism, providing a morphism $(q, X/\sim) \rightarrow (f, Y)$ in the comma category. This construction can be reproduced for every $Y \in \mathbf{HausCMon}$ and every Σ -homomorphism $f: X \rightarrow G(Y)$. Consequently, S is a weakly initial set in the comma category $(X \Rightarrow G)$ and the claim that G has a left adjoint follows from the general adjoint functor theorem (see Theorem 1.4.36). $\square$

Corollary 2.2.45. *There is a left adjoint to the functor $G: \mathbf{HausAb} \rightarrow \Sigma_g$.*

Proof. The claim follows from the same argument used in the previous proposition with the exception that, in this case, we must prove that $(X/\sim, \tau_\Sigma, +_\Sigma)$ is a Hausdorff abelian group. It has already been established that $X/\sim \in \mathbf{HausCMon}$, so we only need to prove that inverses exist in $X/\sim$ for each element and that the inversion map is continuous. Since in this case X is a Σ -group, the existence of inverses in $X/\sim$ is immediate, where the inverse of every $[x] \in X/\sim$ is $[-x]$. To prove that the inversion map $[x] \mapsto [-x]$ is continuous, first notice that for every net $\alpha: D \rightarrow X$ the following is satisfied:

$$\begin{aligned}
\lim q\alpha \simeq [a] &\implies \lim f\alpha \simeq f(a) && (u\bar{f} \text{ continuous, } f = u\bar{f}q) \\
&\implies \lim (-f\alpha) \simeq -f(a) && (\text{continuous inversion map in } Y) \\
&\implies \lim f(-\alpha) \simeq f(-a) && (\text{group homomorphism } f) \\
&\implies \lim u\bar{f}[-\alpha] \simeq u\bar{f}[-a] && (f = u\bar{f}q) \\
&\implies \lim [-\alpha] \simeq [-a]. && (\text{def. } \tau_\Sigma, \text{ Proposition 2.2.22})
\end{aligned}$$

Recall that every net $\alpha': D \rightarrow X/\sim$ may be (non-uniquely) characterised by a net $\alpha: D \rightarrow X$ such that $\alpha' = q\alpha$ since q is surjective. Thus, according to Proposi-

tion 2.2.19 and the chain of implications above, the inversion map in $X/\sim$ is continuous and it follows that $(X/\sim, \tau_\Sigma, +_\Sigma)$ is a Hausdorff abelian group, completing the proof. $\square$

2.3 Related work

The main goal of this chapter was to lay the groundwork for Chapter 3 which builds upon Σ_* -enriched categories to formalise iteration in a categorical setting. Such was the objective of Haghverdi's work on unique decomposition categories [28], which are an instance of Σ_s -enriched categories that will take a central role in the next chapter. To this end, Haghverdi introduced a notion of Σ -monoids that has been generalised in Section 2.1 of this thesis. On the other hand, Section 2.2 is, for the most part, an introduction to general topology and Hausdorff commutative monoids; the main contribution in it being the construction of a functor $\mathbf{HausCMon} \rightarrow \Sigma_{\text{ft}}$ that extends the monoid operation to infinitary inputs using the notion of convergence of nets from general topology. A brief discussion of a selection of works closely related to the contents of this chapter is provided below.

Haghverdi [28]. Haghverdi's (strong) Σ -monoids are defined in terms of two axioms: the unary sum axiom and the partition-associativity axiom. The former corresponds precisely to the singleton axiom from weak Σ -monoids (Definition 2.1.1) whereas the latter is equivalent to the combination of the axioms of subsummability, strong bracketing and strong flattening (Definition 2.1.11). Separating partition-associativity into its two directions of implication (*i.e.* strong bracketing and strong flattening) allow us to weaken each of them in a different way. The motivation behind the weakening of flattening is evident from a close inspection of Proposition 2.1.12, since it is strong flattening on *infinite* collections of families what prevents the existence of additive inverses. On the other hand, the motivation of the weakening of bracketing comes from the desire to capture Hausdorff commutative monoids as instances of weak Σ -monoids (see Remark 2.2.38).

Hoshino [33]. Hoshino continued Haghverdi's line of work on Σ_s -enriched unique decomposition categories. In [33], Hoshino proved that the category of total strong Σ -monoids (*i.e.* those where Σ is a total function) is a *reflective* subcategory of strong Σ -monoids. This was achieved via a non-constructive argument using the general ad-

joint functor theorem and its strategy is reproduced in Section 2.1.3 to prove that the embedding $\Sigma_{\text{ft}} \hookrightarrow \Sigma_{\text{w}}$ has a left adjoint. Interestingly, the proof of Proposition 2.1.18 which establishes that Σ_{s} is a reflective subcategory of Σ_{w} can be used (virtually without alteration) to give a constructive proof of Hoshino's result; in this case, the intersection construction from Lemma 2.1.17 is no longer necessary. Moreover, the proof of the existence of a monoidal structure given by the tensor product on every Σ_* category (Section 2.1.5) follows the same strategy Hoshino used in [33] to prove such a result for the particular case of Σ_{s} .

Higgs [32]. The Σ -groups defined in this thesis are equivalent to a subclass of Higgs' notion of Σ -groups; in particular, to those where the set of summable families is restricted to contain countable families only. Thus, the weak Σ -monoids introduced in Section 2.1 are general enough to capture two disjoint notions of infinitary monoids: Haghverdi's strong Σ -monoids (which cannot have inverses) and Higgs' Σ -groups (which must have inverses). To verify that the axioms in Higgs' definition of Σ -groups [32] coincides with ours, notice that the neutral element axiom and the singleton axiom of weak Σ -monoids appear explicitly in Higgs' definition as axioms ($\Sigma 1$) and ($\Sigma 2$) respectively, whereas finite totality is imposed by Higgs by requiring that the Σ -group (X, Σ) contains an abelian group. The bracketing axiom of weak Σ -monoids is Higgs' ($\Sigma 4$) axiom, whereas Higgs' ($\Sigma 3$) axiom corresponds to flattening and is stated as follows (paraphrased using this thesis' notation):

$$\Sigma \mathbf{x} \simeq x \text{ and } \Sigma \mathbf{x}' \simeq x' \implies \Sigma(\mathbf{x} \uplus -\mathbf{x}') \simeq x - x'.$$

Since flattening of weak Σ -monoids only discusses addition of finite collection of families and since Σ -groups are required to be finitely total, Higgs' axiom implies flattening. Moreover, when $\mathbf{x} = \emptyset$, Higgs' axiom implies that every summable family $\mathbf{x}'$ has a counterpart $-\mathbf{x}'$ so that $\Sigma(-\mathbf{x}') \simeq -(\Sigma \mathbf{x}')$. Thus, Higgs combines in a single axiom both the flattening axiom of weak Σ -monoids and the requirement that the inverse mapping is a Σ -homomorphism. With these remarks in mind, it is straightforward to check that Higgs' definition of Σ -groups is equivalent to the definition presented in this thesis.

On another note, the realisation that Hausdorff abelian groups can be captured as instances of Σ -groups is briefly mentioned in the introduction of Higgs' paper [32], although no explicit proof is provided. It seems that such a result does not hold if Σ -groups are required to satisfy strong bracketing (see Remark 2.2.38) and it was Higgs'

definition that revealed the subtle weakening to bracketing that enables us to define a functor $\mathbf{HausAb} \rightarrow \Sigma_g$. For the purposes of this thesis, it is sufficient to study the connection between Σ -monoids and Hausdorff commutative monoids; but Higgs was interested in the more general framework of net groups.⁷ Higgs proved that every Σ -group is a net group and, moreover, the corresponding embedding functor has a right adjoint. This is somewhat dual to the result from Section 2.2.4 where the functor realising every Hausdorff abelian group as a Σ -group — via a Σ function defined in terms of the convergence of certain net — is shown to have a left adjoint.

⁷A net group is an abelian group $(X, +)$ equipped with a relation $\rightarrow$ between nets α in X and elements $x \in X$, along with some axioms that make it reasonable to interpret $\alpha \rightarrow x$ as the statement ‘the net α converges to x ’ without the use of topology.

Chapter 3

Categorical study of quantum iteration

This chapter builds upon the previous one to introduce Σ_* -enriched categories and study categorical traces on them. In particular, we are interested in proving the validity of the execution formula as a categorical trace in certain categories of quantum processes. The execution formula is closely related to the notion of iteration in computer science and, in fact, the main motivation behind this chapter is to provide the categorical foundations to study unbounded quantum iterative loops.

This chapter is structured as follows: Section 3.1 introduces the framework of traced monoidal categories along with standard examples of categorical traces, including the execution formula; Section 3.2 introduces Haghverdi's unique decomposition categories and builds upon them to provide a framework capable of capturing quantum iteration; Section 3.3 proves that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a totally traced category, where ex corresponds to the execution formula; Section 3.4 introduces a particular category of quantum processes over time and shows that the execution formula is a well-defined categorical trace on it; finally, Section 3.5 discusses related work.

3.1 Traced monoidal categories (*Preamble*)

Assume $(\mathbf{C}, \otimes, I)$ is a monoidal category whose morphisms $f: A \rightarrow B$ may be interpreted as processes that receive input of type A and yield output of type B ; Section 1.5 provides examples of such categories in the context of quantum processes. If we wish to formalise the notion of an iterative process within the category $\mathbf{C}$ we need to define

an operation that connects the output of a process back to one of its inputs:

$$\begin{array}{c} \text{U} \\ \text{---} \end{array} \boxed{f} \begin{array}{c} \text{U} \\ \text{---} \end{array} \quad \mapsto \quad \begin{array}{c} \text{U} \\ \text{---} \end{array} \boxed{f} \begin{array}{c} \text{U} \\ \text{---} \end{array} \quad (3.1)$$

More precisely, we need a map of type $\mathbf{C}(A \otimes U, B \otimes U) \rightarrow \mathbf{C}(A, B)$ for all $A, B, U \in \mathbf{C}$ – which we will refer to as the *trace operator* – and, as usual, we ought to impose a set of axioms that ensure this trace operator interacts nicely with composition and the monoidal structure. Such a framework is captured by traced monoidal categories, originally proposed by Joyal, Street and Verity [36] and whose definition is given below. Traced monoidal categories are not limited to iterative loops; for instance, Section 3.1.3 provides an example of a traced monoidal category that captures the notion of calculating the trace of a matrix, which has little in common with the notion of iteration. Consequently, for the purposes of this thesis it will be necessary to find an appropriate definition of the trace operator that captures iteration; such will be the role of the execution formula, as discussed in Section 3.1.1.

Definition 3.1.1. Let $(\mathbf{C}, \otimes, I)$ be a symmetric monoidal category and let there be a family of maps

$$\text{Tr}_{A,B}^U: \mathbf{C}(A \otimes U, B \otimes U) \rightarrow \mathbf{C}(A, B)$$

for all $A, B, U \in \mathbf{C}$. The mapping $\text{Tr}_{A,B}^U$ may be represented pictorially as in (3.1) and, for the sake of brevity, the subscript is often omitted, writing Tr^U instead. $\mathbf{C}$ is a *traced monoidal category* if the following axioms are satisfied.

- *Naturality.* For all morphisms $f: A \otimes U \rightarrow B \otimes U$, $g: A' \rightarrow A$ and $h: B \rightarrow B'$ in $\mathbf{C}$,

$$\begin{array}{c} \text{U} \\ \text{---} \end{array} \boxed{g} \text{---} \boxed{f} \text{---} \boxed{h} \begin{array}{c} \text{U} \\ \text{---} \end{array} \quad = \quad \begin{array}{c} \text{U} \\ \text{---} \end{array} \boxed{g} \boxed{f} \boxed{h} \begin{array}{c} \text{U} \\ \text{---} \end{array}$$

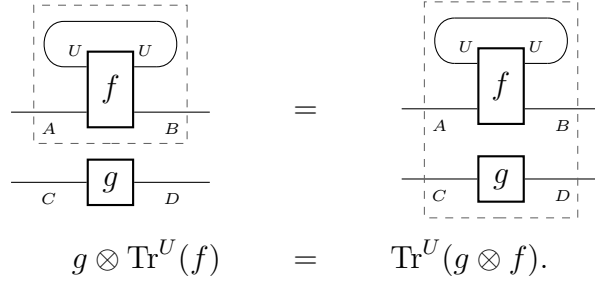
$$h \circ \text{Tr}^U(f) \circ g = \text{Tr}^U((h \otimes \text{id}_U) \circ f \circ (g \otimes \text{id}_U)).$$

- *Dinaturality.* For all morphisms $f: A \otimes U \rightarrow B \otimes U'$ and $g: U' \rightarrow U$ in $\mathbf{C}$,

$$\begin{array}{c} \text{U} \\ \text{---} \end{array} \boxed{f} \boxed{g} \begin{array}{c} \text{U}' \\ \text{---} \end{array} \quad = \quad \begin{array}{c} \text{U}' \\ \text{---} \end{array} \boxed{g} \boxed{f} \begin{array}{c} \text{U}' \\ \text{---} \end{array}$$

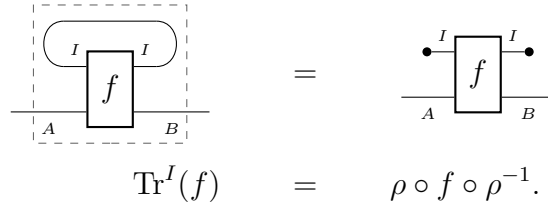
$$\text{Tr}^U((\text{id}_B \otimes g) \circ f) = \text{Tr}^{U'}(f \circ (\text{id}_A \otimes g)).$$

- *Superposing.* For all morphisms $f: A \otimes U \rightarrow B \otimes U$ and $g: C \rightarrow D$ in $\mathbf{C}$,



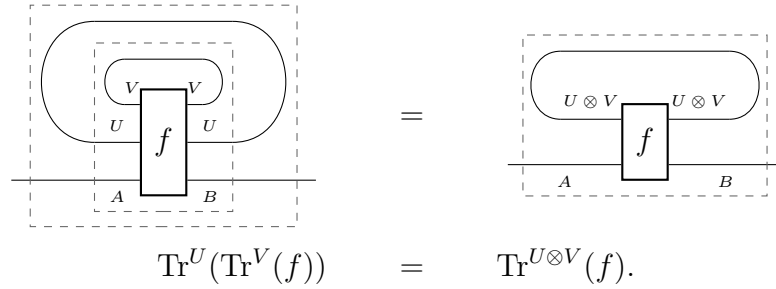
$$g \otimes \text{Tr}^U(f) = \text{Tr}^U(g \otimes f).$$

- *Vanishing I.* For all morphisms $f: A \otimes I \rightarrow B \otimes I$ in $\mathbf{C}$,



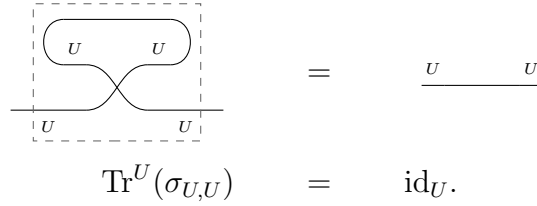
$$\text{Tr}^I(f) = \rho \circ f \circ \rho^{-1}.$$

- *Vanishing II.* For all morphisms $f: A \otimes U \otimes V \rightarrow B \otimes U \otimes V$ in $\mathbf{C}$,



$$\text{Tr}^U(\text{Tr}^V(f)) = \text{Tr}^{U \otimes V}(f).$$

- *Yanking.* For all objects $U \in \mathbf{C}$ where $\sigma_{U,U}$ is the symmetric braiding,



$$\text{Tr}^U(\sigma_{U,U}) = \text{id}_U.$$

In this thesis, traced monoidal categories are denoted by a triple $(\mathbf{C}, \otimes, \text{Tr})$ where Tr is the collection of all trace operators $\text{Tr}_{A,B}^U$ and the monoidal unit is omitted when unambiguous. In certain cases of interest to this thesis, the trace operator $\text{Tr}_{A,B}^U$ cannot be defined for all morphisms in its domain. Haghverdi and Scott [29] proposed a generalisation of traced categories where the trace operator need not be total. To define these, the Kleene equality is used (see Notation 2.0.4) so that, for two partial functions f and g , the expression $f(a) \simeq g(a')$ indicates that $f(a)$ is defined iff $g(a')$ is defined and their results agree.

Definition 3.1.2. Let $(\mathbf{C}, \otimes, I)$ be a symmetric monoidal category and let there be a family of partial maps

$$\mathrm{Tr}_{A,B}^U: \mathbf{C}(A \otimes U, B \otimes U) \rightharpoonup \mathbf{C}(A, B)$$

for all $A, B, U \in \mathbf{C}$. $\mathbf{C}$ is a *partially traced category* if the following axioms are satisfied.

- *Naturality.* For all $f: A \otimes U \rightarrow B \otimes U$, $g: A' \rightarrow A$ and $h: B \rightarrow B'$,

$$\mathrm{Tr}^U(f) \text{ defined} \implies h \circ \mathrm{Tr}^U(f) \circ g \simeq \mathrm{Tr}^U((h \otimes \mathrm{id}_U) \circ f \circ (g \otimes \mathrm{id}_U)).$$

- *Dinaturality.* For all $f: A \otimes U \rightarrow B \otimes U'$ and $g: U' \rightarrow U$,

$$\mathrm{Tr}^U((\mathrm{id}_B \otimes g) \circ f) \simeq \mathrm{Tr}^{U'}(f \circ (\mathrm{id}_A \otimes g)).$$

- *Superposing.* For all $f: A \otimes U \rightarrow B \otimes U$ and $g: C \rightarrow D$,

$$\mathrm{Tr}^U(f) \text{ defined} \implies g \otimes \mathrm{Tr}^U(f) \simeq \mathrm{Tr}^U(g \otimes f).$$

- *Vanishing I.* For all $f: A \otimes I \rightarrow B \otimes I$,

$$\mathrm{Tr}^I(f) \simeq \rho \circ f \circ \rho^{-1}.$$

- *Vanishing II.* For all $f: A \otimes U \otimes V \rightarrow B \otimes U \otimes V$,

$$\mathrm{Tr}^V(f) \text{ defined} \implies \mathrm{Tr}^U(\mathrm{Tr}^V(f)) \simeq \mathrm{Tr}^{U \otimes V}(f).$$

- *Yanking.* For all objects $U \in \mathbf{C}$,

$$\mathrm{Tr}^U(\sigma_{U,U}) \simeq \mathrm{id}_U.$$

Any traced monoidal category $(\mathbf{C}, \otimes, \mathrm{Tr})$ as given in Definition 3.1.1 is, evidently, an instance of a partially traced category where all operators $\mathrm{Tr}_{A,B}^U$ are totally defined; consequently, we refer to these as *totally traced categories*. The PhD thesis of Octavio Malherbe [43] and its accompanying paper [44] are valuable references on partially traced categories. The following proposition is due to Malherbe and collaborators, and it lets us induce a partial trace on $\mathbf{C}$ from the partial trace of another category $\mathbf{D}$.

Proposition 3.1.3 (Malherbe [43], Proposition 4.3.1). *Let $F: \mathbf{C} \rightarrow \mathbf{D}$ be a faithful strong symmetric monoidal functor with $(\mathbf{D}, \otimes, \text{Tr})$ a partially traced category and $(\mathbf{C}, \otimes)$ a symmetric monoidal category. For every morphism $f \in \mathbf{C}(A \oplus U, B \oplus U)$ define:*

$$\widehat{\text{Tr}}^U(f) = \begin{cases} g & \text{if } \exists g \in \mathbf{C}(A, B) \text{ s.t. } \text{Tr}^{F(U)}(\mu_{B,U}^{-1} F(f) \mu_{A,U}) \simeq F(g) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then, $(\mathbf{C}, \otimes, \widehat{\text{Tr}})$ is a partially traced category.

Proof. (Sketch). For simplicity, assume F is strict monoidal (i.e. μ is the identity); for a detailed proof in the general case see [43], Proposition 4.3.1. First, notice that $\widehat{\text{Tr}}$ is well-defined: if $g \in \mathbf{C}(A, B)$ exists such that $\text{Tr}^{F(U)}(F(f)) \simeq F(g)$ then such g is unique due to F being faithful. To check each axiom of partially traced categories we proceed as follows:

- assume $\widehat{\text{Tr}}^U(f)$ is defined and appears on the left (or right) hand side of one of the axioms;
- by definition, $F(\widehat{\text{Tr}}^U(f)) = \text{Tr}^{F(U)}(F(f))$ and we may use that $\mathbf{D}$ is partially traced to derive the existence of the right (or left) hand side of the axiom in $\mathbf{D}$;
- since F is a strict symmetric monoidal functor, the left (and right) hand side of the axiom can be rewritten as F being applied to the corresponding left (and right) hand side of the axiom in $\mathbf{C}$;
- since F is faithful it follows that the axiom in $\mathbf{C}$ is satisfied.

Consequently, $(\mathbf{C}, \otimes, \widehat{\text{Tr}})$ is a partially traced category, as claimed. $\square$

3.1.1 The execution formula

The execution formula is a candidate for a categorical trace in multiple categories relevant in computer science [45, 28, 33]. The primary goal of this section is to explain how the execution formula captures the notion of iterative loops and identify the requirements that a category must meet so that the execution formula may be defined on it.

In the picture below, the box labelled f may be interpreted as a physical device with input ports A and U and output ports B and U whose U -ports have been joined

together; arrows have been added to clarify the direction of information flow within the wires.



Once the connection between the U -ports is made, any input on A may only leave through B , but before doing so, it may traverse the loop an arbitrary number of times. If we intend to describe the relationship between inputs on A and outputs on B , we must aggregate the contribution of all of these possible paths; that is the key idea behind the execution formula. To formalise this in a monoidal category $(\mathbf{C}, \oplus, Z)$ the following conditions must be satisfied:

- morphisms $f: A \oplus U \rightarrow B \oplus U$ must admit a characterisation in terms of components $f_{YX}: X \rightarrow Y$ for each $X \in \{A, U\}$ and $Y \in \{B, U\}$, so that we may study the different paths from A to B in isolation, for instance:

$$A \xrightarrow{f_{BA}} B,$$

$$A \xrightarrow{f_{UA}} U \xrightarrow{f_{UU}} U \xrightarrow{f_{UU}} U \xrightarrow{f_{UU}} U \xrightarrow{f_{BU}} B,$$

etc. — this can be achieved by requiring $A \oplus B$ to be a biproduct for all objects $A, B \in \mathbf{C}$ but, as we will see in Section 3.2, certain categories without biproducts may also satisfy this condition;

- there must be a notion of infinite summation of morphisms, so that the contribution of all paths may be aggregated — this can be achieved by requiring that $\mathbf{C}$ is a category enriched over Σ -monoids, as discussed in Chapter 2.

In such a situation, the *execution formula* may be informally presented as:

$$\text{ex}^U(f) = f_{BA} + \sum_{n=0}^{\infty} f_{BU} f_{UU}^n f_{UA}$$

for any morphism $f: A \oplus U \rightarrow B \oplus U$. The abstract diagram from (3.2) can be interpreted to depict any process f whose component f_{UU} may be applied an arbitrary number of times; if these processes are described by computer programs, we recover the notion of iteration from computer science (more on this in Section 3.2.1). When the execution formula provides a valid categorical trace some works in the literature [1] refer to it as a “particle-style” trace.

Whenever a category $\mathbf{C}$ is traced with respect to the execution formula, we are guaranteed that $\mathbf{C}$ captures a well-behaved notion of iteration, in the sense that it satisfies the axioms of traced monoidal categories and, hence, the execution formula interacts nicely with both composition and the symmetric monoidal structure. But, perhaps surprisingly, proving that the execution formula is a categorical trace is not a simple matter, and the study of unique decomposition categories, initiated by Haghverdi [28], aims to characterise what must be required of $\mathbf{C}$ for this to be the case. This chapter builds upon the framework of unique decomposition categories to formalise the notion of iteration on categories of quantum processes. But, before doing so, we discuss other candidates for trace operators on categories of quantum processes and how these relate to iteration.

3.1.2 The kernel-image trace

As discussed in the previous section, the execution formula may only be defined in a category enriched over Σ -monoids since there is an infinite number of execution paths that need to be aggregated. This section discusses a couple of closely related trace operators, defined on categories enriched over abelian groups.

Definition 3.1.4. An *additive category* is an $\mathbf{Ab}$ -enriched category that has all finite biproducts. Every additive category $\mathbf{C}$ has a canonical monoidal structure $(\mathbf{C}, \oplus, 0)$ where $\oplus$ is the canonical extension of binary biproducts to a monoidal product and 0 is a zero object.

Notation 3.1.5. For a morphism $f: A \oplus B \rightarrow C \oplus D$ it is useful to refer to its components using the subscript shorthand f_{CA} , in this case referring to the morphism

$$f_{CA} = A \xrightarrow{\iota} A \oplus B \xrightarrow{f} C \oplus D \xrightarrow{\pi} C$$

where ι and π are the injection and projection morphisms of the biproduct. The order of the subscripts facilitates the readability of composition: for another morphism $h: X \rightarrow A \oplus B$, we may compose $f_{CA} \circ h_{AX}$ and it is immediate to read, from right to left, the objects that the morphism factors through.

Haghverdi and Scott [29] studied the following partial trace, which may be defined on every additive category.

Proposition 3.1.6 (Haghverdi and Scott [29]). *Every additive category $(\mathbf{C}, \oplus, 0)$ is partially traced, where its trace operator $\text{Tr}_{A,B}^U$ is defined as follows:*

$$\text{Tr}_{A,B}^U(f) = \begin{cases} f_{BA} + f_{BU}(\text{id}_U - f_{UU})^{-1}f_{UA} & \text{if } \text{id}_U - f_{UU} \text{ is invertible} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The proof of this result has been omitted for brevity and can be found in [29]. This Haghverdi-Scott trace is somewhat similar to the execution formula, especially in the light of the following well-known identity on the real numbers $|r| < 1$:

$$\sum_{n=0}^{\infty} r^n = (1 - r)^{-1}.$$

However, there is a caveat: the Haghverdi-Scott trace of morphisms such as the identity $f = \text{id}_{A \oplus U}$ is undefined since

$$\text{id}_U - f_{UU} = \text{id}_U - \pi_U \circ \text{id}_{A \oplus U} \circ \iota_U = \text{id}_U - \text{id}_U = 0$$

is clearly not invertible. This is somewhat odd since $\text{Tr}^U(\text{id}_{A \oplus U}) \simeq \text{id}_A \oplus \text{Tr}^U(\text{id}_U)$ according to the superposing axiom and, even though $\text{Tr}^U(\text{id}_U)$ is undefined as well, its type is $0 \rightarrow 0$ suggests that it may naturally be defined as the zero morphism. In such a scenario, the superposing axiom would yield $\text{Tr}^U(\text{id}_{A \oplus U}) \simeq \text{id}_A$ which matches the intuition behind iteration since tracing $\text{id}_{A \oplus U}$ would create an isolated loop on U that no input can reach. Indeed, Malherbe, Scott and Selinger [44] proposed a generalisation of the Haghverdi-Scott trace where identities and other morphisms that would ‘leave an isolated loop’ can be traced.

Definition 3.1.7 (Malherbe *et al.* [44]). Let $\mathbf{C}$ be an additive category. A morphism $f: A \oplus U \rightarrow B \oplus U$ in $\mathbf{C}$ is said to have a trace witnessed by morphisms $i: A \rightarrow U$ and $k: U \rightarrow B$ if the diagram

$$\begin{array}{ccc} A & \xrightarrow{\quad i \quad} & U \\ & \searrow f_{UA} & \downarrow \text{id} - f_{UU} \\ & & U \\ & & \nearrow f_{BU} \\ & & B \\ & \swarrow k & \\ & & \end{array}$$

commutes. Whenever it does, we write $(k, i) \Vdash \text{Tr}^U(f)$ and define $\text{Tr}^U(f): A \rightarrow B$ as follows:

$$\text{Tr}^U(f) = f_{BA} + k \circ f_{UA} = f_{BA} + f_{BU} \circ i$$

and otherwise we leave $\text{Tr}^U(f)$ undefined. The collection of all such partial functions $\text{Tr}^U: \mathbf{C}(A \oplus U, B \oplus U) \rightarrow \mathbf{C}(A, B)$ for all objects $A, B, U \in \mathbf{C}$ is known as the *kernel-image trace*.

Proposition 3.1.8 (Malherbe *et al.* [44], Proposition 3.17). *Any additive category is partially traced using the kernel-image trace.*

We write Tr_{ki} when we wish to distinguish the kernel-image trace from other traces available on the same category. The following proposition justifies the name given to the kernel-image trace.

Proposition 3.1.9 (Malherbe *et al.* [44], Remark 3.16). *A morphism $f: A \oplus U \rightarrow B \oplus U$ in $\mathbf{Vect}$ satisfies $(k, i) \Vdash \text{Tr}^U(f)$ for some morphisms $k \in \mathbf{Vect}(U, B)$ and $i \in \mathbf{Vect}(A, U)$ iff the following inclusions hold:*

$$\begin{aligned} \text{im}(f_{UA}) &\subseteq \text{im}(\text{id} - f_{UU}) \\ \ker(\text{id} - f_{UU}) &\subseteq \ker(f_{BU}). \end{aligned} \tag{3.3}$$

Proof. Assume that $(k, i) \Vdash \text{Tr}^U(f)$ is satisfied; this implies the existence of a morphism $i: A \rightarrow U$ such that $f_{UA} = (\text{id} - f_{UU}) \circ i$ and, hence, $u \in \text{im}(f_{UA})$ implies $u \in \text{im}(\text{id} - f_{UU})$. Similarly, since a morphism $k: U \rightarrow B$ satisfying $f_{BU} = k \circ (\text{id} - f_{UU})$ exists, it follows that for any $u \in U$ such that $(\text{id} - f_{UU})(u) = 0$ we must have $f_{BU}(u) = 0$. Therefore, $(k, i) \Vdash \text{Tr}^U(f)$ implies both inclusions (3.3).

Conversely, assume that both of inclusions (3.3) are satisfied and let $h = \text{id} - f_{UU}$. Decompose h as follows:¹

$$h = U \xrightarrow{q} U/\ker(h) \xrightarrow{h'} \text{im}(h) \xrightarrow{m} U$$

where $U/\ker(h)$ is the quotient space induced by the equivalence relation $\sim$ where:

$$u \sim v \iff (u - v) \in \ker(h)$$

and q is its quotient map. The linear map h' sends equivalence classes $[u] \in U/\ker(h)$ to $h(u)$; this is well-defined since $[u] = [v]$ implies $h(u - v) = 0$ and hence $h(u) = h(v)$. It can be shown that h' is bijective, q is surjective and m is injective. Consequently, $h' \circ q$ is surjective, so there is at least one linear map $r: \text{im}(h) \rightarrow U$ acting as its right inverse, $(h' \circ q) \circ r = \text{id}$. Similarly, $m \circ h'$ is injective, so there is at least one linear map $l: U \rightarrow U/\ker(h)$ acting as its left inverse, $l \circ (m \circ h') = \text{id}$. Now, since we assumed that $\text{im}(f_{UA}) \subseteq \text{im}(h)$ we may decompose f_{UA} as follows:

$$f_{UA} = A \xrightarrow{f'_{UA}} \text{im}(h) \xrightarrow{m} U.$$

¹This decomposition is available in every abelian category, which $\mathbf{Vect}$ is an example of.

Let $i = r \circ f'_{UA}$; it is immediate that:

$$(\text{id} - f_{UU}) \circ i = (m \circ h' \circ q) \circ (r \circ f'_{UA}) = m \circ f'_{UA} = f_{UA}.$$

On the other hand, since we assumed that $\ker(h) \subseteq \ker(f_{BU})$ we may decompose f_{BU} as follows:

$$f_{BU} = U \xrightarrow{q} U/\ker(h) \xrightarrow{f'_{BU}} B.$$

Let $k = f'_{BU} \circ l$; it is immediate that:

$$k \circ (\text{id} - f_{UU}) = (f'_{BU} \circ l) \circ (m \circ h' \circ q) = f'_{BU} \circ q = f_{BU}.$$

Therefore, the kernel-image inclusions (3.3) imply $(k, i) \Vdash \text{Tr}^U(f)$. $\square$

Corollary 3.1.10. *A morphism $f: A \oplus U \rightarrow B \oplus U$ in $\mathbf{FdHilb}$ satisfies $(k, i) \Vdash \text{Tr}^U(f)$ for some morphisms $k \in \mathbf{FdHilb}(U, B)$ and $i \in \mathbf{FdHilb}(A, U)$ iff the kernel-image inclusions (3.3) hold.*

Proof. The claim can be proven using the argument from the previous proposition. However, we must check that the linear maps i and k defined in the second part of the proof are bounded. But this is immediate since every linear map between finite-dimensional spaces is bounded. $\square$

The case of $\mathbf{Hilb}$ is more complicated. On one hand, $(\mathbf{Hilb}, \oplus, \{0\})$ is an additive category, so it follows from Proposition 3.1.8 that $(\mathbf{Hilb}, \oplus, \text{Tr}_{\text{ki}})$ is a partially traced category. On the other hand, it appears that the kernel-image inclusions (3.3) are not sufficient for a morphism f in $\mathbf{Hilb}$ to be traceable. This is because when the domain and codomain are infinite-dimensional the image of f need not be a closed subspace, which causes problems when attempting to prove the existence and boundness of i and k as constructed in the proof of Proposition 3.1.9. Taking this into account, the following proposition establishes sufficient conditions for f to be traceable.

Proposition 3.1.11. *For any $f \in \mathbf{Hilb}(A \oplus U, B \oplus U)$, if*

- *it satisfies the kernel-image inclusions (3.3) and*
- *$\text{im}(\text{id} - f_{UU})$ is a closed subspace,*

then $(k, i) \Vdash \text{Tr}^U(f)$ for some morphisms $k \in \mathbf{Hilb}(U, B)$ and $i \in \mathbf{Hilb}(A, U)$.

Proof. (Sketch). As in the proof of Proposition 3.1.9, let $h = \text{id} - f_{UU}$ and decompose it as follows:

$$h = U \xrightarrow{q} U/\ker(h) \xrightarrow{h'} \text{im}(h) \xrightarrow{m} U.$$

This decomposition exists in **Hilb** precisely because $\text{im}(h)$ is closed by assumption and, hence, it is a Hilbert space, whereas it can be shown that $U/\ker(h)$ is isomorphic to the orthogonal complement of $\ker(h)$ and, hence, a Hilbert space as well; since h is bounded, it follows that h' is bounded and it is simple to show that m and q are bounded as well. Then, it is immediate that $h' \circ q$ is surjective, so there is at least one linear map $r: \text{im}(h) \rightarrow U$ acting as its right inverse. We must show that r is bounded; according to Proposition 4.6.1 from [7] this is implied by $h' \circ q$ being a surjective bounded linear map. Similarly, $m \circ h'$ is injective, so there is at least one linear map $l: U \rightarrow U/\ker(h)$ acting as its left inverse. We must show that l is bounded; according to Proposition 4.5.2 from [7] this is implied by $m \circ h'$ being an injective bounded linear map and $\text{im}(m \circ h') = \text{im}(h)$ being a closed subspace — the latter being satisfied by assumption. It is straightforward to check that both f_{UA} and f_{BU} may be decomposed into

$$\begin{aligned} f_{UA} &= A \xrightarrow{f'_{UA}} \text{im}(h) \xrightarrow{m} U \\ f_{BU} &= U \xrightarrow{q} U/\ker(h) \xrightarrow{f'_{BU}} B. \end{aligned}$$

where f'_{UA} and f'_{BU} are bounded. It follows that $i = r \circ f'_{UA}$ and $k = f'_{BU} \circ l$ are bounded linear maps and, hence, morphisms in **Hilb**. What remains of the proof follows by the same argument from Proposition 3.1.9. $\square$

The following two propositions are well-known facts about the orthogonal complement² of Hilbert spaces and they will be used in Section 3.3 to prove that the kernel-image inclusions (3.3) are satisfied by every morphism in **FdContraction**.

Proposition 3.1.12. *Every morphism $f \in \mathbf{Hilb}(A, B)$ satisfies:*

$$\ker(f^\dagger) = \text{im}(f)^\perp$$

where $f^\dagger$ is the adjoint of f .³

²Given a Hilbert space H and a set of vectors $S \subseteq H$, the orthogonal complement of S is defined as follows: $S^\perp = \{v \in H \mid \forall u \in S, \langle u|v \rangle = 0\}$. It can be shown that $S^\perp$ is a closed subspace of H .

³Recall that the adjoint of a bounded linear map $f \in \mathbf{Hilb}(A, B)$ is the bounded linear map $f^\dagger \in \mathbf{Hilb}(B, A)$ satisfying $\langle f(a)|b \rangle = \langle a|f^\dagger(b) \rangle$ for every $a \in A$ and $b \in B$.

Proof. Recall that in a Hilbert space A , every vector $a \in A$ satisfies:

$$a = 0 \iff \forall a' \in A, \langle a' | a \rangle = 0. \quad (3.4)$$

Then, the following sequence of implications prove the claim:

$$\begin{aligned} v \in \ker(f^\dagger) &\iff \forall a \in A, \langle a | f^\dagger(v) \rangle = 0 \quad (\text{implication (3.4)}) \\ &\iff \forall a \in A, \langle f(a) | v \rangle = 0 \quad (\text{def. of adjoint}) \\ &\iff v \in \text{im}(f)^\perp \quad (\text{def. of orthogonal complement}). \quad \square \end{aligned}$$

Proposition 3.1.13. *Let H be a Hilbert space, K a closed subspace of H and S a subset of H . Then,*

$$S \subseteq K \iff K^\perp \subseteq S^\perp.$$

Proof. Assume $S \subseteq K$; for any $v \in K^\perp$ we have that for all $u \in K$, $\langle u | v \rangle = 0$. Thus, in particular, for all $u \in S$, $\langle u | v \rangle = 0$ and $v \in S^\perp$. To prove the other direction, assume $K^\perp \subseteq S^\perp$; then the previous argument establishes that $S^{\perp\perp} \subseteq K^{\perp\perp}$. It is immediate that $S \subseteq S^{\perp\perp}$ and, for any closed subspace, $K = K^{\perp\perp}$. Thus, $S \subseteq S^{\perp\perp} \subseteq K^{\perp\perp} = K$ and the proof is complete. $\square$

3.1.3 The canonical trace on compact closed categories

This section describes the canonical trace on the monoidal category $(\mathbf{FdHilb}, \otimes, \mathbb{C})$ and, more generally, on any compact closed category. It will be argued that the trace in $(\mathbf{FdHilb}, \otimes, \mathbb{C})$ is not well suited to capture the notion of quantum iteration. Nevertheless, this compact category is of great importance in the study of categorical quantum mechanics [31, 18] and it is worth discussing it briefly.

A monoidal category $(\mathbf{C}, \otimes, I)$ is said to have (right) duals if for every object $A \in \mathbf{C}$ there is an object $A^* \in \mathbf{C}$ and morphisms $\eta: I \rightarrow A^* \otimes A$ and $\varepsilon: A \otimes A^* \rightarrow I$ such that the following diagrams commute:

$$\begin{array}{ccccc} A & \xrightarrow{\rho^{-1}} & A \otimes I & \xrightarrow{\text{id} \otimes \eta} & A \otimes (A^* \otimes A) \\ \text{id} \downarrow & & & & \downarrow \alpha^{-1} \\ A & \xleftarrow{\lambda} & I \otimes A & \xleftarrow{\varepsilon \otimes \text{id}} & (A \otimes A^*) \otimes A \end{array} \quad (3.5)$$

$$\begin{array}{ccccc} A^* & \xrightarrow{\lambda^{-1}} & I \otimes A^* & \xrightarrow{\eta \otimes \text{id}} & (A^* \otimes A) \otimes A^* \\ \text{id} \downarrow & & & & \downarrow \alpha \\ A^* & \xleftarrow{\rho} & A^* \otimes I & \xleftarrow{\text{id} \otimes \varepsilon} & A^* \otimes (A \otimes A^*) \end{array}$$

A symmetric monoidal category with duals which interact nicely with the monoidal structure is known as a compact closed category (see Chapter 3, Definition 3.34 from [31] for a proper definition). Every compact closed category is totally traced, with the trace of a morphism $f: A \otimes U \rightarrow B \otimes C$ defined as follows:

$$\mathrm{Tr}^U(f) = (\mathrm{id}_B \otimes \varepsilon) \circ (f \otimes \mathrm{id}_{U^*}) \circ (\mathrm{id}_A \otimes \sigma\eta). \quad (3.6)$$

In particular, $\mathbf{FdHilb}$ is a compact closed category: Section 1.5 already established that $(\mathbf{FdHilb}, \otimes, \mathbb{C})$ is symmetric monoidal, and it can be shown that the dual of a finite-dimensional Hilbert space H is the space defined on the hom-set $H^* = \mathbf{FdHilb}(H, \mathbb{C})$ using pointwise addition and scalar multiplication; since H^* is finite-dimensional it can be made into a Hilbert space by defining an appropriate inner product. Given an orthonormal basis $\{e_i \in H\}_{1 \leq i \leq \dim(H)}$ of H , an orthonormal basis of H^* is given by the collection $\{\phi_i: H \rightarrow \mathbb{C}\}_{1 \leq i \leq \dim(H)}$ where ϕ_i is defined for each $v \in H$ as follows:

$$\phi_i(v) = \langle e_i | v \rangle.$$

The morphisms $\eta: \mathbb{C} \rightarrow H^* \otimes H$ and $\varepsilon: H \otimes H^* \rightarrow \mathbb{C}$ are defined by linear extension of:

$$\eta(1) = \sum_{1 \leq i \leq \dim(H)} \phi_i \otimes e_i \quad \text{and} \quad \varepsilon(e_i \otimes \phi_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

and it is straightforward to check that the diagrams (3.5) commute. Then, for any morphism $f \in \mathbf{FdHilb}(A \otimes U, B \otimes U)$ the trace defined using the compact closed structure (3.6) yields:

$$\mathrm{Tr}^U(f) = \sum_{1 \leq i \leq \dim(U)} (\mathrm{id}_B \otimes \phi_i) \circ f \circ (\mathrm{id}_A \otimes e_i) \quad (3.7)$$

which corresponds to the linear algebraic trace of a block matrix $f = (f_{i,j})_{i,j \leq \dim(U)}$ determined by the orthonormal basis $\{e_i\}_{i \leq \dim(U)}$ of U , where each block is of type $f_{i,j}: A \rightarrow B$.

Such a categorical trace on $\mathbf{FdHilb}$ has little in common with the notion of iterative loops: Section 3.1.1 identifies the requirement that morphisms $f: A \otimes U \rightarrow B \otimes U$ must be characterisable in terms of components f_{BA} , f_{BU} , f_{UA} and f_{UU} , which is certainly not the case for the tensor product $\otimes$. Importantly, the fact that $(\mathbf{FdHilb}, \otimes, \mathbb{C})$ cannot be given a categorical trace that captures iteration is only due to its monoidal structure and has nothing to do with its compact closedness. In fact, there are compact closed categories whose canonical trace coincides with the execution formula; for

instance, this is the case on any category obtained as the result of applying the **Int** construction (see [36]) to a category traced with respect to the execution formula.

3.2 Unique decomposition categories

Unique decomposition categories were introduced by Haghverdi [28] as a general framework where a morphism $f: A \oplus U \rightarrow B \oplus U$ may be decomposed as a matrix

$$f = \begin{pmatrix} f_{BA}: A \rightarrow B & f_{BU}: U \rightarrow B \\ f_{UA}: A \rightarrow U & f_{UU}: U \rightarrow U \end{pmatrix}$$

and where a notion of infinite summation of morphisms exists, so that the execution formula may be defined:

$$\text{ex}^U(f) = f_{BA} + \sum_{n=0}^{\infty} f_{BU} f_{UU}^n f_{UA}. \quad (3.8)$$

The execution formula is meant to behave as a categorical trace, and the intuition is that it aggregates all possible paths that go from A to B . However, as discussed by Hoshino in Appendix B from [33], Haghverdi's original definition is too weak: it admits categories whose execution formula violates vanishing I. Hoshino's proposal was to define a stronger version of unique decomposition categories; their definition is provided below.

Definition 3.2.1. Let Σ_* be one of the categories of Σ -monoids from Definition 2.1.15 endowed with a monoidal structure $(\Sigma_*, \otimes, I)$ given by the tensor product as described in Section 2.1.5. For a Σ_* -category $\mathbf{C}$ let there be:

- a zero object $Z \in \mathbf{C}$ and
- a functor $\oplus: \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ such that its action on hom-objects on arbitrary $A, B, C, D \in \mathbf{C}$ is described by a morphism in Σ_*

$$- \oplus -: \mathbf{C}(A, C) \times \mathbf{C}(B, D) \rightarrow \mathbf{C}(A \oplus B, C \oplus D)$$

whose domain is a categorical product in Σ_* .

If $(\mathbf{C}, \oplus, Z)$ is a symmetric monoidal category we say that it is a Σ_* -enriched *unique decomposition category* (Σ_* -UDC).

Slight changes have been made to Hoshino's original definition [33]. On one hand, Hoshino only discusses Σ_s -enriched UDCs; the definition has been generalised so that all flavours of Σ -monoids introduced in Chapter 2 may be used. On the other hand, Hoshino asks that id_Z is the neutral element of the Σ -monoid $\mathbf{C}(Z, Z)$, instead of requiring that the monoidal unit is a zero object. These conditions are equivalent: if $\text{id}_Z = \Sigma\emptyset$ any morphism $f: Z \rightarrow A$ is equal to the neutral element:

$$f = f \circ \text{id}_Z = f \circ (\Sigma\emptyset) = \Sigma\emptyset$$

due to composition being Σ -bilinear; thus, Z is initial and a similar argument shows that Z is also terminal. Notice that the zero morphism of each hom-object $\mathbf{C}(A, B)$ — *i.e.* the unique morphism that factors through Z — is precisely the neutral element of the Σ -monoid $\mathbf{C}(A, B)$. Moreover, if Z is a zero object then $\mathbf{C}(Z, Z)$ is a singleton hom-set, so id_Z must be the neutral element of the Σ -monoid $\mathbf{C}(Z, Z)$. Furthermore, Hoshino requires that

$$\Sigma\{\text{id}_A \oplus 0_{B,B}, 0_{A,A} \oplus \text{id}_B\} \simeq \text{id}_{A \oplus B}$$

but this can be inferred from our definition:

$$\begin{aligned} \Sigma\{\text{id}_A \oplus 0_{B,B}, 0_{A,A} \oplus \text{id}_B\} &\simeq \Sigma\{\text{id}_A, 0_{A,A}\} \oplus \Sigma\{0_{B,B}, \text{id}_B\} && \text{(product in } \Sigma_*) \\ &\simeq \text{id}_A \oplus \text{id}_B && \text{(\Sigma-monoid axioms)} \\ &= \text{id}_{A \oplus B}. && \text{(\oplus functor)} \end{aligned}$$

Alternatively, it can be shown that this equality imposes that the functor $\oplus$ acts on hom-objects as a Σ -homomorphism whose domain is a categorical product and, hence, our definition is equivalent to Hoshino's in the case of Σ_s -UDCs.

Every monoidal category whose unit is a zero object has certain morphisms that act in a similar way to projections and injections, even though they may lack their universal property. These morphisms are essential in the discussion of UDCs and we define them below.

Definition 3.2.2. For each finite collection of objects $\{A_j \in \mathbf{C}\}_J$ and each $i \in J$, define the *canonical quasi-projection* $\pi_i: \oplus_J A_j \rightarrow A_i$ to be the unique morphism of its type generated by $\{\text{id}, \alpha, \alpha^{-1}, \lambda, \rho\}$ together with the zero morphism $0: A_j \rightarrow Z$ for each $j \in J$. Similarly, for each $i \in J$ define the *canonical quasi-injection* $\pi_i: A_i \rightarrow \oplus_J A_j$ to be the unique morphism generated by $\{\text{id}, \alpha, \alpha^{-1}, \lambda^{-1}, \rho^{-1}\}$ along with the zero morphism $0: Z \rightarrow A_j$ for each $j \in J$.

The equations Haghverdi required in the definition of a UDC [28] are satisfied in the case of canonical quasi-injections and canonical quasi-projections, as shown in the following remark.

Remark 3.2.3. It follows from the definition of the canonical quasi-projection that $\pi_A: A \oplus B \rightarrow A$ is $\pi_A = \rho(\text{id}_A \oplus 0)$ and the canonical quasi-injection $\iota_A: A \rightarrow A \oplus B$ is $\iota_A = (\text{id}_A \oplus 0)\rho^{-1}$; similarly for π_B and ι_B using the left unitor λ instead. It is immediate that $\pi_A \iota_A = \text{id}_A$ whereas $\pi_A \iota_B = 0$. The fact that $\Sigma\{\text{id}_A \oplus 0, 0 \oplus \text{id}_B\} \simeq \text{id}_{A \oplus B}$ implies that:

$$\Sigma\{\iota_A \pi_A, \iota_B \pi_B\} \simeq \text{id}_{A \oplus B}.$$

This suggests that the monoidal structure on $\mathbf{C}$ is very similar to the canonical one on a category with biproducts; indeed, every additive category (see Definition 3.1.4) is a Σ_g -UDC, as established below.

Example 3.2.4. Every additive category $(\mathbf{C}, \oplus, 0)$ is a Σ_g -UDC. To check this, recall that an additive category is enriched over $\mathbf{Ab}$ and every abelian group is a Σ -group whose Σ function on infinite families is undefined. Moreover, the monoidal unit of an additive category is a zero object according to the canonical definition of its monoidal structure, and the requirement that $\Sigma\{\text{id}_A \oplus 0, 0 \oplus \text{id}_B\} = \text{id}_{A \oplus B}$ is immediate from the universal property of biproducts.

Example 3.2.5. Both $\mathbf{Hilb}$ and $\mathbf{FdHilb}$ are additive categories and, hence, they are Σ_g -UDCs. However, as discussed above, such a trivially defined Σ_g -enrichment assumes no infinite family is summable. In contrast, Section 3.3 will provide a Hausdorff topology on each hom-set of $\mathbf{Hilb}$, thus realising each of them as a Σ -group via the functor $\mathbf{HausAb} \rightarrow \Sigma_g$ from Definition 2.2.40 which defines summability of infinite families in terms of convergence. Since the definition of UDC (Definition 3.2.1) does not impose any extra condition on the summability of infinite families, it is immediate that $\mathbf{Hilb}$ and $\mathbf{FdHilb}$ are still Σ_g -UDCs. Similarly, $\mathbf{Contraction}$ and $\mathbf{FdContraction}$ are Σ_w -UDCs; the Σ function on each hom-set of $\mathbf{Contraction}$ is induced from the inclusion $\mathbf{Contraction}(A, B) \hookrightarrow \mathbf{Hilb}(A, B)$ using the construction from Lemma 2.1.8. On the other hand, $\mathbf{Isometry}$ and $\mathbf{Unitary}$, along with their subcategories on finite dimensions are not UDCs since they lack a terminal object (and hence, a zero object).

Notice that a Σ_* -UDC need not have biproducts; for instance, this is the case for

Contraction since the universal morphism for the diagram:

$$\begin{array}{ccccc}
 & & A & & \\
 & \swarrow \text{id} & \downarrow \text{---} & \searrow \text{id} & \\
 A & \xleftarrow{\pi_A} & A \oplus A & \xrightarrow{\pi_B} & A
 \end{array}$$

would need to be the diagonal map $a \mapsto (a, a)$ for all $a \in A$, which is evidently not a contraction. However, whenever the corresponding universal morphism does exist, it is unique, as established by the following proposition. This is perhaps the most important characteristic of unique decomposition categories, justifying their name.

Proposition 3.2.6 (Haghverdi [28]). *Let $(\mathbf{C}, \oplus, Z)$ be a Σ_* -UDC. Let $f: \oplus_I A_i \rightarrow \oplus_J B_j$ be a morphism in $\mathbf{C}$, with finite I and J . There is a unique summable family $\{f_{ji} \in \mathbf{C}(A_i, B_j)\}_{J \times I}$ satisfying:*

$$\Sigma\{\iota_j \circ f_{ji} \circ \pi_i\}_{J \times I} \simeq f.$$

Proof. It straightforward to check that the following equations are satisfied for any finite collection of objects $\{A_i \in \mathbf{C}\}_I$:

$$\pi_i \circ \iota_{i'} = \begin{cases} \text{id} & \text{if } i = i' \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \Sigma\{\iota_i \circ \pi_i\}_I \simeq \text{id}. \quad (3.9)$$

The first equation follows trivially from the definition of the canonical quasi-injections and quasi-projections, the second one follows from the axiom $\Sigma\{\text{id}_A \oplus 0, 0 \oplus \text{id}_B\} \simeq \text{id}_{A \oplus B}$ of Σ_* -UDCs and the flattening axiom of weak Σ -monoids.

Let $f: \oplus_I A_i \rightarrow \oplus_J B_j$ be a morphism in $\mathbf{C}$, with finite I and J . Let $f_{ji} = \pi_j f \iota_i$; then:

$$\begin{aligned}
 f &= \text{id} \circ f \circ \text{id} = \Sigma\{\iota_j \pi_j\}_J \circ f \circ \Sigma\{\iota_i \pi_i\}_I \quad (\text{equation (3.9)}) \\
 &\simeq \Sigma\{\iota_j \pi_j f \circ \Sigma\{\iota_i \pi_i\}_I\}_J \quad (\circ \text{ is } \Sigma\text{-bilinear}) \\
 &\simeq \Sigma\{\Sigma\{\iota_j \pi_j f \iota_i \pi_i\}_I\}_J \quad (\circ \text{ is } \Sigma\text{-bilinear}) \\
 &\simeq \Sigma\{\iota_j f_{ji} \pi_i\}_{J \times I} \quad (\text{flattening axiom and def. of } f_{ji}).
 \end{aligned}$$

Thus, there is at least one family $\{f_{ji}\}_{J \times I}$ satisfying the claim. To show uniqueness, assume there is another family $\{g_{j'i'}\}_{J \times I}$ satisfying $\Sigma\{\iota_{j'} \circ g_{j'i'} \circ \pi_{i'}\}_{J \times I} \simeq f$; then:

$$f_{ji} = \pi_j f \iota_i = \pi_j (\Sigma\{\iota_{j'} g_{j'i'} \pi_{i'}\}_{J \times I}) \iota_i \simeq \Sigma\{\pi_j \iota_{j'} g_{j'i'} \pi_{i'} \iota_i\}_{J \times I} = \Sigma\{g_{ji}\} = g_{ji}$$

due to composition being Σ -bilinear. Therefore, both families $\{f_{ji}\}_{J \times I}$ and $\{g_{j'i'}\}_{J \times I}$ contain the same elements and are thus the same family. $\square$

The family of components has a suggestive representation as a matrix; for instance, for $f: A \oplus B \rightarrow C \oplus D$ we may write:

$$f = \begin{pmatrix} f_{CA} & f_{CB} \\ f_{DA} & f_{DB} \end{pmatrix}$$

The composition of two morphisms can be given in terms of their ‘matrix multiplication’ where multiplication of entries is replaced by their composition, as established below.

Proposition 3.2.7. *Let $(\mathbf{C}, \oplus, Z)$ be a Σ_* -UDC. Let $f: \oplus_I A_i \rightarrow \oplus_J B_j$ and $g: \oplus_J B_j \rightarrow \oplus_K C_k$ be two morphisms in $\mathbf{C}$, with finite I, J and K . Then, for each $(k, i) \in K \times I$ the family $\{g_{kj} \circ f_{ji}\}_J$ is summable and the unique decomposition of $g \circ f$ is given by the collection of morphisms*

$$(g \circ f)_{ki} = \Sigma\{g_{kj} \circ f_{ji}\}_J$$

for all $(k, i) \in K \times I$.

Proof. We show that the family $\{g_{kj} \circ f_{ji}\}_J$ is summable for all $(k, i) \in K \times I$:

$$\begin{aligned} \pi_k(g \circ f)\iota_i &\simeq \pi_k g \circ (\Sigma\{\iota_j \pi_j\}_J) \circ f \iota_i && \text{(equation (3.9))} \\ &\simeq \Sigma\{\pi_k g \iota_j \pi_j f \iota_i\}_J && (\circ \text{ is } \Sigma\text{-bilinear}) \\ &= \Sigma\{g_{kj} \circ f_{ji}\}_J. && \text{(def. of } g_{kj} \text{ and } f_{ji}) \end{aligned}$$

Recall that $(g \circ f)_{ki} = \pi_k(g \circ f)\iota_i$ by uniqueness of the decomposition, thus completing the proof of the claim. $\square$

Haghverdi’s original definition of UDCs simply required $\mathbf{C}$ to be Σ_s -enriched, symmetric monoidal, and satisfy (3.9). Unlike Hoshino, Haghverdi did not impose a particular choice of morphisms to act as quasi-injections and quasi-projections, which permitted the choice of ones that altered the input in a subtle way, causing vanishing I to fail for the execution formula (see Appendix B from [33] for further details). In the Σ_* -UDCs of Hoshino, the symmetric monoidal structure satisfies some useful properties discussed in the following remarks.

Remark 3.2.8. Let $f: A \rightarrow B$ and $g: C \rightarrow D$ be morphisms in a Σ_* -UDC $(\mathbf{C}, \oplus, Z)$. It follows that

$$\begin{aligned} \pi_B(f \oplus g)\iota_A &= \rho(\text{id}_B \oplus 0)(f \oplus g)(\text{id}_A \oplus 0)\rho^{-1} && \text{(def. of } \pi_B \text{ and } \iota_A) \\ &= \rho(f \oplus 0)\rho^{-1} && \text{(functoriality of } \oplus, \text{ zero morphism)} \\ &= f \circ \rho(\text{id}_A \oplus \text{id}_Z)\rho^{-1} = f && \text{(nat. of } \rho, \text{id}_Z = 0) \end{aligned}$$

and similarly $\pi_D(f \oplus g)\iota_C = g$, whereas $\pi_B(f \oplus g)\iota_C = 0$ and $\pi_D(f \oplus g)\iota_A = 0$. Therefore, the unique decomposition of $f \oplus g$ is:

$$f \oplus g = \begin{pmatrix} f & 0 \\ 0 & g \end{pmatrix}.$$

Remark 3.2.9. Let $\sigma: A \oplus B \rightarrow B \oplus A$ be the symmetric braiding of a Σ_* -UDC $(\mathbf{C}, \oplus, Z)$. It follows that

$$\begin{aligned} \pi_A \sigma \iota_A &= \lambda(0 \oplus \text{id}_A) \sigma(\text{id}_A \oplus 0) \rho^{-1} && \text{(def. of } \pi_A \text{ and } \iota_A) \\ &= \lambda(0 \oplus \text{id}_A)(0 \oplus \text{id}_A) \sigma \rho^{-1} && \text{(naturality of } \sigma) \\ &= \lambda(0 \oplus \text{id}_A) \sigma \rho^{-1} && \text{(functoriality of } \oplus) \\ &= \lambda \sigma \rho^{-1} && (\text{id}_Z = 0, \text{ nat. of } \lambda) \\ &= \text{id}_A && \text{(coherence theorem)} \end{aligned}$$

with the last step due to the coherence theorem of symmetric monoidal categories, which establishes that if two morphisms have the same type and both are built from composition and monoidal product of $\{\text{id}, \alpha, \lambda, \rho, \sigma\}$ and their inverses, then the two morphisms are equivalent (see Corollary 1.42 from [31]). Similarly, $\pi_B \sigma \iota_B = \text{id}_B$, whereas $\pi_A \sigma \iota_B = 0$ and $\pi_B \sigma \iota_A = 0$. Therefore, the unique decomposition of the symmetric braiding is:

$$\sigma = \begin{pmatrix} 0 & \text{id}_B \\ \text{id}_A & 0 \end{pmatrix}.$$

These remarks together with the monoidal unit being a zero object and composition being Σ -bilinear are *almost* sufficient for Σ_* -UDCs to be partially traced categories with respect to the execution formula. This is made precise by the following lemma.

Lemma 3.2.10. *Let $(\mathbf{C}, \oplus, Z)$ be a Σ_* -UDC. Then, for any morphism $f \in \mathbf{C}(A \oplus U, B \oplus U)$, the execution formula*

$$\text{ex}^U(f) \simeq \Sigma \left(\{f_{BA}\} \uplus \{f_{BU} f_{UU}^n f_{UA}\}_{\mathbb{N}} \right)$$

is guaranteed to satisfy all of the axioms of partially traced categories but vanishing II.

Proof. Each of the axioms of partially traced categories (except vanishing II) are proven below.

- *Naturality.* For all $f: A \oplus U \rightarrow B \oplus U$, if $\text{ex}^U(f)$ is defined then:

$$\begin{aligned} h \circ \text{ex}^U(f) \circ g &\simeq h \circ \left(\Sigma \left(\{f_{BA}\} \uplus \{f_{BU} f_{UU}^n f_{UA}\}_{\mathbb{N}} \right) \right) \circ g && \text{(def. of ex)} \\ &\simeq \Sigma \left(\{h f_{BA} g\} \uplus \{h f_{BU} f_{UU}^n f_{UA} g\}_{\mathbb{N}} \right) && (\circ \text{ is } \Sigma\text{-bilinear}) \\ &\simeq \text{ex}^U((h \oplus \text{id}) \circ f \circ (g \oplus \text{id})) && \text{(def. of ex)} \end{aligned}$$

so that naturality is satisfied.

- *Dinaturality.* For all $f: A \oplus U \rightarrow B \oplus U'$ and $g: U' \rightarrow U$, if $\text{ex}^U((\text{id} \oplus g)f)$ is defined then:

$$\begin{aligned} \text{ex}^U((\text{id} \oplus g) \circ f) &\simeq \Sigma \left(\{f_{BA}\} \uplus \{f_{BU}(g f_{U'U})^n (g f_{U'A})\}_{\mathbb{N}} \right) && \text{(def. of ex)} \\ &= \Sigma \left(\{f_{BA}\} \uplus \{(f_{BU}g)(f_{U'U}g)^n f_{U'A}\}_{\mathbb{N}} \right) && \text{(associativity of } \circ) \\ &\simeq \text{ex}^{U'}(f \circ (\text{id} \oplus g)). && \text{(def. of ex)} \end{aligned}$$

Similarly, if $\text{ex}^{U'}(f \circ (\text{id} \oplus g))$ is defined then $\text{ex}^U((\text{id} \oplus g) \circ f)$ is defined as well and they coincide, so that dinaturality is satisfied.

- *Superposing.* For all $f: A \oplus U \rightarrow B \oplus U$ and $g: C \rightarrow D$, if $\text{ex}^U(f)$ is defined then, according to the matrix decomposition of $g \oplus f$ (see Remark 3.2.8):

$$\text{ex}^U(g \oplus f) = \text{ex}^U \begin{pmatrix} g & 0 \\ 0 & f \end{pmatrix} \simeq \begin{pmatrix} \Sigma \left(\{g\} \uplus \{0 f_{UU}^n 0\}_{\mathbb{N}} \right) & 0 \\ 0 & \text{ex}^U(f) \end{pmatrix} \simeq g \oplus \text{ex}^U(f)$$

where the neutral element and singleton axioms of weak Σ -monoids have been used. Thus, superposing is satisfied.

- *Yanking.* The axiom follows trivially from the matrix decomposition of the symmetry $\sigma: U \oplus U \rightarrow U \oplus U$ (see Remark 3.2.9):

$$\text{ex}^U(\sigma) = \text{ex}^U \begin{pmatrix} 0 & \text{id}_U \\ \text{id}_U & 0 \end{pmatrix} \simeq \Sigma \left(\{0\} \uplus \{\text{id}_U 0^n \text{id}_U\}_{\mathbb{N}} \right) \simeq \text{id}_U$$

where the neutral element and singleton axioms of weak Σ -monoids have been used. Thus, yanking is satisfied.

- *Vanishing I.* For all $f: A \oplus Z \rightarrow B \oplus Z$ notice that $f_{ZA} = 0$, $f_{BZ} = 0$ and $f_{ZZ} = 0$ due to Z being a zero object, implying:

$$\text{ex}^Z(f) \simeq f_{BA} = \pi_B f \iota_A = \rho(\text{id}_B \oplus 0)f(\text{id}_A \oplus 0)\rho^{-1} = \rho f \rho^{-1}$$

where the neutral element and singleton axioms of weak Σ -monoids have been used. Thus, vanishing I is satisfied. $\square$

The proof above uses that composition is Σ -bilinear along with the neutral element and singleton axioms of weak Σ -monoids. However, neither the bracketing nor the flattening axioms of weak Σ -monoids are used. The theorem below establishes that, in the case of Σ_s -UDCs, the vanishing II axiom is also satisfied. The proof relies on the strong bracketing and the strong flattening axioms from the Σ_s -enrichment. The same result was proven by Hoshino in [33] using a different strategy: Hoshino provides a representation theorem for Σ_s -UDCs, establishing that every such category embeds in a categories with countable biproducts, then derives a partial trace in the original category using a similar approach to Proposition 3.1.3. In contrast, our proof illustrates the importance of strong bracketing and strong flattening explicitly and, hence, hints at the challenge of generalising this result to Σ_w -UDCs where these axioms are not available. For further discussion on the connection between our work and Hoshino's, see Section 3.5.

Theorem 3.2.11. *Every Σ_s -UDC $(\mathbf{C}, \oplus, Z)$ is partially traced using the execution formula.*

Proof. According to the previous lemma, all axioms except vanishing II are guaranteed to be satisfied. Thus, we only need to show that, whenever the enrichment is over Σ_s , vanishing II is satisfied as well. Let $f: A \oplus U \oplus V \rightarrow B \oplus U \oplus V$ be a morphism in $\mathbf{C}$ such that $\text{ex}^V(f)$ is defined. For each $n \in \mathbb{N}$, let $\mathbf{d}_n \in \mathbf{C}(A, B)^*$ be the family of morphisms $A \rightarrow B$ built from chains of compositions of the form:

$$f_{BW_n} \circ \dots \circ f_{W_2 W_1} \circ f_{W_1 A}$$

where each W_i may be either U or V ; in the case of $n = 0$, define $\mathbf{d}_0 = \{f_{BA}\}$. Let $\mathbf{p} = \uplus_{n \in \mathbb{N}} \mathbf{d}_n$ and, for each $n \in \mathbb{N}$, let $\mathbf{q}_n$ be the subfamily of $\mathbf{p}$ comprised of all morphisms built from chains that factor through U exactly n times; evidently, $\mathbf{p} = \uplus_{n \in \mathbb{N}} \mathbf{q}_n$. Notice that for $n > 0$ the sum of $\mathbf{d}_n$ would correspond to:

$$\Sigma \mathbf{d}_n \simeq \begin{pmatrix} f_{BU} & f_{BV} \end{pmatrix} \begin{pmatrix} f_{UU} & f_{UV} \\ f_{VU} & f_{VV} \end{pmatrix}^{n-1} \begin{pmatrix} f_{UA} \\ f_{VA} \end{pmatrix}$$

and thus $\mathbf{d}_n$ is summable according to Proposition 3.2.7. Since $\Sigma \mathbf{d}_0 = f_{BA}$, the definition of $\text{ex}^{U \oplus V}$ implies that:

$$\text{ex}^{U \oplus V}(f) \simeq \Sigma \{\Sigma \mathbf{d}_n\}_{\mathbb{N}}. \quad (3.10)$$

Similarly, $\Sigma \mathbf{q}_0 \simeq \pi_B \circ \text{ex}^V(f) \circ \iota_A$ due to composition being Σ -bilinear and, for each $n > 0$,

$$\Sigma \mathbf{q}_n \simeq (\pi_B \text{ex}^V(f) \iota_U) \circ (\pi_U \text{ex}^V(f) \iota_U)^{n-1} \circ (\pi_U \text{ex}^V(f) \iota_A)$$

where each family $\mathbf{q}_n$ is summable due to $\text{ex}^V(f)$ being defined (by assumption) and composition being Σ -bilinear. Then, according to the definition of ex^U :

$$\text{ex}^U(\text{ex}^V(f)) \simeq \Sigma\{\Sigma\mathbf{q}_n\}_{\mathbb{N}}. \quad (3.11)$$

In summary, we have provided two partitions of the family $\mathbf{p}$, both satisfying that for all $n \in \mathbb{N}$ each subfamily $\mathbf{d}_n$ and $\mathbf{q}_n$ is summable. Moreover, we have shown that the family of their sums is defined iff either side of the vanishing II axiom is satisfied; it is now straightforward to prove vanishing II using the strong flattening and strong bracketing axioms from the Σ_s -enrichment. Assume $\text{ex}^U(\text{ex}^V(f))$ is defined, then it follows that:

$$\begin{aligned} \text{ex}^U(\text{ex}^V(f)) &\simeq \Sigma\{\Sigma\mathbf{q}_n\}_{\mathbb{N}} && \text{(equation (3.11))} \\ &\simeq \Sigma\mathbf{p} && \text{(strong flattening)} \\ &\simeq \Sigma\{\Sigma\mathbf{d}_n\}_{\mathbb{N}} && \text{(strong bracketing)} \\ &\simeq \text{ex}^{U \oplus V}(f). && \text{(equation (3.10))} \end{aligned}$$

Similarly, if we assume that $\text{ex}^{U \oplus V}(f)$ is defined it follows that $\text{ex}^U(\text{ex}^V(f))$ is defined as well and their results coincide. Therefore, vanishing II is satisfied and the proof is complete. $\square$

An immediate question is whether this theorem can be generalised to Σ_w -UDCs. Unfortunately, these are generally not partially traced since vanishing II cannot be guaranteed to hold, as shown in the following counterexample.

Example 3.2.12. Let the Σ_g -enrichment of $\mathbf{FdHilb}$ be induced by the usual operator norm topology (see Proposition 3.2.25 for details) and consider the following morphism in $\mathbf{FdHilb}$:

$$\begin{aligned} f: \mathbb{C} \oplus \mathbb{C} \oplus \mathbb{C} &\rightarrow \mathbb{C} \oplus \mathbb{C} \oplus \mathbb{C} \\ f &= \begin{pmatrix} 0 & 1 & 1 \\ 1 & -\frac{2}{3} & 1 \\ 1 & 1 & \frac{1}{3} \end{pmatrix}. \end{aligned}$$

Explicit calculation yields the following result:

$$\text{ex}^{\mathbb{C}}(f) = \text{ex}^{\mathbb{C}} \left(\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & -\frac{2}{3} & 1 \\ 1 & 1 & \frac{1}{3} \end{array} \right) = \begin{pmatrix} \frac{3}{2} & \frac{5}{2} \\ \frac{5}{2} & \frac{5}{6} \end{pmatrix}$$

so that $\left|\frac{5}{6}\right| \leq 1$ and, hence, $\text{ex}^{\mathbb{C}}(\text{ex}^{\mathbb{C}}(f))$ is well-defined, whereas

$$\text{ex}^{\mathbb{C} \oplus \mathbb{C}}(f) = \text{ex}^{\mathbb{C} \oplus \mathbb{C}} \left(\begin{array}{c|cc} 0 & 1 & 1 \\ \hline 1 & -\frac{2}{3} & 1 \\ 1 & 1 & \frac{1}{3} \end{array} \right) = \sum_{n=0}^{\infty} \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} -\frac{2}{3} & 1 \\ 1 & \frac{1}{3} \end{pmatrix}^n \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

does not converge in norm and, hence, is undefined. Therefore, this morphism is an instance of vanishing II failing in a Σ_g -UDC.

The construction of such a counterexample is enabled by the presence of additive inverses; in particular, entry $-\frac{2}{3}$ in the definition of f . The goal of the rest of this section is to establish sufficient conditions for vanishing II to be satisfied in Σ_w -UDCs. But first, we discuss the relevance of the result for Σ_s -UDCs (Theorem 3.2.11) in the context of classical iterative loops.

3.2.1 Classical iterative loops

This section provides three examples of Σ_s -UDCs which, according to Theorem 3.2.11, are partially traced with respect to the execution formula. The relevance of each of these categories in classical and quantum computing is briefly discussed and, in particular, it is argued that their execution formula captures the semantics of classical iterative loops in their respective frameworks.

Classical reversible programs. A bijection $f: A \rightarrow B$ is interpreted to describe the input-output behaviour of a reversible program whose possible inputs and outputs are elements of the sets A and B , respectively.

Definition 3.2.13. Let **Bijection** be the category whose objects are sets and whose morphisms are bijections. For any partial function $f: A \rightarrow B$ let $\text{dom}(f) \subseteq A$ be the subset where f is defined. Let **PInj** be the category whose objects are sets and whose morphisms $f \in \mathbf{PInj}(A, B)$ are partial injective functions, *i.e.* partial functions whose restriction $\text{dom}(f) \rightarrow B$ is injective.

Proposition 3.2.14 (Haghverdi [28]). *Let $(\mathbf{PInj}, \uplus, \emptyset)$ be the symmetric monoidal category induced by disjoint union. A family of morphisms $\{f_i\}_I \in \mathbf{PInj}(A, B)^*$ is summable iff $\text{dom}(f_i) \cap \text{dom}(f_{i'}) = \emptyset$ and $\text{im}(f_i) \cap \text{im}(f_{i'}) = \emptyset$ for all $i \neq i'$; in such a case, the partial function $\Sigma\{f_i\}_I: A \rightarrow B$ is defined as follows for every $a \in A$:*

$$(\Sigma\{f_i\}_I)(a) = \begin{cases} f_i(a) & \text{if } \exists i \in I \text{ s.t. } a \in \text{dom}(f_i) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

These definitions make $\mathbf{PInj}$ a Σ_s -UDC.

Proof. (Sketch). We first check that each hom-set $\mathbf{PInj}(A, B)$ is a strong Σ -monoid using the definition of Σ given above. The neutral element of the Σ -monoid is the partial function that is undefined on every $a \in A$ and the singleton axiom holds trivially. Strong flattening holds as well: let $\{f_j \in \mathbf{PInj}(A, B)^*\}_J$ be an indexed set of summable families and let $f = \uplus_J f_j$; assume that the family $\{\Sigma f_j\}_J$ is summable, this implies that $\text{dom}(\Sigma f_j) \cap \text{dom}(\Sigma f_{j'}) = \emptyset$ for every $j \neq j'$ and hence, necessarily, the domains of the elements of $f_j \uplus f_{j'}$ are all pair-wise disjoint. Since this applies to every pair of indices $j \neq j'$ in J and a similar argument holds for the images, it follows that $f = \uplus_J f_j$ is summable and it is trivial to check that $\Sigma f = \Sigma\{\Sigma f_j\}_J$ so that strong flattening is satisfied. The proof of the strong bracketing axiom is dual to this argument. Finally, subsummability holds trivially since the summability condition is imposed on every pair of elements from the family.

To show that $\mathbf{PInj}$ is a Σ_s -enriched category we must check that composition is Σ -bilinear. Assume $\{f_i\}_I \in \mathbf{PInj}(A, B)^*$ is summable and let $g \in \mathbf{PInj}(B, C)$; then $\{g \circ f_i\}_I$ trivially satisfies the condition on disjoint domains. That $\{g \circ f_i\}_I$ also satisfies the condition on disjoint images can be shown by contradiction: suppose there is an element $c \in \text{im}(g \circ f_i) \cap \text{im}(g \circ f_{i'})$ for $i \neq i'$ then there must be elements $b \in \text{im}(f_i)$ and $b' \in \text{im}(f_{i'})$ such that $g(b) \simeq c \simeq g(b')$, but $b \neq b'$ is necessary for $\{f_i\}_I$ to be summable, so this contradicts g being injective. A similar argument can be used to prove that composition is a Σ -homomorphism on the left as well, making it Σ -bilinear.

It is straightforward to check that $(\mathbf{PInj}, \uplus, \emptyset)$ is a symmetric monoidal category and its monoidal unit $\emptyset$ is a zero object. Moreover, $\Sigma\{\text{id}_A \uplus 0, 0 \uplus \text{id}_B\} \simeq \text{id}_{A \uplus B}$ holds trivially. Thus, $\mathbf{PInj}$ is a Σ_s -UDC, as claimed. $\square$

Notice that $\mathbf{PInj}$ does not have coproducts since the codiagonal map $\nabla: A \uplus A \rightarrow A$ cannot be injective because it would need to map both $\iota_l(a), \iota_r(a) \in A \uplus A$ to $a \in A$. And, evidently, $\uplus$ is not a categorical product; nevertheless, morphisms $f: A \uplus B \rightarrow C \uplus D$ in $\mathbf{PInj}$ may uniquely be characterised by their matrix decomposition thanks to Proposition 3.2.6. According to Theorem 3.2.11, the category $(\mathbf{PInj}, \uplus, \text{ex})$ is a partially traced and, furthermore, we can show that it is totally traced. For an arbitrary morphism $f \in \mathbf{PInj}(A \uplus U, B \uplus U)$, let $p_0 = f_{BA}$ and for all $n \in \mathbb{N}$ let $p_{n+1} = f_{BU} f_{UU}^n f_{UA}$. It can be shown by induction that no element $b \in B$ can be both in $\text{im}(p_n)$ and $\text{im}(p_m)$ for $n \neq m$ since this would contradict f being injective; similarly, no $a \in A$ can be both in $\text{dom}(p_n)$ and $\text{dom}(p_m)$. Consequently, the family $\{p_n\}_{\mathbb{N}}$ is

always summable and **PInj** is in fact totally traced (see Example 2.23 (i) from [28] for more details).

Even though **Bijection** itself is not a UDC — quasi-projections are necessarily partial — it is a subcategory of **PInj**; therefore, every bijection $f: A \uplus B \rightarrow C \uplus D$ may be decomposed into a matrix form whose entries are in **PInj**. We may transport the trace of **PInj** to $(\mathbf{Bijection}, \uplus, \widehat{\text{ex}})$ using Proposition 3.1.3 which defines:

$$\widehat{\text{ex}}^U(f) = \begin{cases} g & \text{if } \text{ex}^U(f) \text{ is a bijection} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

If f denotes a reversible program then the bijection $\widehat{\text{ex}}^U(f): A \rightarrow B$ denotes a program that repeatedly applies f whenever the output is in U — an iterative loop. Notice that **Bijection** is only partially traced; for instance, if $U = \mathbb{N}$ we may define a bijection so that $f_{UU}(n) = n + 1$ for all $n \in \mathbb{N}$ and, for some $a \in A$, $f_{UA}(a) = 0$ so that $a \notin \text{dom}(\text{ex}^U(f))$ and $\text{ex}^U(f)$ is only a partial injection. The latter case would correspond to an iterative loop that never halts if the input is a .

Classical probabilistic programs. On each input $a \in A$, a probabilistic program may yield an output $b \in B$ with probability $p_a(b)$. These probabilities may be arranged in a $|B| \times |A|$ matrix M with entries $m_{ba} = p_a(b)$ so that each of its columns add up to one:

$$\sum_{b \in B} m_{ba} = \sum_{b \in B} p_a(b) = 1.$$

These are known as *stochastic* matrices. If the sum of each of their columns is only required to be bounded by one $\sum_{b \in B} m_{ba} \leq 1$ we refer to them as *substochastic* matrices. Evidently, every stochastic matrix is substochastic.

Definition 3.2.15. Let **SubStoch** be the category whose objects are finite sets and whose morphisms $f \in \mathbf{SubStoch}(A, B)$ are substochastic matrices $|B| \times |A|$. Composition corresponds to matrix multiplication and identities correspond to the usual identity matrices. Let **Stoch** be the subcategory of **SubStoch** whose morphisms are restricted to be stochastic matrices.

Similarly to the case with **PInj** and **Bijection**; the category we are interested in is **Stoch**, but we instead work on the larger category **SubStoch** that happens to be a Σ_s -UDC, as established below.

Proposition 3.2.16. *Let $(\mathbf{SubStoch}, \oplus, \emptyset)$ be the symmetric monoidal category where $\oplus$ acts as disjoint union of sets and direct sum of matrices. A family of morphisms $\{f_i\}_I \in \mathbf{SubStoch}(A, B)^*$ is summable iff the standard sum of matrices⁴ yields a substochastic matrix, in which case $\Sigma\{f_i\}_I$ is defined to be the corresponding result. These definitions make $\mathbf{SubStoch}$ a Σ_s -UDC.*

Proof. (Sketch). We first check that each hom-set $\mathbf{SubStoch}(A, B)$ is a strong Σ -monoid using the definition of Σ given above. Its neutral element is the matrix with all zero entries and the singleton axiom is trivially satisfied. Due to the lack of negative entries, both strong bracketing and strong flattening are immediately implied by the fact that a converging series of positive real numbers is convergent no matter the ordering of the sequence; subsummability is also straightforward to check.

To show that $\mathbf{SubStoch}$ is a Σ_s -enriched category we must check that composition is Σ -bilinear. If $\{f_i\}_I \in \mathbf{SubStoch}(A, B)^*$ is summable then $g \circ \Sigma\{f_i\}_I$ is a substochastic matrix for all $g \in \mathbf{SubStoch}(B, C)$. Given that matrix multiplication distributes over matrix addition, the family $\{g \circ f_i\}_I$ is summable, with the result being the substochastic matrix $g \circ \Sigma\{f_i\}_I$. A similar argument can be used to prove that composition is a Σ -homomorphism on the left as well, making it Σ -bilinear.

It is straightforward to check that $(\mathbf{SubStoch}, \oplus, \emptyset)$ is a symmetric monoidal category and its monoidal unit $\emptyset$ is a zero object. Moreover, $\Sigma\{\text{id}_A \oplus 0, 0 \oplus \text{id}_B\} \simeq \text{id}_{A \oplus B}$ holds trivially. Thus, $\mathbf{SubStoch}$ is a Σ_s -UDC, as claimed. $\square$

In the case of $\mathbf{SubStoch}$ objects $A \oplus B$ are coproducts; however, $A \oplus B$ is not a categorical product since the diagonal morphism $\Delta: A \rightarrow A \oplus A$ would need to be

$$\Delta = \begin{pmatrix} \text{id}_A \\ \text{id}_A \end{pmatrix}$$

which is not a substochastic matrix — each of its columns add up to 2.

According to Theorem 3.2.11, the category $(\mathbf{SubStoch}, \oplus, \text{ex})$ is partially traced. Moreover, it is straightforward to check that, whenever U is a singleton set, $\text{ex}^U(f)$ is well-defined (due to the convergence criteria of the geometric series); then, due to objects being finite sets, we may trace out each element of any arbitrary set U one by one and, after to a finite number of applications of vanishing II:

$$\text{ex}^U(f) \simeq \text{ex}^{\{u\}}(\text{ex}^{\{u'\}}(\dots(f)))$$

⁴An infinite family of matrices is summable iff it is entry-wise summable according to the notion of absolute convergence on real numbers (see Example 2.1.5).

so that we conclude $(\mathbf{SubStoch}, \oplus, \text{ex})$ is totally traced. Using Proposition 3.1.3 we can induce a partial trace on $\mathbf{Stoch}$ using ex from $\mathbf{SubStoch}$. Notice that the resulting partially traced category $(\mathbf{Stoch}, \oplus, \widehat{\text{ex}})$ is not total since

$$\widehat{\text{ex}}^u \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} = 0 + \sum_{n=0}^{\infty} 0 \cdot (1)^n \cdot 1 = 0.$$

Quantum programs with classical control flow. The category $\mathbf{CPTP}$ was introduced in Section 1.5.1 to capture both coherent quantum operations and non-coherent ones (*e.g.* measurements) in a single framework. Along it, $\mathbf{CPTR}$ was introduced to take up a similar role $\mathbf{SubStoch}$ plays with respect to $\mathbf{Stoch}$. The objects in $\mathbf{CPTR}$ are finite-dimensional C^* -algebras whose density operators represent *mixed* quantum states: probability distributions of *pure* quantum states. It was discussed in Section 1.5.1 that for any two C^* -algebras A and B , if $(\rho, \rho') \in A \oplus B$ is a density operator, then (ρ, ρ') may be interpreted as probability distributions comprised of outcome $\frac{\rho}{\text{tr}(\rho)}$ with probability $\text{tr}(\rho)$ and outcome $\frac{\rho'}{\text{tr}(\rho')}$ with probability $\text{tr}(\rho')$. Consequently, a morphism $f \in \mathbf{CPTR}(A, B \oplus C)$ may be understood to apply either $f_{BA}: A \rightarrow B$ or $f_{CA}: A \rightarrow C$ with some probability. Intuitively, it follows that the execution formula in $\mathbf{CPTR}$ will correspond to the sum of probabilistic paths and, to a certain extent, it will be similar to the case of $\mathbf{SubStoch}$.

Proposition 3.2.17. *A family of CPTR maps $\{f_i\}_I \in \mathbf{CPTR}(A, B)^*$ is summable iff its pointwise sum⁵ yields a CPTR map, in which case $\Sigma\{f_i\}_I$ is defined to be the corresponding result. Then, the monoidal category $(\mathbf{CPTR}, \oplus, \{0\})$ is a Σ_s -UDC.*

Proof. (Sketch). We first check that each hom-set $\mathbf{CPTR}(A, B)$ is a strong Σ -monoid using the definition of Σ given above. Its neutral element is the map that sends any $\rho \in A$ to the zero operator in B ; the singleton axiom is trivially satisfied. Notice that the pointwise addition of CPTR maps does not have additive inverses: considering that morphisms $f, g \in \mathbf{CPTR}(A, B)$ send positive elements $\rho \in A$ to positive elements in B , the pointwise addition of CPTR maps satisfies

$$(f + g)(\rho) = f(\rho) + g(\rho) \geq 0.$$

⁵A definition in terms of pointwise sums requires that the codomain B has a notion of summability for infinite families. For finite families, addition is given by the standard addition in B and, since B is a finite-dimensional C^* -algebra, it can be regarded as the direct sum of a finite collection of finite-dimensional Hilbert spaces. Thus, the notion of infinite summability assigned to B corresponds to the usual one derived from operator norm convergence in $\mathbf{FdHilb}$ (see Proposition 3.2.25 for further details).

Then, it follows that strong flattening and strong bracketing are implied by commutativity and associativity of addition. Subsummability is satisfied as well since if $\Sigma\{f_i\}_I$ is trace-reducing, summing up fewer maps $J \subseteq I$ means that $\Sigma\{f_j\}_J$ is guaranteed to be trace-reducing.

It is immediate that composition distributes over pointwise addition, so that composition is Σ -bilinear. It is evident that $\{0\}$ is a zero object and $\Sigma\{\text{id}_A \oplus 0, 0 \oplus \text{id}_B\} \simeq \text{id}_{A \oplus B}$ holds:

$$\Sigma\{\text{id}_A \oplus 0, 0 \oplus \text{id}_B\}(a, b) = (\text{id}_A \oplus 0)(a, b) + (0 \oplus \text{id}_B)(a, b) = (a, 0) + (0, b) = (a, b).$$

Thus, **CPTR** is a Σ_s -UDC, as claimed. $\square$

Objects $A \oplus B$ in **CPTR** are coproducts; however, $A \oplus B$ is not a categorical product since the diagonal morphism $\Delta: A \rightarrow A \oplus A$ is not trace-reducing. According to Theorem 3.2.11, the category $(\mathbf{CPTR}, \oplus, \text{ex})$ is partially traced. In fact, **CPTR** is totally traced, as previously shown by Selinger in [49] (where **CPTR** is denoted **Q**). The trace given in [49] is not exactly ex , in the sense that it is not defined in terms of Σ -monoids, but rather in terms of recursive applications of binary addition; nevertheless, it is apparent that the two notions of trace coincide in their result. Alternatively, it may be shown that $(\mathbf{CPTR}, \oplus, \text{ex})$ is totally traced employing a similar argument to the one used for **SubStoch**, relying on the fact that objects in **CPTR** are finite-dimensional. The proof is cumbersome due to the need to manage **CPTR** maps, but the intuition is the same.

We may transport the trace in **CPTR** to $(\mathbf{CPTP}, \oplus, \widehat{\text{ex}})$ using Proposition 3.1.3 which defines:

$$\widehat{\text{ex}}^U(f) = \begin{cases} g & \text{if } \text{ex}^U(f) \text{ is trace-preserving} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

As in the case of **Stoch**, this is not a totally traced category.

The fact that **CPTR** is Σ_s -enriched tells us that there is no interference between the paths being added up in the execution formula; thus, this is a clear example of “quantum data, classical control”. In contrast, in a quantum iterative loop we would expect that paths may cancel out with each other, allowing for additive inverses of morphisms in the category, preventing it from being Σ_s -enriched. Such is the case of **Contraction** as will be shown in what remains of this chapter.

Some traced monoidal functors. There is an obvious faithful functor $M : \mathbf{FinPInj} \rightarrow \mathbf{SubStoch}$ where $\mathbf{FinPInj}$ is the subcategory of $\mathbf{PInj}$ where objects are restricted to finite sets. The functor M acts as the identity on objects and maps partial injections $f : A \rightarrow B$ to the matrix $M(f)$ of entries:

$$m_{ba} = \begin{cases} 1 & \text{if } b = f(a) \\ 0 & \text{otherwise.} \end{cases}$$

It is trivial to check that $\sum_{b \in B} m_{ba} = 1$ if $a \in \text{dom}(f)$ whereas $\sum_{b \in B} m_{ba} = 0$ otherwise, so that the matrix $M(f)$ is indeed substochastic. It is straightforward to check that M is a strict monoidal functor and, moreover, for each family $\mathbf{f} \in \mathbf{PInj}(A, B)^*$ we have that:

$$\Sigma \mathbf{f} \simeq f \implies \Sigma M \mathbf{f} \simeq M(f)$$

so that the action of M on hom-sets is a Σ -homomorphism. Then, it is immediate to show that $M : (\mathbf{FinPInj}, \uplus, \text{ex}) \rightarrow (\mathbf{SubStoch}, \uplus, \text{ex})$ is a traced monoidal functor. Intuitively, this corresponds to the fact that reversible (hence, deterministic) programs are a subclass of probabilistic programs.

On a similar note, there is a faithful functor $P : \mathbf{SubStoch} \rightarrow \mathbf{CPTR}$ mapping finite sets A to C^* -algebras $\oplus_{a \in A} B(\mathbb{C})$. The details of how P acts on morphisms are rather verbose to give explicitly, but the concept is simple: an element $(p_a)_A \in \oplus_{a \in A} B(\mathbb{C})$ that is a positive operator can be represented as a vector of positive real numbers and, if it has trace one (*i.e.* if it is a density operator) then $\sum_{a \in A} p_a = 1$. Consequently, the elements in the C^* -algebra $P(A)$ correspond precisely to probability distributions over A and it is straightforward to see how multiplying a substochastic matrix $f \in \mathbf{SubStoch}(A, B)$ with the vector $(p_a)_A \in \mathbb{R}^{|A|}$ would result in a sub-probability⁶ distribution over B and, consequently, a positive operator in $P(B)$ with trace at most one. Following this intuition, it can be shown that substochastic matrices $A \rightarrow B$ induce CPTR maps $P(A) \rightarrow P(B)$. It is immediate from the definition of P on objects that P is a strict monoidal functor. Since summability in $\mathbf{CPTR}$ is defined pointwise, it is straightforward to show that the action of P on hom-sets is a Σ -homomorphism. Consequently, it follows that $P : (\mathbf{SubStoch}, \uplus, \text{ex}) \rightarrow (\mathbf{CPTR}, \oplus, \text{ex})$ is a traced monoidal functor. Intuitively, this corresponds to the fact that classical probabilistic programs are a subclass of quantum programs with classical control flow.

⁶In a sub-probability distribution the probabilities are only required to sum up to a value smaller than or equal to one.

3.2.2 Towards an execution formula with additive inverses

The goal of the following sections is to prove that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a totally traced category. Lemma 3.2.10 established that the execution formula on every Σ_* -UDC automatically satisfies all axioms of partially traced categories but vanishing II. The Σ_g -UDC $(\mathbf{FdHilb}, \oplus, \{0\})$ was shown *not* to be a partially traced category with respect to the execution formula, due to vanishing II failing. The importance of vanishing II is made apparent in the discussion from the previous section where, for instance, **SubStoch** was shown to be totally traced via an induction argument using vanishing II. In Section 3.2.4 sufficient conditions for the execution formula to satisfy vanishing II are presented. But first, we shall explain why the proof strategy from Theorem 3.2.11 — which establishes that Σ_s -UDCs satisfy vanishing II and are, hence, partially traced — cannot be used in the general case where additive inverses are allowed. The obvious answer is that, in general, weak Σ -monoids do not satisfy strong flattening and strong bracketing, which are at the core of the proof of Theorem 3.2.11. In essence, the proof made use of the fact that $\text{ex}^U(\text{ex}^V(f))$ and $\text{ex}^{U \oplus V}(f)$ add up the same morphisms, but they do so in different arrangements. Unfortunately, it turns out that there are examples of morphisms f in **FdContraction** where both $\text{ex}^U(\text{ex}^V(f))$ and $\text{ex}^{U \oplus V}(f)$ are well-defined and coincide but the ‘flattened’ family is not summable and, hence, a proof strategy via the flattening and bracketing axioms will not work. Such an example is given below.

In **FdContraction**, a family of morphisms $A \rightarrow B$ will be considered summable iff the sum of their operator norms converges (see Proposition 3.2.26 for more details); in the particular case of morphisms $\mathbb{C} \rightarrow \mathbb{C}$ this corresponds to absolute convergence. Let $f: \mathbb{C} \oplus \mathbb{C} \oplus \mathbb{C} \rightarrow \mathbb{C} \oplus \mathbb{C} \oplus \mathbb{C}$ be the following contraction:⁷

$$f = \begin{pmatrix} f_{00} & f_{01} & f_{02} \\ f_{10} & f_{11} & f_{12} \\ f_{20} & f_{21} & f_{22} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & 1 \end{pmatrix}.$$

Multiplying the lower right 2×2 block of f with itself results in $\frac{1}{2} \text{id}$, so we can easily

⁷An easy way to check that this is indeed a contraction is to realise that it corresponds to the lower right 3×3 block of the matrix $H \otimes H$ where H is the unitary matrix:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

calculate $\text{ex}^{\mathbb{C} \oplus \mathbb{C}}(f)$:

$$\begin{aligned} \text{ex}^{\mathbb{C} \oplus \mathbb{C}}(f) &= f_{00} + \sum_{n=0}^{\infty} \begin{pmatrix} f_{01} & f_{02} \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix}^n \begin{pmatrix} f_{10} \\ f_{20} \end{pmatrix} \\ &= -\frac{1}{2} + \frac{1}{2} \sum_{n \text{ even}} \left(\frac{1}{2}\right)^{\frac{n}{2}} + \frac{1}{4} \sum_{n \text{ odd}} \left(\frac{1}{2}\right)^{\frac{n-1}{2}} \\ &= -\frac{1}{2} + \frac{1}{2} \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^n + \frac{1}{4} \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^n = 1. \end{aligned}$$

On the other hand, we have that:

$$\text{ex}^{\mathbb{C}}(f) = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} + \frac{1}{4} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \left(\sum_{n=0}^{\infty} \frac{1}{2^n} \right) \begin{pmatrix} -1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

so that $\text{ex}^{\mathbb{C}}(\text{ex}^{\mathbb{C}}(f)) = 0 + \sum_{n=0}^{\infty} 1(0)^n 1 = 1$. Thus, $\text{ex}^{\mathbb{C}}(\text{ex}^{\mathbb{C}}(f)) \simeq \text{ex}^{\mathbb{C} \oplus \mathbb{C}}(f)$ as required by vanishing II. In contrast, their ‘flattened’ family contains all $\mathbb{C} \rightarrow \mathbb{C}$ morphisms arising from chains of the form:

$$f_{0w_n} \circ \dots \circ f_{w_2 w_1} \circ f_{w_1 0}$$

where each w_i is either subscript 1 or 2 and n is an arbitrary positive integer. It is straightforward to check that, in general, there are 2^{k-1} chains of length k and each chain of length k has absolute value $\frac{1}{2^k}$, consequently, the sum of absolute values of all elements in the flattened family is

$$\sum_{k=1}^{\infty} \frac{1}{2^k} \cdot 2^{k-1} = \sum_{k=1}^{\infty} \frac{1}{2}$$

and, hence, the sum is not absolutely convergent and the flattened family is not summable. Thus, such a morphism f provides an example where the correctness of vanishing II cannot be shown by means of the flattening and bracketing axioms. Instead, the following subsections build upon the definition of the kernel-image trace and the notion of limit from Hausdorff commutative monoids to provide sufficient conditions for vanishing II to be satisfied in a Σ_w -UDC.

3.2.3 Σ_g -UDCs and their kernel-image trace

Section 3.1.2 discussed the kernel-image trace from [44]: a partial trace available on every $\mathbf{Ab}$ -enriched category with finite biproducts. The proof that this is a categorical trace appears in Proposition 3.17 from [44]; in it, the existence of biproducts is only

required so that morphisms of type $A \oplus B \rightarrow C \oplus D$ may be uniquely characterised by their matrix decomposition. However, as discussed in Proposition 3.2.6, the structure of a Σ_* -UDC is sufficient for this unique decomposition to exist. Moreover, every Σ -group (X, Σ) can be seen as an abelian group $(X, \Sigma\{-, -\})$, so we may regard any Σ_g -enriched category as an **Ab**-enriched one by simply forgetting the action of Σ on infinite families. Therefore, it is to be expected that any Σ_g -UDC is partially traced using the kernel-image trace. The proof of this claim is sketched below, it follows the same strategy used by the authors of [44] to prove that additive categories are partially traced.

Notation 3.2.18. To improve readability, given a Σ -group (X, Σ) and elements $x, y \in X$, we use the shorthand $x + y$ and $x - y$ to refer to $\Sigma\{x, y\}$ and $\Sigma\{x, -y\}$ respectively.

Recall that, for an arbitrary morphism $f: A \oplus U \rightarrow B \oplus U$ in $\mathbf{C}$, we write $(k, i) \Vdash \text{Tr}^U(f)$ iff there are morphisms $i: A \rightarrow U$ and $k: U \rightarrow B$ in $\mathbf{C}$ such that the diagram

$$\begin{array}{ccc}
 A & \xrightarrow{\quad i \quad} & U \\
 \searrow f_{UA} & \downarrow \text{id} - f_{UU} & \searrow f_{BU} \\
 & U & \xrightarrow{\quad k \quad} B
 \end{array} \tag{3.12}$$

commutes.

Proposition 3.2.19. *Let $\mathbf{C}$ be a Σ_g -UDC. Then, the kernel-image trace (as defined in Proposition 3.1.8) makes $\mathbf{C}$ a partially traced category.*

Proof. (Sketch). First, let's check that the kernel-image trace Tr^U is well-defined, i.e. it does not depend on the choice of morphisms k and i . Assume $(k, i) \Vdash \text{Tr}^U(f)$ and $(k', i') \Vdash \text{Tr}^U(f)$ then,

$$k \circ f_{UA} = k \circ (\text{id} - f_{UU}) \circ i' = f_{BU} \circ i' = k' \circ f_{UA}$$

and, similarly, $f_{BU} \circ i = f_{BU} \circ i'$, so

$$\text{Tr}^U(f) = f_{BA} + f_{BU} \circ i = f_{BA} + k \circ f_{UA}$$

is well-defined. The proof of each axiom of partially traced categories is sketched below; for further details see Proposition 3.17 from [44].

- *Naturality.* If $(k, i) \Vdash \text{Tr}^U(f)$, then $(hk, ig) \Vdash \text{Tr}^U((h \oplus \text{id})f(g \oplus \text{id}))$ follows trivially, and $\text{Tr}^U((h \oplus \text{id})f(g \oplus \text{id})) = h \circ \text{Tr}^U(f) \circ g$ due to composition being Σ -bilinear.

- *Dinaturality.* If $(k, i) \Vdash \text{Tr}^U((\text{id} \oplus g)f)$, then $(kg, j) \Vdash \text{Tr}^{U'}(f(\text{id} \oplus g))$ where $j = f_{U'A} + f_{U'U'} \circ i$; this follows from the fact that $i = gj$ thanks to the left triangle of diagram (3.12) commuting. That $\text{Tr}^U((\text{id} \oplus g)f) = \text{Tr}^{U'}(f(\text{id} \oplus g))$ follows trivially from the definition of Tr^U , using the witnesses k and kg respectively. The opposite direction of implication can be proven using a similar argument.
- *Superposing.* If $(k, i) \Vdash \text{Tr}^U(f)$, then it is immediate from the matrix decomposition of $g \oplus f$ (see Remark 3.2.8) that $(\iota k, i\pi) \Vdash \text{Tr}^U(g \oplus f)$ and $\text{Tr}^U(g \oplus f) = g \oplus \text{Tr}^U(f)$.
- *Yanking.* The axiom follows trivially from the matrix decomposition of the symmetry σ (see Remark 3.2.9), with $(\text{id}, \text{id}) \Vdash \text{Tr}^U(\sigma)$.
- *Vanishing I.* For every morphism $f: A \oplus Z \rightarrow B \oplus Z$ each of its components f_{ZA} , f_{ZZ} and f_{BZ} are zero morphisms due to Z being a zero object. Then, $(0, 0) \Vdash \text{Tr}^Z(f)$ and $\text{Tr}^Z(f) = f_{BA} + 0$. Moreover, $f_{BA} = \pi_B f \iota_A = \rho f \rho^{-1}$ due to Remark 3.2.3; therefore, $\text{Tr}^Z(f) = \rho f \rho^{-1}$.
- *Vanishing II.* Let $f: A \oplus U \oplus V \rightarrow A \oplus U \oplus V$ and assume $(k, i) \Vdash \text{Tr}^V(f)$; vanishing II follows from uniqueness of the matrix decomposition along with Σ_g -enrichment: if $(i', k') \Vdash \text{Tr}^U(\text{Tr}^V(f))$ then $(i'', k'') \Vdash \text{Tr}^{U \oplus V}(f)$ where $i''_{UA} = i'$, $k''_{BU} = k'$, $i''_{VA} = i_{VA} + i_{VU}i'$ and $k''_{BV} = k_{BV} + k_{UV}k'$. Similarly, if $(i'', k'') \Vdash \text{Tr}^{U \oplus V}(f)$ then $(i''_{UA}, k''_{BU}) \Vdash \text{Tr}^U(\text{Tr}^V(f))$ and, in both cases, $\text{Tr}^U(\text{Tr}^V(f)) = \text{Tr}^{U \oplus V}(f)$ follows from algebraic manipulation using the Σ_g -enrichment. $\square$

The left triangle of diagram (3.12) imposes that $(k, i) \Vdash \text{Tr}^U(f)$ implies $f_{UA} = (\text{id} - f_{UU}) \circ i$. Thanks to the Σ_g -enrichment, composition distributes over addition, and additive inverses are available for every morphism. Then, the previous equation is equivalent to $i = f_{UA} + f_{UU} \circ i$ and we may expand this expression recursively, obtaining:

$$i = f_{UA} + f_{UU} \circ (f_{UA} + f_{UU} \circ i) = \dots \quad (3.13)$$

Thus, for any arbitrary $n \in \mathbb{N}$ we have that $(k, i) \Vdash \text{Tr}^U(f)$ implies:

$$\begin{aligned} \sum_{j=0}^n f_{UU}^j f_{UA} &= i - f_{UU}^{n+1} \circ i \\ \sum_{j=0}^n f_{BU} f_{UU}^j &= k - k \circ f_{UU}^{n+1} \end{aligned}$$

which, when $n \rightarrow \infty$, resembles the infinite sum from the execution formula (3.8). This hints at a strategy to prove the validity of the execution formula as a categorical trace: use the kernel-image trace as a stepping stone, and find out under which circumstances the above equation can be extended to an infinite sum. However, as $n \rightarrow \infty$ these equations may become ill-defined unless we can show that $f_{UU}^n \circ i$ tends to zero. With this situation in mind, Chapter 2 gave a brief introduction to topological monoids, where limits may be taken, and Section 2.2.4 established the relationship between these and Σ -monoids. Continuing this line of work, the following subsection presents a subclass of Σ_g -UDCs whose hom-sets are endowed with a topology, enabling us to use nets and their limits to discuss the convergence of infinite sums.

3.2.4 A limit condition

This section introduces the notion of *hom-convergence UDC* which **Hilb** is an example of. In Section 3.3.2, the special properties of these UDCs will be used to prove that **FdContraction** is a totally traced category with respect to the execution formula. Conceptually, the goal is to study Σ_g -UDCs whose hom-sets are Hausdorff abelian groups, so that the notion of limit is readily available. The first idea that comes to mind is to define **HausAb**-UDCs; unfortunately, the category **HausAb** of Hausdorff abelian groups does not have tensor products and such an enrichment would be ill-defined (see Remark 2.2.42). The alternative presented below is to consider Σ_g -UDCs whose hom-sets are identified with Hausdorff abelian groups and whose composition is continuous. To formalise this notion, we may use the faithful functor $G: \mathbf{HausAb} \rightarrow \Sigma_g$ that maps each $(X, \tau, +) \in \mathbf{HausAb}$ to $(X, \Sigma) \in \Sigma_g$, where Σ is the extended group operation induced by the topology (see Definition 2.2.40).

Definition 3.2.20. Let $\mathbf{C}$ be a Σ_g -UDC. Let Φ be a collection containing, a Σ -isomorphism⁸ $\varphi: \mathbf{C}(A, B) \rightarrow G(X_{A,B})$ for each hom-object $\mathbf{C}(A, B) \in \Sigma_g$, where $X_{A,B}$ is some Hausdorff abelian group. The pair $(\mathbf{C}, \Phi)$ is a *hom-convergence UDC* if composition is continuous in each variable; more precisely: for each triple of objects $A, B, C \in \mathbf{C}$ and morphisms $f \in \mathbf{C}(A, B)$ and $g \in \mathbf{C}(B, C)$, the Σ -homomorphisms

$$\varphi(- \circ f)\varphi^{-1}: G(X_{B,C}) \rightarrow G(X_{A,C})$$

$$\varphi(g \circ -)\varphi^{-1}: G(X_{A,B}) \rightarrow G(X_{A,C})$$

are both in the image of $G: \mathbf{HausAb} \rightarrow \Sigma_g$.

⁸A Σ -isomorphism is simply an isomorphism in a Σ_* category, i.e. a Σ -homomorphism with an inverse Σ -homomorphism.

Recall that composition in a Σ_* -enriched category is a Σ -homomorphism $\mathbf{C}(B, C) \otimes \mathbf{C}(A, B) \rightarrow \mathbf{C}(A, C)$ and, hence, by the definition of the tensor product in Σ_* categories (see Section 2.1.5), both $- \circ f$ and $g \circ -$ are Σ -homomorphisms. Moreover, G acts as the identity on morphisms so, if as required by the definition of hom-convergence UDCs there is some continuous function $\hat{f}$ satisfying that $G(\hat{f}) = \varphi(- \circ f)\varphi^{-1}$, then it is justified to say that $- \circ f$ is continuous. In the framework of hom-convergence UDCs we may recover the notion of limits from topology.

Notation 3.2.21. Recall that $G: \mathbf{HausAb} \rightarrow \Sigma_g$ does not change the underlying set of a given Hausdorff abelian group. Thus, in a hom-convergence UDC $(\mathbf{C}, \Phi)$, every morphism $f \in \mathbf{C}(A, B)$ corresponds to a point $\varphi(f)$ in the corresponding Hausdorff abelian group $X_{A,B}$. Let $\alpha: D \rightarrow \mathbf{C}(A, B)$ be a net; we say that α has a limit point f iff $\lim (\varphi \circ \alpha) \simeq \varphi(f)$ using the notion of limits from the Hausdorff space $X_{A,B}$. Even though (strictly speaking) $\mathbf{C}(A, B)$ is not a Hausdorff abelian group, we recover the notation $\lim \alpha \simeq f$ to indicate that the limit point of α exists and is $f \in \mathbf{C}(A, B)$. The beginning of this section will deal with the limit of certain sequences — recall that a sequence is a net defined on the directed set $(\mathbb{N}, \leq)$. For a sequence $s: \mathbb{N} \rightarrow \mathbf{C}(A, B)$ we will use the shorthand

$$\lim_{n \rightarrow \infty} s(n)$$

to refer to its limit point (if it exists).

It is possible to define a topology on hom-sets of $\mathbf{Hilb}$ so that the category is a hom-convergence UDC. An immediate candidate is the topology induced by the operator norm, but such a topology presents problems in the infinite-dimensional case when we attempt to prove certain results, such as Lemma 3.3.10. Instead, we may use the strong operator topology, defined below. Any reader only interested in the finite-dimensional case may skip to Proposition 3.2.25, where it is shown that the strong operator topology in $\mathbf{FdHilb}$ coincides with the standard operator norm topology.

Definition 3.2.22. Every hom-set $\mathbf{Hilb}(A, B)$ can be assigned a *strong operator topology* denoted τ_{SOT} whose base is comprised of the following open sets:

$$B_{S,\epsilon}^f = \{g \in \mathbf{Hilb}(A, B) \mid \forall a \in S, \|f(a) - g(a)\| < \epsilon\} \quad (3.14)$$

where $f \in \mathbf{Hilb}(A, B)$, S is a finite set of vectors in A and $\epsilon > 0$.

It is not immediate that the definition above provides a valid base; the proof is discussed below. For any $g \in B_{S,\epsilon}^f$ it follows that $\epsilon - \|f(a) - g(a)\| > 0$ for all $a \in S$;

let $\hat{\epsilon}$ be

$$\hat{\epsilon} = \min \{ \epsilon - \|f(a) - g(a)\| \}_{a \in S}.$$

Then, for any $h \in B_{S,\hat{\epsilon}}^g$ and for all $a \in S$ we have that

$$\|g(a) - h(a)\| < \hat{\epsilon} \leq \epsilon - \|f(a) - g(a)\|$$

and using the triangle inequality we obtain:

$$\|f(a) - h(a)\| \leq \|f(a) - g(a)\| + \|g(a) - h(a)\| < \epsilon$$

implying $h \in B_{S,\epsilon}^f$. Therefore,

$$g \in B_{S,\epsilon}^f \implies B_{S,\hat{\epsilon}}^g \subseteq B_{S,\epsilon}^f. \quad (3.15)$$

Assume g is a map belonging to two different open sets $g \in B_{S,\epsilon}^f$ and $g \in B_{S',\epsilon'}^{f'}$ and define the corresponding $\hat{\epsilon}$ and $\hat{\epsilon}'$ as above. It is straightforward to check that

$$B_{S \cup S', \min(\hat{\epsilon}, \hat{\epsilon}')}^g \subseteq B_{S,\hat{\epsilon}}^g \cap B_{S',\hat{\epsilon}'}^g$$

and, thanks to (3.15), we may conclude that:

$$g \in B_{S,\epsilon}^f \cap B_{S',\epsilon'}^{f'} \implies B_{S \cup S', \min(\hat{\epsilon}, \hat{\epsilon}')}^g \subseteq B_{S,\epsilon}^f \cap B_{S',\epsilon'}^{f'}$$

proving the base is valid (see Definition 2.2.2).

Proposition 3.2.23. *The strong operator topology τ_{SOT} on every hom-set $\mathbf{Hilb}(A, B)$ is Hausdorff.*

Proof. Let $f, g \in \mathbf{Hilb}(A, B)$ so that $f \neq g$ and find an element $a \in A$ such that $f(a) \neq g(a)$; fix $\epsilon = \frac{1}{2}\|f(a) - g(a)\|$. Suppose there is a map h both in $B_{\{a\},\epsilon}^f$ and $B_{\{a\},\epsilon}^g$ then:

$$\|f(a) - g(a)\| \leq \|f(a) - h(a)\| + \|h(a) - g(a)\| < 2\epsilon$$

and we reach a contradiction $\|f(a) - g(a)\| < \|f(a) - g(a)\|$. Therefore, $B_{\{a\},\epsilon}^f$ and $B_{\{a\},\epsilon}^g$ must be disjoint, implying that the topology is Hausdorff. $\square$

Remark 3.2.24. In the literature, the strong operator topology is often defined in terms of a subbase: a collection C of sets that induces a base by taking all finite intersections of elements in C . Such a subbase comprises all open sets $B_{S,\epsilon}^f$ where S is a singleton set; let τ'_{SOT} denote the topology generated by this subbase. Every open set in τ'_{SOT} is trivially an open set in τ_{SOT} , since every element in the subbase for τ'_{SOT} is in τ_{SOT} . Moreover, $B_{S,\epsilon}^f = \bigcap_{a \in S} B_{\{a\},\epsilon}^f$ so the converse also holds, implying that $\tau'_{\text{SOT}} = \tau_{\text{SOT}}$.

Proposition 3.2.25. *In hom-sets $\mathbf{FdHilb}(A, B)$, the strong operator topology τ_{SOT} is equivalent to the operator norm⁹ topology τ_{op} whose base is comprised of the following open sets:*

$$B_\epsilon^f = \{g \in \mathbf{FdHilb}(A, B) \mid \|f - g\|_{\text{op}} < \epsilon\} \quad (3.16)$$

for every $f \in \mathbf{FdHilb}(A, B)$ and $\epsilon > 0$.

Proof. We must show that $\tau_{\text{SOT}} \subseteq \tau_{\text{op}}$ and $\tau_{\text{op}} \subseteq \tau_{\text{SOT}}$. Thanks to Proposition 2.2.3, to show that $\tau \subseteq \tau'$ it is sufficient to prove, for their respective bases $\mathcal{B}$ and $\mathcal{B}'$, that for every $x \in X$ and $B \in \mathcal{B}$:

$$x \in B \implies \exists B' \in \mathcal{B}' \text{ s.t. } B' \subseteq B \text{ and } x \in B'.$$

First, we show that $\tau_{\text{SOT}} \subseteq \tau_{\text{op}}$. For each element $B_{S,\epsilon}^f$ of the base for τ_{SOT} and each $g \in B_{S,\epsilon}^f$ define $\epsilon' = \epsilon - \|f - g\|_{\text{op}}$. Consider the element $B_{\epsilon'}^g$ of the base for τ_{op} so that if $h \in B_{\epsilon'}^g$ then $\|g - h\|_{\text{op}} < \epsilon - \|f - g\|_{\text{op}}$ and, consequently,

$$\|f - h\|_{\text{op}} \leq \|g - h\|_{\text{op}} + \|f - g\|_{\text{op}} < \epsilon$$

implying that $h \in B_{S,\epsilon}^f$ due to the definition of the operator norm. Consequently, $B_{\epsilon'}^g \subseteq B_{S,\epsilon}^f$ and, trivially, $g \in B_{\epsilon'}^g$; thus, $\tau_{\text{SOT}} \subseteq \tau_{\text{op}}$ according to Proposition 2.2.3.

To show that $\tau_{\text{op}} \subseteq \tau_{\text{SOT}}$, let S be an orthonormal basis of A . Thanks to A being finite-dimensional, we have that every $f \in \mathbf{FdHilb}(A, B)$ satisfies:

$$\|f\|_{\text{op}} \leq \sqrt{|S|} \cdot \max_{e \in S} \|f(e)\| \quad (3.17)$$

For every open set B_ϵ^f from the base for τ_{op} and every $g \in B_\epsilon^f$ the base element $B_{\epsilon'}^g$ where $\epsilon' = \epsilon - \|f - g\|_{\text{op}}$ satisfies that $B_{\epsilon'}^g \subseteq B_\epsilon^f$ and $g \in B_{\epsilon'}^g$. Let $\hat{\epsilon} = \frac{\epsilon'}{\sqrt{|S|}}$ and consider the open set $B_{S,\hat{\epsilon}}^g$ from the base for τ_{SOT} so that if $h \in B_{S,\hat{\epsilon}}^g$ then:

$$\forall e \in S, \|g(e) - h(e)\| < \frac{\epsilon'}{\sqrt{|S|}}$$

which implies $\|g - h\|_{\text{op}} < \epsilon'$ due to (3.17). Consequently, $B_{S,\hat{\epsilon}}^g \subseteq B_{\epsilon'}^g \subseteq B_\epsilon^f$ and, trivially, $g \in B_{S,\hat{\epsilon}}^g$; thus, $\tau_{\text{op}} \subseteq \tau_{\text{SOT}}$ according to Proposition 2.2.3, completing the proof. $\square$

⁹Recall that the operator norm $\|f\|_{\text{op}}$ of a bounded linear map $f \in \mathbf{Hilb}(A, B)$ is the infimum of the subset of real numbers $c \in \mathbb{R}$ that satisfy:

$$\forall v \in A, \|f(v)\| \leq c \cdot \|v\|.$$

The following proposition establishes that $\mathbf{Hilb}$ is a hom-convergence UDC when the strong operator topology is used. A similar proof can be used to show that $\mathbf{Hilb}$ with the operator norm topology is a hom-convergence UDC. The reason for using the strong operator topology is that it will later enable us to prove results (such as Lemma 3.3.10) that do not hold in $\mathbf{Hilb}$ if the operator norm topology is used.

Proposition 3.2.26. *$\mathbf{Hilb}$ is a hom-convergence UDC, with the topology on each hom-set $\mathbf{Hilb}(A, B)$ given by the strong operator topology.*

Proof. Pointwise addition of bounded linear maps makes $\mathbf{Hilb}(A, B)$ an abelian group; we now prove that the strong operator topology makes it a Hausdorff abelian group. Recall from Proposition 2.2.10 that $+: \mathbf{Hilb}(A, B) \times \mathbf{Hilb}(A, B) \rightarrow \mathbf{Hilb}(A, B)$ is continuous iff for every pair $f, g \in \mathbf{Hilb}(A, B)$ and every open set B from the base for τ_{SOT} the following implication holds:

$$f + g \in B \implies \exists \text{ open sets } U \ni f \text{ and } V \ni g \text{ s.t. } + (U, V) \subseteq B.$$

Thanks to (3.15) it is sufficient to prove this for open sets B of the form $B_{S, \epsilon}^{f+g}$ with arbitrary S and ϵ since any other $B_{S', \epsilon'}^h$ containing $f + g$ will contain some open set of the form $B_{S', \epsilon'}^{f+g}$. For every pair $f, g \in \mathbf{Hilb}(A, B)$ and every S and ϵ we have that

$$(h, h') \in B_{S, \epsilon/2}^f \times B_{S, \epsilon/2}^g \implies h + h' \in B_{S, \epsilon}^{f+g}. \quad (3.18)$$

due to the following, which holds for all $a \in S$:

$$\|(f + g)(a) - (h + h')(a)\| \leq \|f(a) - h(a)\| + \|g(a) - h'(a)\| < \frac{\epsilon}{2} + \frac{\epsilon}{2}.$$

Therefore, $+(B_{S, \epsilon/2}^f, B_{S, \epsilon/2}^g) \subseteq B_{S, \epsilon}^{f+g}$, implying addition is continuous. Moreover, every open set $B_{S, \epsilon}^f$ satisfies $-(B_{S, \epsilon}^f) = B_{S, \epsilon}^f$, so the inversion map is continuous according to Proposition 2.2.10. Therefore, each hom-set $\mathbf{Hilb}(A, B)$ endowed with the strong operator topology is a Hausdorff abelian group.

Recall that any Hausdorff abelian group can be made into a Σ -group via the functor $G: \mathbf{HausAb} \rightarrow \Sigma_g$ which does not alter the underlying set; thus, every $\mathbf{Hilb}(A, B)$ is a Σ -group. To conclude that the Σ_g -enrichment on $\mathbf{Hilb}$ is well-defined, we need to verify that composition is Σ -bilinear. According to Lemma 2.2.39, it is sufficient to prove that $g \circ -$ and $- \circ f$ are continuous monoid homomorphisms for every choice of bounded linear maps f and g . It is immediate that composition is a group homomorphism on both variables due to addition being defined pointwise; below we show that composition is also continuous in each variable.

- For every $f \in \mathbf{Hilb}(A, B)$ and $g \in \mathbf{Hilb}(B, C)$ with $g \neq 0$ it follows from g being a bounded linear map that

$$h \in B_{S, \epsilon / \|g\|_{\text{op}}}^f \implies g \circ h \in B_{S, \epsilon}^{g \circ f}$$

due to

$$\|g(f(a)) - g(h(a))\| \leq \|g\|_{\text{op}} \cdot \|f(a) - h(a)\| < \|g\|_{\text{op}} \cdot \frac{\epsilon}{\|g\|_{\text{op}}}$$

for all $a \in S$ and, hence,

$$(g \circ -) \left(B_{S, \epsilon / \|g\|_{\text{op}}}^f \right) \subseteq B_{S, \epsilon}^{g \circ f};$$

whereas if $g = 0$ then $(0 \circ -)(B_{S, \epsilon}^f) = \{0\}$ so, trivially, $(g \circ -)(B_{S, \epsilon}^f) \subseteq B_{S, \epsilon}^{g \circ f}$ as well. Consequently, since every open set from the base for τ_{SOT} containing $g \circ f$ will contain some open set of the form $B_{S, \epsilon}^{g \circ f}$ — see the discussion of (3.15) — it follows from Proposition 2.2.10 that $g \circ -$ is continuous.

- Alternatively,

$$h \in B_{f(S), \epsilon}^g \implies h \circ f \in B_{S, \epsilon}^{g \circ f}$$

follows trivially, so $- \circ f$ is continuous as well.

Finally, it is immediate that $(\mathbf{Hilb}, \oplus, \{0\})$ is a Σ_g -UDC since its monoidal product is a biproduct. Therefore, $\mathbf{Hilb}$ — or more precisely $(\mathbf{Hilb}, \Phi)$, where Φ is a collection of identity Σ -homomorphisms — is a hom-convergence UDC. $\square$

Corollary 3.2.27. *FdHilb is a hom-convergence UDC, with the topology on each hom-set $\mathbf{Hilb}(A, B)$ given by the operator norm topology.*

Proof. The claim follows from the same argument provided in the previous proof, thanks to the equivalence in the finite-dimensional case between the strong operator topology and the operator norm topology, as established in Proposition 3.2.25. $\square$

Notice that the definition of hom-convergence UDC may be reproduced for Σ_{ft} -enriched UDCs. However, the motivation behind hom-convergence UDCs is that we may consider the limit of the recursive equation (3.13) that arises in categories with a kernel-image trace, and such a trace assumes the existence of additive inverses in its definition. The following lemma illustrates how the execution formula arises from the kernel-image trace in a hom-convergence UDC.

Lemma 3.2.28. *Let $\mathbf{C}$ be a hom-convergence UDC and let $f: A \oplus U \rightarrow B \oplus U$ be a morphism in $\mathbf{C}$ satisfying $(k, i) \Vdash \text{Tr}_{\text{ki}}^U(f)$ for some morphisms $k \in \mathbf{C}(U, B)$ and $i \in \mathbf{C}(A, U)$. If f satisfies either*

$$\lim_{n \rightarrow \infty} f_{UU}^n f_{UA} \simeq 0 \quad \text{or} \quad \lim_{n \rightarrow \infty} f_{BU} f_{UU}^n \simeq 0$$

then the sequence of partial sums $s: \mathbb{N} \rightarrow \mathbf{C}(A, B)$

$$s(n) = \sum_{j=0}^n f_{BU} f_{UU}^j f_{UA}$$

is convergent, with

$$\lim_{n \rightarrow \infty} s(n) \simeq \text{Tr}_{\text{ki}}^U(f) - f_{BA}.$$

Proof. Recall from the discussion of equation (3.13) that for any $n \in \mathbb{N}$ we have:

$$\sum_{j=0}^n f_{UU}^j f_{UA} = i - f_{UU}^{n+1} \circ i.$$

In particular, the sequence $\alpha: \mathbb{N} \rightarrow \mathbf{C}(A, B)$ given by

$$\alpha(n) = s(n) + f_{BU} f_{UU}^{n+1} \circ i$$

is the constant function $n \mapsto f_{BU} \circ i$ and, hence,

$$\lim_{n \rightarrow \infty} \alpha(n) \simeq f_{BU} \circ i = \text{Tr}_{\text{ki}}^U(f) - f_{BA}.$$

Take the claim's assumption that $\lim_{n \rightarrow \infty} f_{BU} f_{UU}^n \simeq 0$; then, because composition in $\mathbf{C}$ is continuous in each variable,

$$\lim_{n \rightarrow \infty} f_{BU} f_{UU}^n \circ i \simeq 0$$

and we may use that $+$ and the inversion map are continuous to obtain:

$$\lim_{n \rightarrow \infty} (\alpha(n) - f_{BU} f_{UU}^{n+1} \circ i) \simeq \lim_{n \rightarrow \infty} \alpha(n) - \lim_{n \rightarrow \infty} f_{BU} f_{UU}^{n+1} \circ i = (\text{Tr}_{\text{ki}}^U(f) - f_{BA}) - 0$$

which simplifies to:

$$\lim_{n \rightarrow \infty} s(n) \simeq \text{Tr}_{\text{ki}}^U(f) - f_{BA}.$$

If we assume $\lim_{n \rightarrow \infty} f_{UU}^n f_{UA} \simeq 0$ instead, we may use a similar argument along with the recursive expression of k to obtain the same result, completing the proof. $\square$

Notice that the lemma above only implies the existence of the limit of a *sequence*, whereas the Σ function of a hom-convergence UDC is defined (according to the functor $G: \mathbf{HausAb} \rightarrow \Sigma_g$) in terms of the limit of a *net*; this is meant to prevent different orderings of the summands to yield different results. Since the execution formula is defined in terms of the Σ function, we need to upgrade the previous lemma from sequences to nets. To do so, we need to impose an additional condition.

Definition 3.2.29. Let $\mathbf{C}$ be a hom-convergence UDC and let $f: A \oplus U \rightarrow B \oplus U$ be a morphism in $\mathbf{C}$. Let $(\mathcal{F}(\mathbb{N}), \subseteq)$ be the directed set of finite subsets of $\mathbb{N}$. For any $J \in \mathcal{F}(\mathbb{N})$, let

$$\tilde{J} = \{n \in \mathbb{N} \mid n < \max(J) \text{ and } n \notin J\}.$$

If f satisfies both

$$\lim_{J \in \mathcal{F}(\mathbb{N})} \Sigma\{f_{BU} f_{UU}^j f_{UA}\}_{\tilde{J}} \simeq 0 \quad (3.19)$$

and either

$$\lim_{n \rightarrow \infty} f_{UU}^n f_{UA} \simeq 0 \quad \text{or} \quad \lim_{n \rightarrow \infty} f_{BU} f_{UU}^n \simeq 0$$

we say f is *U-tail vanishing*.

For any finite set $J \in \mathcal{F}(\mathbb{N})$, its corresponding $\tilde{J}$ as defined above can be seen as the set of ‘gaps’ that are left by J when enumerating from 0 to $\max(J)$. If $J \subseteq J'$ then $\min(\tilde{J}) \leq \min(\tilde{J}')$ (or $\tilde{J}' \neq \emptyset$) so condition (3.19) may be phrased as follows: *as the first occurrence of a gap tends towards infinity, the sum of the corresponding missing terms approaches zero*.

Lemma 3.2.30. Let $\mathbf{C}$ be a hom-convergence UDC and let $f: A \oplus U \rightarrow B \oplus U$ be a U-tail vanishing morphism in $\mathbf{C}$ satisfying $(k, i) \Vdash \text{Tr}^U(f)$ for some morphisms $k \in \mathbf{C}(U, B)$ and $i \in \mathbf{C}(A, U)$. Then, the execution formula on f :

$$\text{ex}^U(f) = \Sigma(\{f_{BA}\} \uplus \{f_{BU} f_{UU}^n f_{UA}\}_{\mathbb{N}})$$

is well-defined and

$$\text{ex}^U(f) = \text{Tr}_{ki}^U(f).$$

Proof. The previous lemma established that the sequence

$$s(n) = \sum_{j=0}^n f_{BU} f_{UU}^j f_{UA}$$

is convergent under the claim's assumptions, with $\lim s \simeq \text{Tr}_{\text{ki}}^U(f) - f_{BA}$. It is evident that $s \circ \max: \mathcal{F}(\mathbb{N}) \rightarrow \mathbf{C}(A, B)$ is a subnet of s , so

$$\lim_{J \in \mathcal{F}(\mathbb{N})} s(\max(J)) \simeq \text{Tr}_{\text{ki}}^U(f) - f_{BA}.$$

Let $\sigma: \mathcal{F}(\mathbb{N}) \rightarrow \mathbf{C}(A, B)$ be the net of partial sums of the family

$$\{f_{BU}f_{UU}^nf_{UA}\}_{\mathbb{N}}.$$

For any $J \in \mathcal{F}(\mathbb{N})$ we may write:

$$\sigma(J) = s(\max(J)) - \{f_{BU}f_{UU}^j f_{UA}\}_{\bar{j}}$$

and, using that f satisfies (3.19) along with the fact that $+$ and the inversion map are continuous, we obtain

$$\lim_{J \in \mathcal{F}(\mathbb{N})} \sigma(J) \simeq \lim_{J \in \mathcal{F}(\mathbb{N})} s(\max(J)) - \lim_{J \in \mathcal{F}(\mathbb{N})} \{f_{BU}f_{UU}^j f_{UA}\}_{\bar{j}} \simeq (\text{Tr}_{\text{ki}}^U(f) - f_{BA}) - 0$$

implying that $\lim \sigma \simeq \text{Tr}_{\text{ki}}^U(f) - f_{BA}$. Let $\hat{\sigma}$ be the net of partial sums of the family

$$\{f_{BA}\} \uplus \{f_{BU}f_{UU}^nf_{UA}\}_{\mathbb{N}}$$

and recall that, according to the definition of hom-convergence UDC and, in particular, that of the functor $G: \mathbf{HausAb} \rightarrow \Sigma_{\mathbf{g}}$,

$$\Sigma(\{f_{BA}\} \uplus \{f_{BU}f_{UU}^nf_{UA}\}_{\mathbb{N}}) \simeq \lim \hat{\sigma}$$

and, hence, $\text{ex}^U(f) \simeq \lim \hat{\sigma}$. Moreover, it is immediate that $\lim \hat{\sigma} \simeq f_{BA} + \lim \sigma$ so that $\text{ex}^U(f) \simeq \text{Tr}_{\text{ki}}^U(f)$, as claimed. $\square$

The conditions imposed in the definition of tail vanishing morphisms are quite strong but, as shown in the next section, they are satisfied by every morphism in **FdContraction**. This will be sufficient to prove that the execution formula makes **FdContraction** a traced monoidal category. Furthermore, it will be shown that the category $\mathbf{LSI}_{\leq}$ of linear shift invariant quantum processes over discrete time (see Section 3.4) is also traced using the corresponding execution formula.

3.3 Quantum iterative loops and the execution formula

The focus of this section is on categories of Hilbert spaces and contractions. Contractions have some useful properties with regards to convergence. The section begins by introducing some well-known results that will be essential in the following discussions.

3.3.1 Properties of contractions in Hilb (*Preamble*)

The following results are well-known in the literature of contractions and they will be essential in later discussions within this section. The results in this subsection hold for arbitrary Hilbert spaces, thus including infinite-dimensional ones.

Every contraction can be seen as a component of an isometry acting on larger Hilbert spaces; such an isometry is commonly known as a *dilation*. Other dilations of contractions are known (perhaps the most widely known being Sz.-Nagy's dilation theorem: Theorem 4.1 from [50]); the one established in the proposition below is arguably the simplest dilation and it will be sufficient for our purposes.

Proposition 3.3.1 (Halmos [30]). *Let $f \in \mathbf{Contraction}(A, B)$ and define $D_f = \sqrt{\text{id} - f^\dagger f}$. Then, the linear map*

$$g = \begin{pmatrix} -f^\dagger & D_f \\ D_{f^\dagger} & f \end{pmatrix}$$

is an isometry.

Proof. In Section I.3 from [50] it is shown that $D_f^\dagger = D_f$ and $f \circ D_f = D_{f^\dagger} \circ f$. Then,

$$\begin{aligned} g^\dagger g &= \begin{pmatrix} -f & D_{f^\dagger} \\ D_f & f^\dagger \end{pmatrix} \begin{pmatrix} -f^\dagger & D_f \\ D_{f^\dagger} & f \end{pmatrix} \\ &= \begin{pmatrix} ff^\dagger + D_{f^\dagger}^2 & D_{f^\dagger}f - fD_f \\ f^\dagger D_{f^\dagger} - D_f f^\dagger & D_f^2 + f^\dagger f \end{pmatrix} \\ &= \begin{pmatrix} ff^\dagger + \text{id} - ff^\dagger & fD_f - fD_f \\ f^\dagger D_{f^\dagger} - f^\dagger D_{f^\dagger} & \text{id} - f^\dagger f + f^\dagger f \end{pmatrix} \\ &= \begin{pmatrix} \text{id} & 0 \\ 0 & \text{id} \end{pmatrix} \end{aligned}$$

and, hence, $g^\dagger g = \text{id}$, implying that g is an isometry. □

Contractions satisfy a useful property in terms of the sum of the norm of its components, as established below.

Proposition 3.3.2. *For any morphism $f \in \mathbf{Contraction}(A \oplus A', B \oplus B')$ and any $a \in A$,*

$$\|f_{BA}(a)\|^2 + \|f_{B'A}(a)\|^2 \leq \|a\|^2$$

Proof. Suppose there is a vector $a \in A$ such that

$$\|f_{BA}(a)\|^2 + \|f_{B'A}(a)\|^2 > \|a\|^2.$$

Then, the vector $\iota_A(a) = \begin{pmatrix} a \\ 0 \end{pmatrix}$ in $A \oplus A'$ satisfies:

$$\|f(\iota_A(a))\|^2 = \left\| \begin{pmatrix} f_{BA}(a) \\ f_{B'A}(a) \end{pmatrix} \right\|^2 = \|f_{BA}(a)\|^2 + \|f_{B'A}(a)\|^2 > \|a\|^2 = \|\iota_A(a)\|^2.$$

But f is a contraction and we have reached a contradiction $\|f(\iota_A(a))\| > \|\iota_A(a)\|$. Therefore, it must be that $\|f_{BA}(a)\|^2 + \|f_{B'A}(a)\|^2 \leq \|a\|^2$ for all $a \in A$. $\square$

Corollary 3.3.3. *For any $f \in \mathbf{Contraction}(A \oplus A', B \oplus B')$,*

$$\forall a \in A, \|f_{BA}(a)\| = \|a\| \implies f_{B'A} = 0$$

Proof. The previous proposition establishes that for all $a \in A$:

$$\|f_{B'A}(a)\|^2 \leq \|a\|^2 - \|f_{BA}(a)\|^2.$$

Then, if $\|f_{BA}(a)\| = \|a\|$ we have that $\|f_{B'A}(a)\| \leq 0$ for all $a \in A$ and therefore $f_{B'A}$ is the zero map $a \mapsto 0$. $\square$

Contractions have a canonical decomposition into a direct sum of a unitary component and a nonunitary component, as established in Theorem 3.3.7. The following definitions and proposition build towards this result.

Definition 3.3.4. Let $f \in \mathbf{Hilb}(H, H)$ and let $H_0 \subseteq H$ be a closed subspace; let H_1 be the orthogonal complement of H_0 so that $H = H_0 \oplus H_1$. If for every $v \in H_0$ and every $v' \in H_1$ it is satisfied that $f(v) \in H_0$ and $f(v') \in H_1$, we say that H_0 *reduces* f . In such a case, f may be decomposed as follows:

$$f = f_0 \oplus f_1 = \begin{pmatrix} f_0 & 0 \\ 0 & f_1 \end{pmatrix}$$

where $f_0: H_0 \rightarrow H_0$ and $f_1: H_1 \rightarrow H_1$.

Equivalently, reducing subspaces may be defined as subspaces that are invariant both on f and on its adjoint $f^\dagger$.

Proposition 3.3.5. *Let $f \in \mathbf{Hilb}(H, H)$ and let $H_0 \subseteq H$ be a closed subspace; H_0 reduces f iff for all $v \in H_0$ both $f(v) \in H_0$ and $f^\dagger(v) \in H_0$.*

Proof. Let H_1 be the orthogonal complement of H_0 and decompose f as a matrix:

$$f = \begin{pmatrix} f_{00}: H_0 \rightarrow H_0 & f_{01}: H_1 \rightarrow H_0 \\ f_{10}: H_0 \rightarrow H_1 & f_{11}: H_1 \rightarrow H_1 \end{pmatrix}.$$

Assume $f(v) \in H_0$ for all $v \in H_0$, it follows that $\pi_1 f(H_0) = \{0\}$ so that $f_{10} = 0$; similarly, if $f^\dagger(v) \in H_0$ for all $v \in H_0$ then $f_{01}^\dagger = 0$ and therefore $f_{01} = 0$. Thus, $f = f_{00} \oplus f_{11}$ and for all $v' \in H_1$ it is satisfied that $f(v') \in H_1$, implying H_0 reduces f . Conversely, if H_0 reduces f it is immediate that $f^\dagger(v) = f_{00}^\dagger(v) \in H_0$ for all $v \in H_0$. $\square$

Definition 3.3.6 (Section I.3 from [50]). A contraction $f \in \mathbf{Contraction}(H, H)$ is *completely nonunitary* if there is no reducing subspace $H' \subseteq H$ such that the component $f': H' \rightarrow H'$ is unitary.

Theorem 3.3.7 (Theorem 3.2 from [50]). *Every contraction $f \in \mathbf{Contraction}(H, H)$ has a reducing subspace H_0 inducing a decomposition $H = H_0 \oplus H_1$ such that the component $f_0: H_0 \rightarrow H_0$ is unitary and the component $f_1: H_1 \rightarrow H_1$ is completely nonunitary. This decomposition is uniquely determined, with*

$$H_0 = \{v \in H \mid \forall n \in \mathbb{N}, \|f^n(v)\| = \|v\| = \|f^{\dagger n}(v)\|\}.$$

Proof. The proof is paraphrased from [50] (Theorem 3.2) and included here for completeness. For any contraction $f \in \mathbf{Contraction}(H, H)$ let $D_f: H \rightarrow H$ be the linear map $D_f = \sqrt{\text{id} - f^\dagger f}$ and let $K_f = \ker(D_f)$, which is evidently a closed subspace of H and thus $K_f \in \mathbf{Hilb}$. Notice that D_f is self-adjoint and for all $v \in H$:

$$\|D_f(v)\|^2 = \langle v | D_f^2(v) \rangle = \langle v | (\text{id} - f^\dagger f)(v) \rangle = \langle v | v \rangle - \langle f(v) | f(v) \rangle = \|v\|^2 - \|f(v)\|^2$$

and, hence, $v \in K_f$ if and only if $\|f(v)\| = \|v\|$ so K_f may be equivalently defined as the Hilbert space:

$$K_f = \{v \in H \mid \|f(v)\| = \|v\|\}. \quad (3.20)$$

Moreover, if $v \in K_f$ then $D_f^2(v) = (\text{id} - f^\dagger f)(v) = 0$ and we obtain the implication:

$$\|f(v)\| = \|v\| \implies f^\dagger f(v) = v. \quad (3.21)$$

Let H_0 be the following subspace:

$$H_0 = \bigcap_{n=0}^{\infty} (K_{f^n} \cap K_{f^{\dagger n}})$$

which is a Hilbert space since any intersection of closed subspaces is closed. If we expand the right hand side using (3.20), we obtain the definition of H_0 given in the claim. For any $v \in H_0$ and all $n \in \mathbb{N}$:

$$\begin{aligned}\|f^n f(v)\| &= \|f^{n+1}(v)\| = \|v\| = \|f(v)\| \\ \|f^{\dagger n} f(v)\| &= \|f^{\dagger n-1} f^{\dagger} f(v)\| = \|f^{\dagger n-1}(v)\| = \|v\| = \|f(v)\|\end{aligned}$$

where the third equality of the second equation uses (3.21). Therefore, $f(v) \in H_0$ and a similar argument shows that $f^{\dagger}(v) \in H_0$, implying that H_0 reduces f .

Let H_1 be the orthogonal complement of H_0 and let $f = f_0 \oplus f_1$ according to the decomposition $H = H_0 \oplus H_1$. For any $v \in H_0$, we have that $v \in K_f$ so implication (3.21) establishes that $f_0^{\dagger} f_0 = \text{id}_{H_0}$; conversely, we have that $v \in K_{f^{\dagger}}$ so that $f_0 f_0^{\dagger} = \text{id}_{H_0}$, implying f_0 is unitary. On the other hand, let $H_2 \subseteq H_1$ be a closed subspace that reduces f and assume the corresponding $f_2: H_2 \rightarrow H_2$ is unitary. Then, for every $v \in H_2$ and $n \in \mathbb{N}$ we would have $\|f^n(v)\| = \|v\| = \|f^{\dagger n}(v)\|$ and, hence, $v \in H_0$; but $H_2 \subseteq H_1$ and H_1 is orthogonal to H_0 , so we reach a contradiction. Therefore, we conclude that f_2 cannot be unitary and, hence, f_1 is completely nonunitary, as claimed. $\square$

Finally, the following result will be essential to proving that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a totally traced category.

Proposition 3.3.8 (Bartha [10]). *Let $f \in \mathbf{Isometry}(A \oplus U, B \oplus U)$ where $\text{id} - f_{UU}$ is invertible, then*

$$h = f_{BA} + f_{BU}(\text{id} - f_{UU})^{-1} f_{UA}$$

is an isometry.

Proof. The proof is paraphrased from [10] and included here for completeness. For the sake of brevity, use the following shorthand for the components of f :

$$f = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} f_{BA} & f_{BU} \\ f_{UA} & f_{UU} \end{pmatrix}$$

Considering that f is an isometry, we have that $f^{\dagger} f = \text{id}$; this gives us the following equalities:

$$a^{\dagger} a + c^{\dagger} c = \text{id}_A \tag{3.22}$$

$$a^{\dagger} b + c^{\dagger} d = 0_{AU} \tag{3.23}$$

$$b^{\dagger} a + d^{\dagger} c = 0_{UA} \tag{3.24}$$

$$b^{\dagger} b + d^{\dagger} d = \text{id}_U. \tag{3.25}$$

To prove that h is an isometry we show that $h^\dagger h = \text{id}$, where

$$h^\dagger h = (a + b(\text{id} - d)^{-1}c)^\dagger (a + b(\text{id} - d)^{-1}c).$$

Using that $(g \circ f)^\dagger = f^\dagger \circ g^\dagger$ and $((\text{id} - g)^{-1})^\dagger = (\text{id} - g^\dagger)^{-1}$, $h^\dagger h$ is equal to:

$$a^\dagger a + a^\dagger b(\text{id} - d)^{-1}c + c^\dagger(\text{id} - d^\dagger)^{-1}b^\dagger a + c^\dagger(\text{id} - d^\dagger)^{-1}b^\dagger b(\text{id} - d)^{-1}c.$$

Using (3.23) and (3.24), this is equal to:

$$a^\dagger a - c^\dagger d(\text{id} - d)^{-1}c - c^\dagger(\text{id} - d^\dagger)^{-1}d^\dagger c + c^\dagger(\text{id} - d^\dagger)^{-1}b^\dagger b(\text{id} - d)^{-1}c$$

and, thus, $h^\dagger h = a^\dagger a + c^\dagger g c$ where

$$g = (\text{id} - d^\dagger)^{-1}b^\dagger b(\text{id} - d)^{-1} - d(\text{id} - d)^{-1} - (\text{id} - d^\dagger)^{-1}d^\dagger.$$

Thanks to (3.22), to prove that $h^\dagger h = \text{id}$ it is sufficient to show that $g = \text{id}$. We show $g = \text{id}$ by applying $(\text{id} - d^\dagger) \circ -$ and $- \circ (\text{id} - d)$ to both sides of the equation, then rearranging the terms to reach (3.25):

$$\begin{aligned} g = \text{id} &\iff b^\dagger b - (\text{id} - d^\dagger)d - d^\dagger(\text{id} - d) = (\text{id} - d^\dagger)(\text{id} - d) \\ &\iff b^\dagger b - d + d^\dagger d - d^\dagger + d^\dagger d = \text{id} - d - d^\dagger + d^\dagger d \\ &\iff b^\dagger b + d^\dagger d = \text{id} \end{aligned}$$

Consequently, $h^\dagger h = \text{id}$ and h is an isometry, as claimed. $\square$

Corollary 3.3.9. *Let $f \in \text{Contraction}(A \oplus U, B \oplus U)$ where $\text{id} - f_{UU}$ is invertible, then*

$$h = f_{BA} + f_{BU}(\text{id} - f_{UU})^{-1}f_{UA}$$

is a contraction.

Proof. Proposition 3.3.1 established that for every contraction f the following linear map:

$$g = \begin{pmatrix} -f^\dagger & D_f \\ D_{f^\dagger} & f \end{pmatrix}$$

is an isometry, where $D_f = \sqrt{\text{id} - f^\dagger f}$. Next, we decompose $g: B \oplus U \oplus A \oplus U \rightarrow A \oplus U \oplus B \oplus A$ as follows:

$$g = \begin{pmatrix} a: B \oplus U \oplus A \rightarrow A \oplus U \oplus B & b: U \rightarrow A \oplus U \oplus B \\ c: B \oplus U \oplus A \rightarrow U & f_{UU} \end{pmatrix}$$

Since g is an isometry and $\text{id} - f_{UU}$ is invertible by assumption, the previous proposition implies that:

$$h' = a + b(\text{id} - f_{UU})^{-1}c$$

is an isometry. It is straightforward to check that $h = \pi_B \circ h' \circ \iota_A$ where the injection ι_A is an isometry and the projection π_B is a contraction. Therefore, h is a contraction. $\square$

3.3.2 The execution formula in FdContraction

The goal of this subsection is to prove that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a (totally) traced monoidal category, where ex is the execution formula. To do so, the notions of hom-convergence UDC and U -tail vanishing morphisms introduced in Section 3.2.4 will be used. In essence, the strategy is to derive the validity of ex as a categorical trace via the kernel-image trace on $\mathbf{FdHilb}$. Once the result that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a totally traced category has been established, it follows that $\mathbf{FdIsometry}$ and $\mathbf{FdUnitary}$ are themselves totally traced; the latter results were previously proven by Bartha [10] using the framework of matrix iteration theories.

Before the formal discussion begins, it is worth presenting a simple example that will illustrate some of the intricacies of these results. Let the linear map $H: \mathbb{C} \oplus \mathbb{C} \rightarrow \mathbb{C} \oplus \mathbb{C}$ be described by the matrix:

$$H = \begin{pmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (3.26)$$

Notice that H is unitary and hence a morphism in $\mathbf{FdContraction}$. We may evaluate the execution formula:

$$\text{ex}^{\mathbb{C}}(H) = \Sigma(\{h_{00}\} \uplus \{h_{01}h_{11}^n h_{10}\}_{\mathbb{N}}) = \frac{1}{\sqrt{2}} + \sum_{n=0}^{\infty} \frac{1}{\sqrt{2}} \left(-\frac{1}{\sqrt{2}}\right)^n \frac{1}{\sqrt{2}}$$

and realise that $\text{ex}^{\mathbb{C}}(H) = 1$, which is a unitary $\mathbb{C} \rightarrow \mathbb{C}$. Similarly, $\text{ex}^{\mathbb{C}}(\sigma H \sigma) = 1$ as well and $\text{ex}^{\mathbb{C}}(\sigma H) = -1 = \text{ex}^{\mathbb{C}}(H \sigma)$, where σ is the symmetric braiding in $(\mathbf{FdContraction}, \oplus, \{0\})$. What is peculiar about this situation is that, if we were to sum all elements from $\text{ex}^{\mathbb{C}}(\sigma H \sigma)$ but h_{11} , the result is larger than 1:

$$\Sigma\{h_{10}h_{00}^n h_{01}\}_{\mathbb{N}} = \sum_{n=0}^{\infty} \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{2}}\right)^n \frac{1}{\sqrt{2}} = \frac{1}{2-\sqrt{2}} > 1$$

and it is not until we add in $h_{11} = -\frac{1}{\sqrt{2}}$ that the result becomes 1. The key insight to be drawn from this example is that, even when the result of the execution formula is a contraction, its partial sums need not be contractions.

The following lemma establishes that every contraction in **Hilb** satisfies one of the conditions in the definition of tail vanishing morphisms (Definition 3.2.29).

Lemma 3.3.10. *Let $f: A \oplus U \rightarrow B \oplus U$ be a contraction in the hom-convergence UDC **Hilb** defined in terms of the strong operator topology (see Proposition 3.2.26). Then,*

$$\lim_{n \rightarrow \infty} f_{BU} f_{UU}^n \simeq 0.$$

Proof. Recall that this limit exists iff for every finite set of vectors $S \subseteq U$ and every $\epsilon > 0$ there is some n such that for all $k \geq n$ we have that $f_{BU} f_{UU}^k \in B_{S, \epsilon}^0$,¹⁰ i.e. we need to show that:

$$k \geq n \implies \forall u \in S, \|f_{BU} f_{UU}^k(u)\| < \epsilon \quad (3.27)$$

To do so, the first step will be to show that the following is satisfied:

$$\lim_{n \in \mathbb{N}} \left(\|f_{UU}^n(u)\|^2 - \|f_{UU}^{n+1}(u)\|^2 \right) \simeq 0. \quad (3.28)$$

For every $u \in U$, the collection of real numbers $S_u = \{\|f_{UU}^n(u)\|^2\}_{n \in \mathbb{N}}$ is bounded from below by 0 and, hence, by completeness of the total order $(\mathbb{R}, \leq)$, the infimum of the set S_u exists. Thus, for each $\epsilon > 0$ there is some $k \in \mathbb{N}$ such that

$$\|f_{UU}^k(u)\|^2 < \inf(S_u) + \epsilon$$

since otherwise $\inf(S_u) + \epsilon$ would be a lower bound to S_u greater than $\inf(S_u)$, contradicting the definition of infimum. Considering that f_{UU} is necessarily a contraction, every $k' \geq k$ will satisfy

$$\|f_{UU}^{k'}(u)\|^2 - \inf(S_u) \leq \|f_{UU}^k(u)\|^2 - \inf(S_u)$$

and, consequently, $\|f_{UU}^{k'}(u)\|^2 - \inf(S_u) < \epsilon$ so that in the standard topology on $\mathbb{R}$ we have that:

$$\lim_{n \in \mathbb{N}} \|f_{UU}^n(u)\|^2 \simeq \inf(S_u).$$

The previous argument is commonly known as the monotone convergence theorem. Moreover, it is trivial to check that $\inf(S_u) = \inf(S_{f_{UU}(u)})$ for every $u \in U$ and due to addition (and negation) of real numbers being continuous:

$$\lim_{n \in \mathbb{N}} \left(\|f_{UU}^n(u)\|^2 - \|f_{UU}^{n+1}(u)\|^2 \right) \simeq \inf(S_u) - \inf(S_{f_{UU}(u)}) = 0$$

¹⁰As established by Proposition 2.2.18 and implication (3.15), it is sufficient to consider only the open sets from the base for τ_{SOT} ‘centered’ at 0, i.e. of the form $B_{S, \epsilon}^0$ for some S and ϵ .

and, hence, equation (3.28) is satisfied.

Proposition 3.3.2 establishes that for all $u \in U$:

$$\|f_{BU}(u)\|^2 \leq \|u\|^2 - \|f_{UU}(u)\|^2$$

and, in particular, for every $n \in \mathbb{N}$, we have that $f_{UU}^n(u) \in U$ and

$$\|f_{BU}f_{UU}^n(u)\|^2 \leq \|f_{UU}^n(u)\|^2 - \|f_{UU}^{n+1}(u)\|^2.$$

Since the limit of the right hand side is known to be zero (3.28) and the sequence of real numbers $\|f_{BU}f_{UU}^n(u)\|^2$ is bounded from below by 0, it follows that:

$$\lim_{n \rightarrow \infty} \|f_{BU}f_{UU}^n(u)\|^2 \simeq 0.$$

Therefore, for each $u \in U$ and each $\epsilon > 0$ we can find an $n_u \in \mathbb{N}$ such that every $k \geq n_u$ satisfies $\|f_{BU}f_{UU}^k(u)\|^2 < \epsilon^2$. For any given finite set of vectors $S \subseteq U$ we can take $n = \max \{n_u \mid u \in S\}$ so that (3.27) is satisfied, completing the proof. $\square$

The previous lemma applies to contractions between arbitrary Hilbert spaces, including the case of infinite dimensions. In contrast, the rest of the results in this subsection have only been established for the case of finite-dimensional spaces and Section 3.3.4 discusses the obstacles in the way of their generalisation.

Lemma 3.3.11. *Every contraction $f: A \oplus U \rightarrow B \oplus U$ in $\mathbf{FdHilb}$ satisfies $(k, i) \Vdash \text{Tr}^U(f)$ for some morphisms $k \in \mathbf{FdHilb}(U, B)$ and $i \in \mathbf{FdHilb}(A, U)$.*

Proof. The proof will follow from Corollary 3.1.10 which establishes that it suffices to show that:

$$\begin{aligned} \text{im}(f_{UA}) &\subseteq \text{im}(\text{id} - f_{UU}) \\ \ker(\text{id} - f_{UU}) &\subseteq \ker(f_{BU}). \end{aligned}$$

Let $v \in \ker(\text{id} - f_{UU})$; this implies $\|f_{UU}(v)\| = \|v\|$ and, necessarily, $f_{BU}(v) = 0$ for f to be a contraction. Consequently, $\ker(\text{id} - f_{UU}) \subseteq \ker(f_{BU})$ is satisfied. To prove the inclusion of images, we convert it to an statement on kernels as follows:

$$\begin{aligned} \text{im}(f_{UA}) \subseteq \text{im}(\text{id} - f_{UU}) &\iff \text{im}(\text{id} - f_{UU})^\perp \subseteq \text{im}(f_{UA})^\perp \quad (\text{Proposition 3.1.13}) \\ &\iff \ker(\text{id} - f_{UU}^\dagger) \subseteq \ker(f_{UA}^\dagger) \quad (\text{Lemma 3.1.12}) \end{aligned}$$

where the first step holds because $\text{im}(\text{id} - f_{UU})$ is a closed subspace — in $\mathbf{FdHilb}$ every subspace is closed. Considering that the adjoint of any contraction f is also a

contraction,¹¹ the inclusion $\ker(\text{id} - f_{UU}^\dagger) \subseteq \ker(f_{UA}^\dagger)$ follows from the same argument used for $\ker(\text{id} - f_{UU}) \subseteq \ker(f_{BU})$, completing the proof. $\square$

The phrasing of the previous lemma purposely refers to the category **FdHilb** instead of **FdContraction** because, in principle, the morphisms k and i need not be contractions themselves. For instance, in the case of the morphism H from (3.26) we have that $(k, i) \Vdash \text{Tr}_{\text{ki}}^{\mathbb{C}}(\sigma H \sigma)$ where $k = \frac{1}{\sqrt{2}-1} = i$, but $\left| \frac{1}{\sqrt{2}-1} \right| > 1$ and, hence, k and i are morphisms in **FdHilb** but not in **FdContraction**.

The following two lemmas build towards Lemma 3.3.14 where it is established that every contraction $f: A \oplus U \rightarrow B \oplus U$ between finite-dimensional Hilbert spaces is U -tail vanishing.

Lemma 3.3.12. *Let $f \in \mathbf{FdContraction}(H, H)$ and let $f = f_0 \oplus f_1$ be its decomposition into a unitary f_0 and a completely nonunitary contraction $f_1: H_1 \rightarrow H_1$ (see Theorem 3.3.7). Then,*

$$\left\| f_1^{\dim H_1} \right\|_{\text{op}} < 1.$$

Proof. If $H_1 = \{0\}$ then $\|f_1\|_{\text{op}} = 0$ and the proof is trivial. Otherwise, recall from (3.20) that for any contraction g we can define a closed subspace

$$K_g = \{v \in H \mid \|g(v)\| = \|v\|\}.$$

Then, for every $k \in \mathbb{N}$ we may define a closed subspace of H_1 as follows:

$$V_k = \bigcap_{n=0}^k K_{f_1^n} = \{v \in H_1 \mid \forall n \leq k, \|f_1^n(v)\| = \|v\|\}.$$

Clearly, $V_0 = H_1$ and $V_{k+1} \subseteq V_k$ for all $k \in \mathbb{N}$. Fix a value of $k \in \mathbb{N}$ such that $V_k \neq \{0\}$ and suppose — for the sake of an argument by contradiction — that $V_k = V_{k+1}$. Then, $v \in V_k$ implies $f_1(v) \in V_k$ since for all $n \leq k$:

$$\|f_1^n(f_1(v))\| = \|f_1^{n+1}(v)\| = \|v\| = \|f_1(v)\|.$$

Moreover, we have that $\|f_1(v)\| = \|v\|$, so the restriction $f_1|_{V_k}: V_k \rightarrow V_k$ is an isometry. In finite dimensions, every isometry is surjective (and, hence, unitary), implying that for each $v \in V_k$ we also have that $f_1^\dagger(v) \in V_k$. Then, V_k reduces f_1 according to Proposition 3.3.5 but $f_1|_{V_k}$ is unitary, contradicting that f_1 is completely nonunitary. Consequently, it must be that for all $k \in \mathbb{N}$:

$$V_k \neq \{0\} \implies V_k \neq V_{k+1}$$

¹¹Recall that every bounded linear map $f \in \mathbf{Hilb}(A, B)$ satisfies that its adjoint $f^\dagger$ has the same operator norm $\|f\|_{\text{op}} = \|f^\dagger\|_{\text{op}}$. Thus, if f is a contraction then $f^\dagger$ is also a contraction.

and, hence, V_{k+1} is a proper subspace of V_k and $\dim V_{k+1} < \dim V_k$. Then, it is immediate that $V_{\dim H_1} = \{0\}$ which, according to the definition of V_k , implies that for all $v \in H_1$ (other than 0):

$$\|f_1^{\dim H_1}(v)\| < \|v\|.$$

Since H_1 is finite-dimensional it follows that $\|f_1^{\dim H_1}\|_{\text{op}} < 1$, as claimed. $\square$

Lemma 3.3.13. *Let $f: A \oplus U \rightarrow B \oplus U$ be a contraction in $\mathbf{FdHilb}$ and decompose $U = U_0 \oplus U_1$ according to the decomposition of f_{UU} into its unitary and completely nonunitary components respectively. Then, there is a real number M such that:*

$$\sum_{k=0}^{\infty} \|f_{UU}^k f_{UA}\|_{\text{op}} \leq M$$

where

$$M = \begin{cases} \dim U_1 \cdot \left(1 - \|f_{U_1 U_1}^{\dim U_1}\|_{\text{op}}\right)^{-1} & \text{if } \dim U_1 \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Proof. According to Theorem 3.3.7, we may decompose f_{UU} so that f may be expressed as:

$$f = \begin{pmatrix} f_{BA} & f_{BU_0} & f_{BU_1} \\ f_{U_0 A} & f_{U_0 U_0} & f_{U_0 U_1} \\ f_{U_1 A} & f_{U_1 U_0} & f_{U_1 U_1} \end{pmatrix}$$

where $f_{U_0 U_0}$ is unitary. Then, Corollary 3.3.3 establishes that $f_{BU_0} = 0$ and $f_{U_1 U_0} = 0$ and, similarly, $f_{U_0 A} = 0$ and $f_{U_0 U_1} = 0$ because $f_{U_0 U_0}^\dagger$ is also an isometry. Consequently, for every $k \in \mathbb{N}$ the following is satisfied:

$$f_{UU}^k f_{UA} = \begin{pmatrix} f_{U_0 U_0}^k & 0 \\ 0 & f_{U_1 U_1}^k \end{pmatrix} \begin{pmatrix} 0 \\ f_{U_1 A} \end{pmatrix} = \begin{pmatrix} 0 \\ f_{U_1 U_1}^k f_{U_1 A} \end{pmatrix}$$

and, hence,

$$\|f_{UU}^k f_{UA}\|_{\text{op}} = \|f_{U_1 U_1}^k f_{U_1 A}\|_{\text{op}}.$$

Therefore, if $U_1 = \{0\}$ then $\|f_{UU}^k f_{UA}\|_{\text{op}} = 0$ for all $k \in \mathbb{N}$ and it is trivial that $\sum_{k=0}^{\infty} \|f_{UU}^k f_{UA}\|_{\text{op}} = 0$, satisfying the claim in the case where $\dim U_1 = 0$. Otherwise, if $U_1 \neq \{0\}$ we may use the well-known inequality $\|h \circ g\|_{\text{op}} \leq \|h\|_{\text{op}} \cdot \|g\|_{\text{op}}$ to obtain:

$$\sum_{k=0}^{\infty} \|f_{UU}^k f_{UA}\|_{\text{op}} \leq \left(\sum_{r=0}^{\dim U_1 - 1} \|f_{U_1 U_1}^r\|_{\text{op}} \right) \left(\sum_{q=0}^{\infty} \|f_{U_1 U_1}^{q \cdot \dim U_1} f_{U_1 A}\|_{\text{op}} \right)$$

and due to $f_{U_1 U_1}$ and $f_{U_1 A}$ being contractions, we conclude that:

$$\sum_{k=0}^{\infty} \|f_{UU}^k f_{UA}\|_{\text{op}} \leq \dim U_1 \cdot \sum_{q=0}^{\infty} \left(\|f_{U_1 U_1}^{\dim U_1}\|_{\text{op}} \right)^q$$

Finally, according to Lemma 3.3.12, we have that $\|f_{U_1 U_1}^{\dim U_1}\|_{\text{op}} < 1$ so it follows from the convergence criteria of the geometric series that:

$$\sum_{k=0}^{\infty} \|f_{UU}^k f_{UA}\|_{\text{op}} \leq \dim U_1 \cdot \left(1 - \|f_{U_1 U_1}^{\dim U_1}\|_{\text{op}} \right)^{-1}$$

thus proving the claim in the case where $\dim U_1 \neq 0$. $\square$

Lemma 3.3.14. *Every contraction $f: A \oplus U \rightarrow B \oplus U$ in $\mathbf{FdHilb}$ is U -tail vanishing.*

Proof. Lemma 3.3.10 already established that $\lim f_{BU} f_{UU}^n \simeq 0$. It remains to check that

$$\lim_{J \in \mathcal{F}(\mathbb{N})} \Sigma\{f_{BU} f_{UU}^j f_{UA}\}_{\tilde{J}} \simeq 0 \quad (3.29)$$

is satisfied, where

$$\tilde{J} = \{n \in \mathbb{N} \mid n < \max(J) \text{ and } n \notin J\}.$$

This requires us to show that for all $\epsilon > 0$, there is a $J \in \mathcal{F}(\mathbb{N})$ such that

$$J \subseteq J' \implies \|\Sigma\{f_{BU} f_{UU}^j f_{UA}\}_{\tilde{J}'}\|_{\text{op}} < \epsilon.$$

If $\tilde{J}' = \emptyset$ it is trivial to prove that $\Sigma\{f_{BU} f_{UU}^j f_{UA}\}_{\tilde{J}'} = 0$. Otherwise, if $\tilde{J}' \neq \emptyset$ let $m_J \in \mathbb{N}$ be the index of the first ‘gap’ in J , explicitly:

$$m_J = \begin{cases} \min \tilde{J} & \text{if } \tilde{J} \neq \emptyset \\ \max J + 1 & \text{otherwise.} \end{cases}$$

Notice that whenever $J \subseteq J'$ we have that $m_J \leq \min \tilde{J}'$ and, hence, $j - m_J \geq 0$ for all $j \in \tilde{J}'$, so that we may write:

$$\|\Sigma\{f_{BU} f_{UU}^j f_{UA}\}_{\tilde{J}'}\|_{\text{op}} \leq \|f_{BU} f_{UU}^{m_J}\|_{\text{op}} \cdot \|\Sigma\{f_{UU}^{j-m_J} f_{UA}\}_{\tilde{J}'}\|_{\text{op}}.$$

Moreover,

$$\|\Sigma\{f_{UU}^{j-m_J} f_{UA}\}_{\tilde{J}'}\|_{\text{op}} \leq \sum_{k=0}^{\infty} \|f_{UU}^k f_{UA}\|_{\text{op}}$$

due to the triangle inequality of norms along with norms being positive numbers. Recall from Lemma 3.3.13 that there is some $M \in \mathbb{R}$ such that:

$$\sum_{k=0}^{\infty} \|f_{UU}^k f_{UA}\|_{\text{op}} \leq M.$$

We conclude that

$$J \subseteq J' \implies \|\Sigma\{f_{BU} f_{UU}^j f_{UA}\}_{J'}\|_{\text{op}} \leq \|f_{BU} f_{UU}^{m_J}\|_{\text{op}} \cdot M.$$

If $M = 0$ then f_{UU} is unitary (see the proof of Lemma 3.3.13 for further details) and hence $f_{UA} = 0$ according to Corollary 3.3.3 so that (3.29) follows trivially. Otherwise, if $M \neq 0$ we may use that $\lim f_{BU} f_{UU}^n \simeq 0$ (established in Lemma 3.3.10) which implies that for every $\epsilon > 0$ we may choose some $\widehat{m} \in \mathbb{N}$ such that for all $m' \in \mathbb{N}$:

$$\widehat{m} \leq m' \implies \|f_{BU} f_{UU}^{m'}\|_{\text{op}} < \frac{\epsilon}{M}.$$

Then, we simply need to choose some $J \in \mathcal{F}(\mathbb{N})$ such that $m_J = \widehat{m}$ and it will follow that for all $J' \in \mathcal{F}(\mathbb{N})$:

$$J \subseteq J' \implies \|\Sigma\{f_{BU} f_{UU}^j f_{UA}\}_{J'}\|_{\text{op}} < \epsilon$$

thus, verifying that (3.29) is satisfied and proving that f is U -tail vanishing. $\square$

Finally, we reach the main result of this section.

Theorem 3.3.15. *The category $(\mathbf{FdContraction}, \oplus, \text{ex})$ is totally traced, where ex is defined on any morphism $f: A \oplus U \rightarrow B \oplus U$ as follows:*

$$\text{ex}^U(f) = \Sigma \left(\{f_{BA}\} \uplus \{f_{BU} f_{UU}^n f_{UA}\}_{\mathbb{N}} \right).$$

Proof. Corollary 3.2.27 established that $\mathbf{FdHilb}$ is a hom-convergence UDC and, for every contraction $f: A \oplus U \rightarrow B \oplus U$ in $\mathbf{FdHilb}$, Lemma 3.3.11 shows that $(k, i) \Vdash \text{Tr}_{\text{ki}}^U(f)$ whereas Lemma 3.3.14 shows that f is U -tail vanishing. According to Lemma 3.2.30 this is sufficient for $\text{ex}^U(f)$ to be defined in $\mathbf{FdHilb}$ for every contraction f and every $U \in \mathbf{FdHilb}$, with $\text{ex}^U(f) = \text{Tr}_{\text{ki}}^U(f)$. Since Tr_{ki} is a partial trace in $\mathbf{FdHilb}$ and the embedding functor $\mathbf{FdContraction} \hookrightarrow \mathbf{FdHilb}$ is faithful and strict monoidal, Proposition 3.1.3 implies that $(\mathbf{FdContraction}, \oplus, \widehat{\text{ex}})$ is a partially traced category, where:

$$\widehat{\text{ex}}^U(f) = \begin{cases} \text{ex}^U(f) & \text{if } \text{ex}^U(f) \text{ is a contraction} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

To show that $(\mathbf{FdContraction}, \oplus, \widehat{\text{ex}})$ is a totally traced category we need to show that for every contraction $f \in \mathbf{FdContraction}(A \oplus U, B \oplus U)$ the result of the execution formula $\text{ex}^U(f)$ is guaranteed to be a contraction. We prove this by an inductive argument on the dimension of U . If $U = \mathbb{C}$ the component $f_{\mathbb{C}\mathbb{C}}: \mathbb{C} \rightarrow \mathbb{C}$ is just a scalar.

- If $|f_{\mathbb{C}\mathbb{C}}| < 1$ then $1 - f_{\mathbb{C}\mathbb{C}}$ is invertible and

$$\Sigma\{f_{\mathbb{C}\mathbb{C}}^n\}_{\mathbb{N}} = (1 - f_{\mathbb{C}\mathbb{C}})^{-1}.$$

Thanks to $\mathbf{FdHilb}$ being $\Sigma_{\mathbf{g}}$ -enriched, composition distributes over Σ so that:

$$\text{ex}^{\mathbb{C}}(f) = f_{BA} + f_{B\mathbb{C}}(1 - f_{\mathbb{C}\mathbb{C}})^{-1}f_{\mathbb{C}A}.$$

Then, according to Corollary 3.3.9, $\text{ex}^{\mathbb{C}}(f)$ is a contraction.

- Otherwise, if $|f_{\mathbb{C}\mathbb{C}}| = 1$ then Corollary 3.3.3 implies that $f_{B\mathbb{C}} = 0$ and, hence, $\text{ex}^U(f) = f_{BA}$, which is necessarily a contraction.

If $U = \mathbb{C}^{\oplus k}$ for some $k > 1$ it follows from repeated applications of the above that $\text{ex}^{\mathbb{C}}(\dots \text{ex}^{\mathbb{C}}(f))$ is a contraction. Since $(\mathbf{FdContraction}, \oplus, \widehat{\text{ex}})$ has been established to be partially traced, we may use vanishing II repeatedly to obtain:

$$\widehat{\text{ex}}^{\mathbb{C}}(\dots \widehat{\text{ex}}^{\mathbb{C}}(f)) \simeq \widehat{\text{ex}}^{\mathbb{C}^{\oplus k}}(f)$$

and, hence, $\text{ex}^{\mathbb{C}^{\oplus k}}(f)$ must be a contraction. For any arbitrary U there is a unitary $\phi: U \rightarrow \mathbb{C}^{\oplus \dim U}$ due to U being finite-dimensional. Clearly, $f' = (\text{id}_B \oplus \phi)f(\text{id}_A \oplus \phi^\dagger)$ is a contraction and it has already been established that $\text{ex}^{\mathbb{C}^{\oplus \dim U}}(f')$ is a contraction; then, using dinaturality we obtain:

$$\widehat{\text{ex}}^{\mathbb{C}^{\oplus \dim U}}(f') \simeq \widehat{\text{ex}}^U(f \circ (\text{id}_A \oplus \phi^\dagger \phi)) = \widehat{\text{ex}}^U(f)$$

implying that $\text{ex}^U(f)$ is a contraction for any arbitrary $f \in \mathbf{FdContraction}(A \oplus U, B \oplus U)$. Consequently, $(\mathbf{FdContraction}, \oplus, \widehat{\text{ex}})$ is a totally traced category, completing the proof. $\square$

The theorem above along with Lemma 3.2.30 establish that both the execution formula and the kernel-image trace coincide in $\mathbf{FdContraction}$. This is in stark contrast to Proposition 3.14 from [44] where it was shown that no partial trace in $(\mathbf{Vect}, \oplus)$ — nor in $(\mathbf{FdHilb}, \oplus)$ — could simultaneously coincide with the execution formula and

the kernel-image trace. This is no contradiction, though, as the proof of their result is based on a linear map presented as a counterexample, but such a linear map is not a contraction and hence their result need not hold for **FdContraction**. Notice that a similar situation occurred in Example 3.2.12 where a linear map in **FdHilb** (but not in **FdContraction**) violated vanishing II. In retrospective, this is not surprising: notice that the characterisation of tail vanishing morphisms (Definition 3.2.29) immediately rules out any linear map whose operator norm is strictly larger than one. Since we heavily relied on the fact that morphisms in **FdContraction** are tail vanishing (Lemma 3.3.14) in our argument leading to the previous theorem (particularly, Lemma 3.2.30), our argument is exploiting a key difference between **FdHilb** and **FdContraction**.

On the other hand, even though the categories **FdIsometry** and **FdUnitary** are not UDCs (because they lack quasi-projections), we may use their faithful strict monoidal embedding into **FdContraction** to induce a total trace in them, calculated via the execution formula.

Corollary 3.3.16. *The category $(\mathbf{FdIsometry}, \oplus, \widehat{\text{ex}})$ is totally traced, where:*

$$\widehat{\text{ex}}^U(f) = \begin{cases} \text{ex}^U(f) & \text{if } \text{ex}^U(f) \text{ is an isometry} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Similarly, the category $(\mathbf{FdUnitary}, \oplus, \widehat{\text{ex}})$ is totally traced, where $\text{ex}^U(f)$ is instead required to be unitary.

Proof. The embedding functor $\mathbf{FdIsometry} \hookrightarrow \mathbf{FdContraction}$ is faithful and strict monoidal and, hence, $(\mathbf{FdIsometry}, \oplus, \widehat{\text{ex}})$ is partially traced according to Proposition 3.1.3. Next, we must prove that for every isometry $f: A \oplus U \rightarrow B \oplus U$ in **FdHilb** $\text{ex}^U(f)$ is an isometry. This follows from a similar argument as the one used in the previous theorem:

- if $U = \mathbb{C}$ then Proposition 3.3.8 can be used to show that $\text{ex}^{\mathbb{C}}(f)$ is an isometry;
- then, repeated application of this implies that $\text{ex}^{\mathbb{C}}(\dots \text{ex}^{\mathbb{C}}(f))$ is an isometry and, by vanishing II, $\text{ex}^{\mathbb{C}^{\oplus k}}(f)$ is an isometry as well;
- for every U there is a unitary $\phi: U \rightarrow \mathbb{C}^{\oplus \dim U}$ due to U being finite-dimensional, so $f' = (\text{id}_B \oplus \phi)f(\text{id}_A \oplus \phi^\dagger)$ is an isometry and we may use dinaturality to conclude that $\text{ex}^U(f)$ is an isometry.

Consequently, $\widehat{\text{ex}}$ is a total function and $(\mathbf{FdIsometry}, \oplus, \widehat{\text{ex}})$ is a totally traced category.

The case of $(\mathbf{FdUnitary}, \oplus, \widehat{\text{ex}})$ can be proven with a similar argument, this time using that, in $\mathbf{FdHilb}$, a morphism $f: A \rightarrow B$ is unitary if and only if $\dim(A) = \dim(B)$ and f is an isometry. $\square$

Thus, it has been shown that the execution formula — which is closely related to iterative loops — is a well-defined trace in some important categories of quantum processes. This was previously established in the case of $\mathbf{FdIsometry}$ and $\mathbf{FdUnitary}$ by Bartha [10] using a different approach; a brief comparison between these two approaches appears in Section 3.5. The proof of some of the results in this section heavily relied on the fact that the objects of $\mathbf{FdContraction}$ are finite-dimensional Hilbert spaces; some open questions regarding the general case of $\mathbf{Contraction}$ are discussed in Section 3.3.4. Furthermore, the following remark discusses the prospects of proving that the execution formula is continuous.

Remark 3.3.17. We know from Proposition 2.2.19 that a function

$$\text{ex}^U: \mathbf{FdUnitary}(A \oplus U, B \oplus U) \rightarrow \mathbf{FdUnitary}(A, B)$$

is continuous if and only if for every net $\alpha: D \rightarrow \mathbf{FdUnitary}(A \oplus U, B \oplus U)$ the existence of $\lim \alpha$ implies $\lim \text{ex}^U \circ \alpha \simeq \text{ex}^U(\lim \alpha)$. The topology considered here is the strong operator topology (see 3.2.22). Moreover, recall that ex is defined in terms of a Σ function that arises from the limit of certain net of finite partial sums (see 2.2.32). Consequently, we may write:

$$\text{ex}^U(\lim_{d \in D} \alpha(d)) \simeq \lim_{\mathcal{F}(\mathbb{N})} \lim_{d \in D} \sigma_{\alpha(d)}$$

where $\sigma_{\alpha(d)}: \mathcal{F}(\mathbb{N}) \rightarrow \mathbf{FdUnitary}(A, B)$ is the net of finite partial sums corresponding to the sum of paths of $\alpha(d): A \oplus U \rightarrow B \oplus U$. Notice that we know that $\text{ex}^U(\lim \alpha)$ is well defined since $\lim \alpha$ is unitary and ex is total in $\mathbf{FdUnitary}$. Then, we would need to prove that the limit point

$$\lim_{d \in D} (\text{ex}^U \alpha(d)) \simeq \lim_{d \in D} \lim_{\mathcal{F}(\mathbb{N})} \sigma_{\alpha(d)}$$

exists. Notice that the difference between these expressions is the ordering of the limits so it is not immediate that the existence of $\text{ex}^U(\lim \alpha)$ implies that of $\lim \text{ex}^U \circ \alpha$. To prove such a limit exchange holds we often need both that:

- for each $d \in D$ the corresponding limit converges — in this case, $\lim_{J \in \mathcal{F}(\mathbb{N})} \sigma_{\alpha(d)}$ does exist for all $d \in D$ since ex is total — and that
- the net $\sigma_{\alpha(-)}$ *uniformly* converges with respect to D — in this case meaning that for each open neighbourhood U we can choose $d_U \in D$ such that for all $d \geq d_U$ and all $J \in \mathcal{F}(\mathbb{N})$ it holds that $\sigma_{\alpha(d)}(J) \in U$.

Such conditions are sufficient in the case of metric spaces and when limits are taken over sequences (see Moore-Osgood theorem, Theorem 2 from Chapter VII in [25]). However, it is not immediate whether the second condition is satisfied in our case, nor whether these two conditions would be sufficient in our general case.

The following subsection compares $(\mathbf{FdContraction}, \oplus, \text{ex})$ and $(\mathbf{CPTR}, \oplus, \text{ex})$ and shows that their traces cannot be reconciled, hinting at a fundamental difference between quantum and classical iterative loops.

3.3.3 Comparing the traces in $\mathbf{FdContraction}$ and $\mathbf{CPTR}$

In Section 3.2.1, $(\mathbf{CPTR}, \oplus, \text{ex})$ was given as an example of a totally traced category capturing the notion of classical iterative loops of quantum processes. The previous section has established that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is totally traced as well and, in this case, different execution paths may cancel out — additive inverses of morphisms exist — which is a fundamental characteristic of quantum control-flow. We show that the canonical functor from $\mathbf{FdContraction}$ to $\mathbf{CPTR}$ is not a traced monoidal functor, not even when the definition of traced monoidal functor is relaxed.

Definition 3.3.18. Let $E: \mathbf{FdContraction} \rightarrow \mathbf{CPTR}$ be the faithful functor that maps Hilbert spaces $A \in \mathbf{FdContraction}$ to C^* -algebras $B(A) \in \mathbf{CPTR}$ and which maps morphisms $f \in \mathbf{FdContraction}(A, B)$ to the $\mathbf{CPTR}$ map $E(f)$ defined for all $\rho \in B(A)$ as follows:

$$E(f)(\rho) = f \circ \rho \circ f^\dagger.$$

Notice that this functor is only *lax* monoidal: for every $A, B \in \mathbf{FdContraction}$ there is a canonical morphism $\theta_{A,B}: E(A) \oplus E(B) \rightarrow E(A \oplus B)$ given by:

$$\theta_{A,B}(\rho, \rho') = \begin{pmatrix} \rho & 0 \\ 0 & \rho' \end{pmatrix}$$

for all $(\rho, \rho') \in E(A) \oplus E(B)$, but $\theta_{A,B}$ is not an isomorphism. It is straightforward to check that θ is a natural transformation and, indeed, (E, θ) is a lax monoidal functor.

Moreover, considering that θ is injective, it has a left inverse $\phi: E(A \oplus B) \rightarrow E(A) \oplus E(B)$ acting as follows on every $\rho \in E(A \oplus B)$:

$$\phi \begin{pmatrix} \rho_{AA} & \rho_{AB} \\ \rho_{BA} & \rho_{BB} \end{pmatrix} = (\rho_{AA}, \rho_{BB}).$$

It is immediate to check that, indeed, $\phi \circ \theta = \text{id}$ but $\theta \circ \phi \neq \text{id}$. Notice that (E, ϕ) is a colax monoidal functor; in a situation such as this, we may define a weaker notion of traced monoidal functors.

Definition 3.3.19. Let $(\mathbf{C}, \oplus_{\mathbf{C}}, \text{Tr}_{\mathbf{C}})$ and $(\mathbf{D}, \oplus_{\mathbf{D}}, \text{Tr}_{\mathbf{D}})$ be totally traced categories and let $F: \mathbf{C} \rightarrow \mathbf{D}$ be a functor such that (F, θ) is lax monoidal, (F, ϕ) is colax monoidal and $\phi \circ \theta = \text{id}$. Then, F is said to be a *lax traced monoidal functor* if for all $A, B \in \mathbf{C}$ there is a function $\epsilon_{A,B}: \mathbf{D}(F(A), F(B)) \rightarrow \mathbf{D}(F(A), F(B))$ such that, for all $f \in \mathbf{C}(A \oplus U, B \oplus U)$, it maps:

$$F(\text{Tr}_{\mathbf{C}}^U(f)) \mapsto \text{Tr}_{\mathbf{D}}^{F(U)}(\phi \circ F(f) \circ \theta).$$

Alternatively, F is said to be a *colax traced monoidal functor* if $\epsilon_{A,B}$ maps in the other direction:

$$\text{Tr}_{\mathbf{D}}^{F(U)}(\phi \circ F(f) \circ \theta) \mapsto F(\text{Tr}_{\mathbf{C}}^U(f)).$$

The motivation behind this definition is two-fold: on one hand, the requirement (from the standard definition of traced monoidal functor) that F is strong monoidal is lifted and, on the other hand, instead of requiring that $F(\text{Tr}_{\mathbf{C}}(f)) = \text{Tr}_{\mathbf{D}}(\phi \circ F(f) \circ \theta)$ we only require that there is some arbitrary mapping between the results. This is an extremely weak notion of traced monoidal functor and yet, the functor $E: \mathbf{FdContraction} \rightarrow \mathbf{CPTR}$ is neither lax traced nor colax traced, making it apparent that the traces in $\mathbf{FdContraction}$ and $\mathbf{CPTR}$ are fundamentally different. This result is shown by counterexample.

Let σ be the symmetric braiding in $\mathbf{FdContraction}$, let $i: \mathbb{C} \rightarrow \mathbb{C}$ be the morphism in $\mathbf{FdContraction}$ that multiplies a complex number by the imaginary unit and let $H: \mathbb{C} \oplus \mathbb{C} \rightarrow \mathbb{C} \oplus \mathbb{C}$ be the Hadamard map:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Fix the following morphisms of type $\mathbb{C} \oplus \mathbb{C} \oplus \mathbb{C} \rightarrow \mathbb{C} \oplus \mathbb{C} \oplus \mathbb{C}$ in $\mathbf{FdContraction}$:

$$f = \text{id}_{\mathbb{C}} \oplus H \quad g = \text{id}_{\mathbb{C}} \oplus \sigma \quad h = (\text{id}_{\mathbb{C} \oplus \mathbb{C}} \oplus i) \circ g.$$

Applying ex to these morphisms yields:

$$\text{ex}^{\mathbb{C}}(f) = \text{id}_{\mathbb{C} \oplus \mathbb{C}} \quad \text{ex}^{\mathbb{C}}(g) = \text{id}_{\mathbb{C} \oplus \mathbb{C}} \quad \text{ex}^{\mathbb{C}}(h) = \text{id}_{\mathbb{C}} \oplus i \quad (3.30)$$

where the case of $\text{ex}^{\mathbb{C}}(f)$ follows from the example discussed at the beginning of Section 3.3.2, and those of $\text{ex}^{\mathbb{C}}(g)$ and $\text{ex}^{\mathbb{C}}(h)$ follow from the yanking axiom of traced categories. Let $f' : E(\mathbb{C} \oplus \mathbb{C}) \oplus E(\mathbb{C}) \rightarrow E(\mathbb{C} \oplus \mathbb{C}) \oplus E(\mathbb{C})$ be

$$f' = \phi \circ E(f) \circ \theta$$

and define g' and h' in the same manner. Let f' be decomposed as follows:

$$f' = \begin{pmatrix} f'_{00} : E(\mathbb{C} \oplus \mathbb{C}) \rightarrow E(\mathbb{C} \oplus \mathbb{C}) & f'_{01} : E(\mathbb{C}) \rightarrow E(\mathbb{C} \oplus \mathbb{C}) \\ f'_{10} : E(\mathbb{C} \oplus \mathbb{C}) \rightarrow E(\mathbb{C}) & f'_{11} : E(\mathbb{C}) \rightarrow E(\mathbb{C}) \end{pmatrix}$$

and similarly for g' and h' . Let $|\psi\rangle \in \mathbb{C} \oplus \mathbb{C}$ be $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and let $\rho = |\psi\rangle\langle\psi|$; then:

$$f'_{00}(\rho) = \frac{1}{2} \begin{pmatrix} 1 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 1 \end{pmatrix} \quad g'_{00}(\rho) = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad h'_{00}(\rho) = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

and, for all $n \in \mathbb{N}$,

$$\begin{aligned} f'_{01}(f'_{11})^n f'_{10}(\rho) &= \frac{1}{4} \cdot \frac{1}{2^{n+1}} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ g'_{01}(g'_{11})^n g'_{10}(\rho) &= \frac{1}{2} \cdot \frac{1}{2^{n+1}} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ h'_{01}(h'_{11})^n h'_{10}(\rho) &= \frac{1}{2} \cdot \frac{1}{2^{n+1}} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

so that, when all of these paths are summed up, we obtain:

$$\begin{aligned} \text{ex}^{E(\mathbb{C})}(f')(\rho) &= \frac{1}{2} \begin{pmatrix} 1 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 1 \end{pmatrix} \\ \text{ex}^{E(\mathbb{C})}(g')(\rho) &= \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \text{ex}^{E(\mathbb{C})}(h')(\rho) &= \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned} \quad (3.31)$$

According to (3.30), $\text{ex}^{\mathbb{C}}(f) = \text{ex}^{\mathbb{C}}(g)$ and, hence, $E(\text{ex}^{\mathbb{C}}(f)) = E(\text{ex}^{\mathbb{C}}(g))$; however, according to (3.31), $\text{ex}^{E(\mathbb{C})}(f') \neq \text{ex}^{E(\mathbb{C})}(g')$ since they are distinct at least at

ρ . Consequently, there exists no function acting as $E(\text{ex}^{\mathbb{C}}(f)) \mapsto \text{ex}^{E(\mathbb{C})}(f')$ both for f and g , preventing E from being a lax traced monoidal functor. On the other hand, $\text{ex}^{\mathbb{C}}(g) \neq \text{ex}^{\mathbb{C}}(h)$ and $E(\text{ex}^{\mathbb{C}}(g)) \neq E(\text{ex}^{\mathbb{C}}(h))$ due to E being faithful, but $\text{ex}^{E(\mathbb{C})}(g') = \text{ex}^{E(\mathbb{C})}(h')$ so that a function acting as $\text{ex}^{E(\mathbb{C})}(g') \mapsto E(\text{ex}^{\mathbb{C}}(g))$ both for g and h cannot exist, thus preventing E from being colax traced monoidal either.

3.3.4 Open questions in Contraction

Sections 3.2 and 3.3 have been organised in such a way that all of the results whose proofs depend on finite-dimensionality appear in Section 3.3.2. In this section, we discuss the role that finite-dimensionality plays in these proofs; the objective being to shed some light on whether or not $(\mathbf{Contraction}, \oplus, \text{ex})$ is a partially traced category and, if it were, identify the obstacles in the way of proving so.

- Lemma 3.3.11 establishes that every contraction in $\mathbf{FdHilb}$ can be traced using the kernel-image trace. Recall that $(\mathbf{Hilb}, \oplus, \text{Tr}_{\text{ki}})$ is a partially traced category (see Proposition 3.2.19) and, hence, we may ask whether every contraction in $\mathbf{Hilb}$ can be traced. The proof of Lemma 3.3.11 relies on the fact that $\text{im}(\text{id} - f_{UU})$ is a closed subspace of U , which is trivial to prove since it is a finite-dimensional space. Unfortunately, it is unclear whether $\text{im}(\text{id} - f_{UU})$ would be closed for an arbitrary morphism $f: A \oplus U \rightarrow B \oplus U$ in $\mathbf{Contraction}$.
- Lemma 3.3.12 establishes that every completely nonunitary $f_1: H_1 \rightarrow H_1$ in $\mathbf{FdContraction}$ satisfies that $f_1^{\dim H_1}$ is a *strict* contraction. This claim cannot be properly phrased in $\mathbf{Contraction}$ since $\dim H_1$ may be infinite. Moreover, Example 3.3.20 provides a contraction g whose component g_{UU} is completely nonunitary but satisfies $\|g_{UU}^n\|_{\text{op}} = 1$ for all $n \in \mathbb{N}$, giving strong evidence that a result akin to Lemma 3.3.12 cannot be established in $\mathbf{Contraction}$.
- Lemma 3.3.13 establishes an upper bound for $\sum \|f_{UU}^k f_{UA}\|_{\text{op}}$. Its proof heavily relies on Lemma 3.3.12 and, hence, it is unlikely its proof can be generalised to $\mathbf{Contraction}$. Furthermore, Example 3.3.20 provides a contraction g for which $\sum \|g_{UU}^k g_{UA}\|_{\text{op}}$ cannot be bounded from above; thus, Lemma 3.3.13 does not hold in $\mathbf{Contraction}$.

These three lemmas were used in Section 3.3.2 to establish that every contraction in $\mathbf{FdHilb}$ is tail vanishing (Lemma 3.3.14). Due to the above, the approach from Section 3.3.2 cannot be used to prove that every contraction in $\mathbf{Hilb}$ is tail vanishing.

However, recall that Lemma 3.3.10 establishes that $\lim f_{BU} f_{UU}^n \simeq 0$ for every contraction $f: A \oplus U \rightarrow B \oplus U$ in **Hilb** and, hence, for f to be U -tail vanishing we only need to show that:

$$\lim_{J \in \mathcal{F}(\mathbb{N})} \Sigma \{f_{BU} f_{UU}^j f_{UA}\}_{\tilde{J}} \simeq 0$$

where $\tilde{J}$ is the set of ‘gaps’ in J (see Definition 3.2.29). Unlike the previous lemmas, this equation does seem to hold for the contraction provided in Example 3.3.20 and, if this were the case for all contractions, it would imply that $(\mathbf{Contraction}, \oplus, \text{ex})$ is a partially traced category, as established in Lemma 3.2.30. Consequently, there is some hope that **Contraction** is partially traced with respect to the execution formula, but proving so would require an approach different from the one presented in Section 3.3.2.

Example 3.3.20. Let U , A and B be separable Hilbert spaces of countably infinite dimension, each with an orthonormal basis given by $\{u_i\}_{i \in \mathbb{N}}$, $\{a_i\}_{i \in \mathbb{N}}$ and $\{b_i\}_{i \in \mathbb{N}}$, respectively. Let $g: A \oplus U \rightarrow B \oplus U$ be a morphism in **Contraction** where, for all $n \in \mathbb{N}$:

$$g(u_n) = \begin{pmatrix} \sqrt{\frac{1}{2^n}} \cdot b_n \\ \sqrt{1 - \frac{1}{2^n}} \cdot u_{n+1} \end{pmatrix}$$

and $g(a_n) = u_n$. Even though for every $v \in U$ there is some $c \in \mathbb{R}$ strictly smaller than 1 such that $\|g_{UU}(v)\| \leq c \cdot \|v\|$, notice that $\|g_{UU}\|_{\text{op}} = 1$; the same applies to $\|g_{UU}^n\|_{\text{op}} = 1$ for every $n \in \mathbb{N}$. We thank Hari Bercovici for providing us with this example.

On another note, it may be argued that the trace in $(\mathbf{FdContraction}, \oplus, \text{ex})$ is not appropriate to capture iterative loops: we would expect that different execution paths — e.g. f_{BA} and $f_{BU} f_{UU}^2 f_{UA}$ — on the same input would produce their output at different time-steps. However, **FdContraction** cannot capture the notion of time since this is an infinite domain and **FdContraction** can only deal with finite-dimensional Hilbert spaces. To this end, the following section introduces a particular category of quantum processes over time on which the execution formula is a well-defined trace, thus providing a satisfactory formalisation of iterative loops in quantum computing.

3.4 Coherent quantum processes over time

When the output of a process depends only on its input at the previous time-step, the time domain may be omitted from the mathematical formalism. For instance, given any sequence of operations $U_n \in \mathbf{Unitary}(H, H)$ and any initial state $|\psi_0\rangle \in H$ the

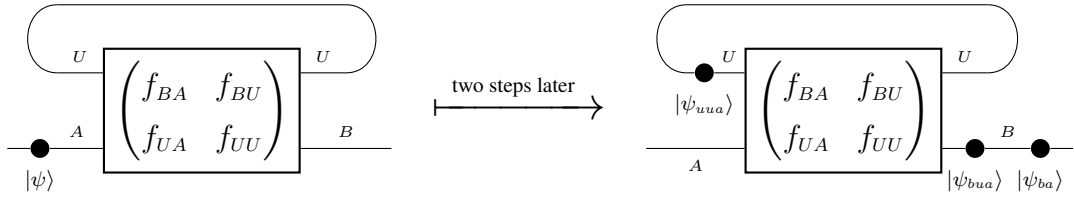


Figure 3.1: A state $|\psi\rangle$ is given as input to a quantum iterative process; two time-steps later, the state is in a superposition $|\psi_{ba}\rangle + |\psi_{bua}\rangle + |\psi_{uua}\rangle$ where $|\psi_{ba}\rangle = f_{BA}|\psi\rangle$, $|\psi_{bua}\rangle = f_{BU}f_{UA}|\psi\rangle$ and $|\psi_{uua}\rangle = f_{UU}f_{UA}|\psi\rangle$. The term $|\psi_{ba}\rangle$ ‘leaves’ the system a step earlier than $|\psi_{bua}\rangle$.

state after t steps can be easily represented as

$$|\psi_t\rangle = U_t \dots U_2 U_1 |\psi_0\rangle.$$

Thus, the time domain may be abstracted away from the state and represented by the sequence of operations that remains to be applied. In contrast, if we turn to open quantum walks (see Section 1.1.2) we encounter situations as the one depicted in Figure 3.1:¹² if the walker enters device $f: A \oplus U \rightarrow B \oplus U$ through A at time-step t_0 , some of its amplitude will be found in B at $t_0 + 1$ but some will traverse the loop and eventually leave at a later time-step. Consequently, the output is not only localised at $t_0 + 1$ but instead its amplitude spreads throughout $[t_0 + 1, \infty)$. This situation may be captured by functions describing the input and output over time. For instance, in the case of Figure 3.1, we may use functions $a: \mathbb{Z} \rightarrow \mathbb{C}$ and $b: \mathbb{Z} \rightarrow \mathbb{C}$ to describe the input (state on A) and output (state on B) respectively:

$$a(t) = \begin{cases} \psi & \text{if } t = t_0 \\ 0 & \text{otherwise} \end{cases} \quad b(t) = \begin{cases} 0 & \text{if } t \leq t_0 \\ f_{BA}(\psi) & \text{if } t = t_0 + 1 \\ f_{BU}f_{UU}^{t-t_0-2}f_{UA}(\psi) & \text{otherwise.} \end{cases}$$

Then, $|b(t)|^2$ determines the probability of finding the walker in B if we were to measure at time-step t . Thus, we would require that the square of the amplitudes sum up to one:

$$\sum_{t=-\infty}^{\infty} |b(t)|^2 = 1$$

but, in order to work with vector spaces, we instead consider states up to normalisation and, hence, impose that functions describing a quantum state must be square-

¹²Figure 3.1 is a copy of Figure 1.1, reproduced here for the reader’s convenience.

summable:

$$\sum_{t=-\infty}^{\infty} |a(t)|^2 < \infty.$$

We can now define a Hilbert space of square-summable functions of type $\mathbb{Z} \rightarrow \mathbb{C}$.

Definition 3.4.1. Let ℓ^2 be the complex Hilbert space defined as follows:¹³

- vectors are functions $\psi: \mathbb{Z} \rightarrow \mathbb{C}$ such that

$$\sum_{t=-\infty}^{\infty} |\psi(t)|^2 < \infty$$

- addition and scalar multiplication are defined pointwise, i.e. for $\psi, \phi \in \ell^2$ and $\alpha \in \mathbb{C}$:

$$(\psi + \phi)(t) = \psi(t) + \phi(t) \quad (\alpha\psi)(t) = \alpha \cdot \psi(t)$$

- inner product is defined as follows:

$$\langle \phi | \psi \rangle = \sum_{t=-\infty}^{\infty} \phi(t)^* \cdot \psi(t)$$

Proving that the inner product given above is well-defined is a common exercise in courses on functional analysis; it can be shown using the Cauchy-Schwarz inequality in $\mathbb{R}^n$ along with absolute convergence in $\mathbb{C}$. We should also check that ℓ^2 is indeed Cauchy complete but, once again, this is a well-known fact and it is omitted here for the sake of brevity. The goal of this section is to discuss linear maps $\ell^2 \rightarrow \ell^2$ that may be interpreted as operators that act on quantum states over the time domain. When discussing these, the convolution of functions in ℓ^2 will take a fundamental role.

Definition 3.4.2. For any two functions $f, g \in \ell^2$ we define $g * f: \mathbb{Z} \rightarrow \mathbb{C}$ acting on every $t \in \mathbb{Z}$ as follows:¹⁴

$$(g * f)(t) = \sum_{\tau=-\infty}^{\infty} g(t - \tau) \cdot f(\tau).$$

We refer to $g * f$ as the *convolution* of f and g .

¹³This is slightly different from the usual ℓ^2 space, which is defined as the space of square-summable sequences. In essence, instead of a vector being a sequence (a function $\mathbb{N} \rightarrow \mathbb{C}$), we use $\mathbb{Z}$ as the domain to describe both directions of time.

¹⁴It can be shown that $f, g \in \ell^2$ implies that $g * f$ is well-defined; the argument is similar to the one used to prove that the inner product in ℓ^2 is well-defined, reproducing it for $(g * f)(t)$ for every $t \in \mathbb{Z}$ and using that S_τ is an isometry. However, the function $g * f$ need not be square-summable.

It is straightforward to check that convolution is associative and commutative. Moreover, let $\delta \in \ell^2$ be the function:

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.32)$$

then $\delta * f = f$ for any $f \in \ell^2$ and, due to commutativity, $f * \delta = f$ as well. Moreover, convolution distributes over pointwise addition of functions, as established below.

Proposition 3.4.3. *Let $f, g, h \in \ell^2$ and let $(f + g)(t) = f(t) + g(t)$, then:*

$$\begin{aligned} h * (f + g) &= (h * f) + (h * g) \\ (f + g) * h &= (f * h) + (g * h) \end{aligned}$$

Proof. By explicit calculation, for all $t \in \mathbb{Z}$:

$$\begin{aligned} (h * (f + g))(t) &= \sum_{\tau=-\infty}^{\infty} h(t - \tau) \cdot (f + g)(\tau) \\ &= \sum_{\tau=-\infty}^{\infty} h(t - \tau) \cdot (f(\tau) + g(\tau)) \\ &= \left(\sum_{\tau=-\infty}^{\infty} h(t - \tau) \cdot f(\tau) \right) + \left(\sum_{\tau=-\infty}^{\infty} h(t - \tau) \cdot g(\tau) \right) \\ &= (h * f)(t) + (h * g)(t) = ((h * f) + (h * g))(t). \end{aligned}$$

This proves the first identity from the claim; the second one follows from it due to commutativity of convolution. $\square$

The ℓ^2 space lets us formalise the notion of quantum states over the discrete-time domain. Similarly, the definition below describes a Hilbert space L^2 of square-integrable functions, formalising quantum states over continuous time. An appropriate notion of convolution of functions in L^2 is also provided.

Definition 3.4.4. Let L^2 be the complex Hilbert space defined as follows:

- vectors are functions $\psi: \mathbb{R} \rightarrow \mathbb{C}$ such that

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty$$

- addition and scalar multiplication are defined pointwise, *i.e.* for $\psi, \phi \in L^2$ and $\alpha \in \mathbb{C}$:

$$(\psi + \phi)(t) = \psi(t) + \phi(t) \quad (\alpha\psi)(t) = \alpha \cdot \psi(t)$$

- inner product is defined as:

$$\langle \phi | \psi \rangle = \int_{-\infty}^{\infty} \phi(t)^* \cdot \psi(t) dt.$$

Definition 3.4.5. For any two functions $f, g \in L^2$ we define $g * f: \mathbb{R} \rightarrow \mathbb{C}$ acting on every $t \in \mathbb{R}$ as follows:

$$(g * f)(t) = \int_{-\infty}^{\infty} g(t - \tau) \cdot f(\tau) d\tau.$$

Remark 3.4.6. Convolution in L^2 is associative and commutative, but it has no unit. Physicists often circumvent this by defining δ to be the Dirac delta: a map that is zero everywhere except at 0, and whose square-integral is equal to 1; but, evidently, no such a function exists. More formally, we may consider a sequence of functions δ_n in L^2 whose square-integral equals 1 and whose support is centered at zero, becoming narrower as n tends to infinity. Such a sequence does not converge, but we can nonetheless choose an n large enough so that the support of δ_n is sufficiently narrow for the approximation $\delta_n * f \approx f$ to be satisfactory for practical purposes. In what follows, $\delta_\bullet \in L^2$ will be used to denote a choice of δ_n where n is some arbitrary large number.

3.4.1 The discrete-time Fourier transform (*Preamble*)

A fundamental result in physics and engineering establishes that the convolution of two functions in L^2 can be calculated by pointwise multiplication of the Fourier transformed functions.¹⁵ The main goal of this subsection is to introduce its ℓ^2 version, known as the discrete-time Fourier transform. Both the discrete-time Fourier transform and its convolution theorem (Theorem 3.4.9) will be key ingredients in subsequent discussions of quantum processes over discrete time.

Definition 3.4.7. The *discrete-time Fourier transform (DTFT)* is a linear map $\mathcal{F}_d$ that sends any $f \in \ell^2$ to the function $\mathcal{F}_d[f]: \mathbb{R} \rightarrow \mathbb{C}$ defined below:

$$\mathcal{F}_d[f](\omega) = \sum_{t=-\infty}^{\infty} f(t) e^{-i\omega t}.$$

It is well-known (although not immediate) that if $f \neq 0$ then $\mathcal{F}_d[f]$ is a periodic function with period 2π ; consequently, $\mathcal{F}_d[f]$ cannot possibly be square-integrable,

¹⁵See Section 9.2 from [7] to learn about the theory around the Fourier transform and Proposition 9.2.3 for this particular result).

$\mathcal{F}_d[f] \notin L^2$. Nevertheless, the DTFT has a left inverse obtained by integrating over a full 2π period:

$$\mathcal{F}_d^l[g] = \frac{1}{2\pi} \int_0^{2\pi} g(\omega) e^{i\omega t} d\omega \quad (3.33)$$

so that $(\mathcal{F}_d^l \circ \mathcal{F}_d)[f] = f$ for any $f \in \ell^2$. This can be derived from the following identity, which is satisfied for integers $n \in \mathbb{Z}$:

$$\int_0^{2\pi} e^{i\omega n} d\omega = 2\pi \cdot \delta(n) \quad (3.34)$$

where, for $n = 0$ the integral is just $\int_0^{2\pi} d\omega = 2\pi$ and for any other integer $n \in \mathbb{Z}$, the interval of integration contains $|n|$ full periods of the function $e^{i\omega n}$ and, hence, for each $\omega \in (0, \pi)$ the value of $e^{i\omega}$ cancels out with $e^{i(\omega+\pi)}$ and the integral evaluates to 0. Notice that the existence of this left inverse implies that $\mathcal{F}_d$ is an injective map.

Remark 3.4.8. When functions $\psi \in \ell^2$ are interpreted as (non-normalised) quantum states whose amplitude spreads over discrete time, the DTFT is understood to yield a function $\mathcal{F}_d[\psi]: \mathbb{R} \rightarrow \mathbb{C}$ describing the same quantum state over the frequency domain. This interpretation becomes clearer after examining the identity $\psi = (\mathcal{F}_d^l \circ \mathcal{F}_d)[\psi]$:

$$\psi(t) = \frac{1}{2\pi} \int_0^{2\pi} \mathcal{F}_d[\psi](\omega) \cdot \gamma_\omega(t) d\omega$$

where $\gamma_\omega(t) = e^{i\omega t}$ is the *plane wave* of angular frequency ω . Consequently, the state ψ is described as a superposition (*i.e.* a linear combination) of plane waves γ_ω for all frequencies $\omega \in \mathbb{R}$, where each complex number $\mathcal{F}_d[\psi](\omega)$ determines the amplitude (and phase) of each plane wave.

The convolution theorem for functions in ℓ^2 is presented below.

Theorem 3.4.9. For any $\omega \in \mathbb{R}$ and any two functions $f, g \in \ell^2$:

$$\mathcal{F}_d[g * f](\omega) = \mathcal{F}_d[g](\omega) \cdot \mathcal{F}_d[f](\omega).$$

Proof. The theorem follows from a simple rearrangement of terms:

$$\begin{aligned}
\mathcal{F}_d[g](\omega) \cdot \mathcal{F}_d[f](\omega) &= \\
&= \left(\sum_{t'=-\infty}^{\infty} g(t') e^{-i\omega t'} \right) \left(\sum_{\tau=-\infty}^{\infty} f(\tau) e^{-i\omega \tau} \right) \quad (\text{def. } \mathcal{F}_d) \\
&= \sum_{t'=-\infty}^{\infty} \sum_{\tau=-\infty}^{\infty} g(t') f(\tau) e^{-i\omega(t'+\tau)} \quad (\text{rearrangement}) \\
&= \sum_{t=-\infty}^{\infty} \sum_{\tau=-\infty}^{\infty} g(t-\tau) f(\tau) e^{-i\omega t} \quad (\text{substitute } t' = t - \tau) \\
&= \sum_{t=-\infty}^{\infty} (g * f)(t) e^{-i\omega t} \quad (\text{def. convolution}) \\
&= \mathcal{F}[g * f](\omega). \quad (\text{def. } \mathcal{F}_d) \quad \square
\end{aligned}$$

The Fourier transform $\mathcal{F}: L^2 \rightarrow L^2$ is unitary, implying that $\|\mathcal{F}[\psi]\| = \|\psi\|$ for any function $\psi \in L^2$ (see Theorem 9.2.1 from [7]). In contrast, the function $\mathcal{F}_d[\psi]$ is not even square-integrable. Nevertheless, a similar result can be obtained for the DTFT by integrating the values of $|\mathcal{F}_d[\psi](\omega)|^2$ over a full period, as established below.

Proposition 3.4.10. *For any $\psi \in \ell^2$ the following identity is satisfied:*

$$\|\psi\|^2 = \frac{1}{2\pi} \int_0^{2\pi} |\mathcal{F}_d[\psi](\omega)|^2 d\omega.$$

Proof. The norm of any $\psi \in \ell^2$ is:

$$\|\psi\| = \sqrt{\sum_{t=-\infty}^{\infty} \psi(t)^* \cdot \psi(t)}$$

as determined by the inner product in ℓ^2 . Then:

$$\begin{aligned}
\frac{1}{2\pi} \int_0^{2\pi} |\mathcal{F}_d[\psi](\omega)|^2 d\omega &= \\
&= \frac{1}{2\pi} \int_0^{2\pi} \left| \sum_{t=-\infty}^{\infty} \psi(t) e^{-i\omega t} \right|^2 d\omega \quad (\text{def. } \mathcal{F}_d) \\
&= \frac{1}{2\pi} \int_0^{2\pi} \left(\sum_{\tau=-\infty}^{\infty} \psi(\tau)^* e^{i\omega \tau} \right) \left(\sum_{t=-\infty}^{\infty} \psi(t) e^{-i\omega t} \right) d\omega \quad (\text{def. } |\alpha|^2 \text{ for } \alpha \in \mathbb{C}) \\
&= \frac{1}{2\pi} \sum_{t=-\infty}^{\infty} \sum_{\tau=-\infty}^{\infty} \psi(\tau)^* \cdot \psi(t) \cdot \int_0^{2\pi} e^{i\omega(\tau-t)} d\omega \quad (\text{rearrange})
\end{aligned}$$

$$\begin{aligned}
&= \sum_{t=-\infty}^{\infty} \sum_{\tau=-\infty}^{\infty} \psi(\tau)^* \cdot \psi(t) \cdot \delta(\tau - t) && \text{(identity (3.34))} \\
&= \sum_{t=-\infty}^{\infty} \psi(t)^* \cdot \psi(t) = \|\psi\|^2. && \text{(def. } \delta \text{ and } \|\psi\|) \quad \square
\end{aligned}$$

3.4.2 Linear shift invariant maps

Physical processes that act on states over an infinite domain are often *shift invariant*. The input-output behaviour of a shift invariant device $D: \ell^2 \rightarrow \ell^2$ acting on quantum states is independent of the time-step the input reaches the device. To formalise this notion we define a *shift operator* $S_\tau: \ell^2 \rightarrow \ell^2$ for all $\tau \in \mathbb{Z}$:

$$S_\tau[\psi](t) = \psi(t - \tau)$$

which captures the passing of an amount τ of time. Then, a *linear shift invariant (LSI)* map $D: \ell^2 \rightarrow \ell^2$ is a linear map that commutes with every shift operator:

$$S_\tau D = D S_\tau$$

for all $\tau \in \mathbb{Z}$. Crucially, any operator $\ell^2 \rightarrow \ell^2$ that is both linear and shift invariant can be fully characterised by a function in ℓ^2 , as shown below.

Proposition 3.4.11. *For any LSI map $D: \ell^2 \rightarrow \ell^2$, there is a unique function $\chi^D \in \ell^2$ such that for all $\psi \in \ell^2$:*

$$D[\psi] = \chi^D * \psi.$$

Such a function is precisely $\chi^D = D[\delta]$ where δ is the unit of the convolution (3.32).

Proof. Considering δ is the unit of convolution, we have that:

$$\psi(t) = (\delta * \psi)(t) = \sum_{\tau=-\infty}^{\infty} \delta(t - \tau) \cdot \psi(\tau) = \sum_{\tau=-\infty}^{\infty} S_\tau[\delta](t) \cdot \psi(\tau).$$

Then, due to D being linear and shift invariant we can compute $D[\psi]$ as follows:

$$D[\psi] = \sum_{\tau=-\infty}^{\infty} D[S_\tau[\delta]] \cdot \psi(\tau) = \sum_{\tau=-\infty}^{\infty} S_\tau[D[\delta]] \cdot \psi(\tau) = D[\delta] * \psi.$$

Therefore, if we let $\chi^D = D[\delta]$ then $D[\psi] = \chi^D * \psi$ as claimed. To prove uniqueness, assume there is a different function ξ^D such that $D[\psi] = \xi^D * \psi$ for all $\psi \in \ell^2$. It then follows that $D[\delta] = \xi^D * \delta = \xi^D$ due to δ being the unit of convolution and, hence, we conclude that $\xi^D = \chi^D$. $\square$

Due to convolution being associative, it is apparent that composition of LSI maps $D, D': \ell^2 \rightarrow \ell^2$ corresponds to convolution of their characteristic functions:

$$(D' \circ D)[\psi] = (\chi^{D'} * \chi^D) * \psi.$$

Notice that $D' \circ D$ is again a LSI map $\ell^2 \rightarrow \ell^2$ and, hence, $\chi^{D'} * \chi^D$ is necessarily in ℓ^2 . A similar result holds for LSI maps $D: L^2 \rightarrow L^2$; however, in this case the characterisation of D as a function in L^2 is only an approximation. This is due to the lack of a unit of convolution in L^2 (see Remark 3.4.6).

Proposition 3.4.12. *For any LSI map $D: L^2 \rightarrow L^2$, let $\chi^D \in L^2$ be $\chi^D = D[\delta_\bullet]$; then, for all $\psi \in L^2$:*

$$D[\psi] \approx \chi^D * \psi.$$

Proof. The claim follows from the same argument as that for ℓ^2 (Proposition 3.4.11), provided that we are dealing with approximations, as discussed in Remark 3.4.6. $\square$

For any finite set A , the vector space $\oplus_A \ell^2$ over the field $\mathbb{C}$ has collections

$$\Psi = \{\psi_a \in \ell^2\}_{a \in A}$$

as vectors, with addition and scalar multiplication defined index-wise:

$$\Phi + \Psi = \{\phi_a + \psi_a\}_{a \in A}$$

$$\alpha \Psi = \{\alpha \psi_a\}_{a \in A}$$

The inner product in $\oplus_A \ell^2$ is defined in terms of the inner product in ℓ^2 :

$$\langle \Phi | \Psi \rangle = \sum_{a \in A} \langle \phi_a | \psi_a \rangle$$

Considering that A is a finite set, it follows that $\oplus_A \ell^2$ is complete and, hence, a Hilbert space. For each $a \in A$ there is a projection $\pi_a: \oplus_A \ell^2 \rightarrow \ell^2$ that yields the component at index a , $\pi_a[\Psi] = \psi_a$, and an injection $\iota_a: \ell^2 \rightarrow \oplus_A \ell^2$ that maps $\phi \in \ell^2$ to a collection $\{\phi_a\}_{a \in A}$ where $\phi_a = \phi$ and all other $\phi_{a'}$ are the constant zero function. It is trivial to check that for all $\Psi \in \oplus_A \ell^2$ the following identity is satisfied:

$$\Psi = \sum_{a \in A} (\iota_a \pi_a)[\Psi] = \sum_{a \in A} \iota_a[\psi_a]. \quad (3.35)$$

We can then consider linear maps $D: \oplus_A \ell^2 \rightarrow \oplus_B \ell^2$ where both A and B are finite sets, and say that these are shift invariant iff for all $(b, a) \in B \times A$ the map $\pi_b D \iota_a: \ell^2 \rightarrow \ell^2$ is shift invariant. Once again, convolution lets us describe these maps in terms of functions in ℓ^2 , as established in the following proposition.

Proposition 3.4.13. *Let A and B be two arbitrary finite sets. For any LSI map $D: \oplus_A \ell^2 \rightarrow \oplus_B \ell^2$ and for each $(b, a) \in B \times A$ let $\chi_{ba}^D \in \ell^2$ be*

$$\chi_{ba}^D = (\pi_b D \iota_a)[\delta]$$

Then, for all $\Psi \in \oplus_A \ell^2$:

$$D[\Psi] = \left\{ \sum_{a \in A} \chi_{ba}^D * \psi_a \right\}_{b \in B}.$$

Proof. Since for each $(b, a) \in B \times A$ the map $\pi_b D \iota_a$ is time shift invariant map, we may use Proposition 3.4.11 to compute the B -indexed family $D[\Psi]$ in terms of convolution with the each of characteristic functions χ_{ba}^D :

$$\begin{aligned} D[\Psi] &= \sum_{b \in B} (\iota_b \pi_b) [D[\Psi]] && \text{(equation (3.35))} \\ &= \sum_{b \in B} (\iota_b \pi_b) \left[D \left[\sum_{a \in A} \iota_a [\psi_a] \right] \right] && \text{(equation (3.35))} \\ &= \sum_{b \in B} \iota_b \left[\sum_{a \in A} (\pi_b D \iota_a) [\psi_a] \right] && \text{(linearity)} \\ &= \sum_{b \in B} \iota_b \left[\sum_{a \in A} \chi_{ba}^D * \psi_a \right] && \text{(Proposition 3.4.11)} \\ &= \left\{ \sum_{a \in A} \chi_{ba}^D * \psi_a \right\}_{b \in B}. && \text{(equation (3.35))} \quad \square \end{aligned}$$

Therefore, an LSI map $D: \oplus_A \ell^2 \rightarrow \oplus_B \ell^2$ is characterised by a collection of ℓ^2 functions: $\{\chi_{ba}^D\}_{(b,a) \in B \times A}$. Moreover, if we arrange this collection into a $|B| \times |A|$ matrix and, similarly, represent any $\Psi \in \oplus_A \ell^2$ as an $|A| \times 1$ matrix of its $\psi_a \in \ell^2$ components, the previous proposition establishes that:

$$D[\Psi] = \begin{pmatrix} \chi_{ba}^D & \cdots & \chi_{ba'}^D \\ \vdots & \ddots & \vdots \\ \chi_{b'a}^D & \cdots & \chi_{b'a'}^D \end{pmatrix} \circledast \begin{pmatrix} \psi_a \\ \vdots \\ \psi_{a'} \end{pmatrix}$$

where $\circledast$ denotes formal matrix multiplication whose entry-wise multiplication is replaced by convolution. Furthermore, considering that the DTFT is an injective map — due to it having a left inverse (3.33) — any Ψ and D can be uniquely determined by the collection of their Fourier transformed components:

$$\begin{aligned} \widehat{\Psi} &= \{\mathcal{F}_d[\psi_a]\}_{a \in A} \\ \widehat{D} &= \{\mathcal{F}_d[\chi_{ba}^D]\}_{(b,a) \in B \times A} \end{aligned}$$

where the hat notation is used to indicate that $\widehat{D}$ and $\widehat{\Psi}$ are representations in the frequency domain (see Remark 3.4.8). For each $\omega \in \mathbb{R}$ let:

$$\begin{aligned}\widehat{\Psi}_\omega &= \{\mathcal{F}_d[\psi_a](\omega)\}_{a \in A} \\ \widehat{D}_\omega &= \{\mathcal{F}_d[\chi_{ba}^D](\omega)\}_{(b,a) \in B \times A}\end{aligned}$$

and notice that the elements of these collections are complex numbers. Then, $\widehat{D}$ can be identified with a collection of linear maps $\widehat{D}_\omega: \mathbb{C}^{|A|} \rightarrow \mathbb{C}^{|B|}$ indexed by $\omega \in \mathbb{R}$.

The importance of these remarks is due to the convolution theorem (Theorem 3.4.9), which turns the convolution of entries used in $\circledast$ into standard multiplication, so that for each $\omega \in \mathbb{R}$:

$$(\widehat{D[\Psi]})_\omega = \widehat{D}_\omega \cdot \widehat{\Psi}_\omega \quad (3.36)$$

where $-\cdot-$ corresponds to standard multiplication of matrices. Considering that all of the operations involved in this discussion are associative, composition of LSI maps turns into index-wise composition of the linear maps that comprise them:

$$(\widehat{D' \circ D})_\omega = \widehat{D'}_\omega \circ \widehat{D}_\omega$$

In particular, it unravels a deep connection between the algebra of LSI maps and the algebra of linear maps on finite-dimensional Hilbert spaces. This motivates the definition of a category of LSI processes.

Definition 3.4.14. Let **LSI** be the category whose objects are finite sets and whose morphisms $f: A \rightarrow B$ are LSI maps $\oplus_A \ell^2 \rightarrow \oplus_B \ell^2$ represented as their $\mathbb{R}$ -indexed collections:

$$f = \{\widehat{f}_\omega: \mathbb{C}^{|A|} \rightarrow \mathbb{C}^{|B|}\}_{\omega \in \mathbb{R}}.$$

Composition is given by index-wise composition of the linear maps:

$$g \circ f = \{\widehat{g}_\omega \circ \widehat{f}_\omega\}_{\omega \in \mathbb{R}}$$

and, for any finite set A , its identity morphism is the $\mathbb{R}$ -indexed family of copies of the identity map $\widehat{\text{id}}_\omega: \mathbb{C}^{|A|} \rightarrow \mathbb{C}^{|A|}$.

Remark 3.4.15. Unfortunately, the definition of **LSI** cannot be reproduced for linear shift invariant maps over L^2 since, according to Proposition 3.4.12, $D[\psi]$ is only approximately equal to $\chi^D * \psi$. It would perhaps be possible to define a category where equality is replaced by approximation up to certain factor; unfortunately, every time we compose we would reduce the precision of the approximation.

The following results establish how the $\widehat{D}_\omega$ components of a LSI map interact with the norm in $\oplus_A \ell^2$. The ultimate goal is to define a category of LSI contractions.

Lemma 3.4.16. *For any finite set A and any vector $\Psi \in \oplus_A \ell^2$ the following identity is satisfied:*

$$\|\Psi\|^2 = \frac{1}{2\pi} \int_0^{2\pi} \|\widehat{\Psi}_\omega\|^2 d\omega.$$

Proof. The claim follows from Proposition 3.4.10, as shown below:

$$\begin{aligned} \|\Psi\|^2 &= \sum_{a \in A} \|\psi_a\|^2 && \text{(norm in } \oplus_A \ell^2) \\ &= \sum_{a \in A} \frac{1}{2\pi} \int_0^{2\pi} |\mathcal{F}_d[\psi_a](\omega)|^2 d\omega && \text{(Proposition 3.4.10)} \\ &= \frac{1}{2\pi} \int_0^{2\pi} \sum_{a \in A} |\mathcal{F}_d[\psi_a](\omega)|^2 d\omega && \text{(rearrangement)} \\ &= \frac{1}{2\pi} \int_0^{2\pi} \|\widehat{\Psi}_\omega\|^2 d\omega && \text{(norm in } \mathbb{C}^{|A|}) \quad \square \end{aligned}$$

Theorem 3.4.17. *Let $D: \oplus_A \ell^2 \rightarrow \oplus_B \ell^2$ be an arbitrary LSI map represented by a collection of linear maps $\{\widehat{D}_\omega: \mathbb{C}^{|A|} \rightarrow \mathbb{C}^{|B|}\}_{\omega \in \mathbb{R}}$. If for all $\omega \in \mathbb{R}$ the linear map $\widehat{D}_\omega$ is a contraction, then D is itself a contraction.*

Proof. According to the previous lemma, for each vector $\Psi \in \oplus_A \ell^2$ we may calculate the norm of $D[\Psi]$ as follows:

$$\|D[\Psi]\| = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} \|\widehat{(D[\Psi])}_\omega\|^2 d\omega}$$

and, due to equation (3.36) and the assumption that $\widehat{D}_\omega$ is a contraction,

$$\|\widehat{(D[\Psi])}_\omega\| = \|\widehat{D}_\omega \cdot \widehat{\Psi}_\omega\| \leq \|\widehat{\Psi}_\omega\|.$$

It then follows that:

$$\|D[\Psi]\| \leq \sqrt{\frac{1}{2\pi} \int_0^{2\pi} \|\widehat{\Psi}_\omega\|^2 d\omega}$$

and, according to the previous lemma, the right hand side is equal to $\|\Psi\|$, so that $\|D[\Psi]\| \leq \|\Psi\|$. This holds for every vector $\Psi \in \oplus_A \ell^2$, implying that D is a contraction, as claimed. $\square$

This result motivates the definition of a category of linear shift invariant maps that are *guaranteed* to be contractions.

Definition 3.4.18. Let $\mathbf{LSI}_{\leq}$ be the subcategory of $\mathbf{LSI}$ obtained by imposing that morphisms $f: A \rightarrow B$

$$f = \{\widehat{f}_{\omega}: \mathbb{C}^{|A|} \rightarrow \mathbb{C}^{|B|}\}_{\omega \in \mathbb{R}}$$

must satisfy that each $\widehat{f}_{\omega}$ is a morphism in $\mathbf{FdContraction}(\mathbb{C}^{|A|}, \mathbb{C}^{|B|})$.

Remark 3.4.19. The hom-set $\mathbf{LSI}_{\leq}(A, B)$ may not contain every $\mathbf{LSI}$ contraction $\oplus_A \ell^2 \rightarrow \oplus_B \ell^2$ since the condition that each linear map $\widehat{D}_{\omega}$ is a contraction has only been established to be sufficient (Theorem 3.4.17). Nevertheless, it is reasonable to expect this condition is also necessary: if for some $\omega_0 \in (0, 2\pi)$ there is a vector $v \in \mathbb{C}^{|A|}$ such that $\|\widehat{D}_{\omega_0}(v)\| > \|v\|$, then we may engineer a state $\Psi \in \oplus_A \ell^2$ that, for each $\omega \in \mathbb{R}$, is characterised by:

$$\widehat{\Psi}_{\omega} = v \cdot \delta_{\bullet}(\omega - \omega_0)$$

where $\delta_{\bullet}: \mathbb{R} \rightarrow \mathbb{C}$ approximates the Dirac delta (see Remark 3.4.6). Then, using Lemma 3.4.16 we find that $\|\Psi\|^2 \approx \frac{1}{2\pi}\|v\|^2$ and $\|D[\Psi]\|^2 \approx \frac{1}{2\pi}\|\widehat{D}_{\omega_0}(v)\|^2$ so that we may expect $\|D[\Psi]\| > \|\Psi\|$, preventing D from being a contraction. Thus, it is likely that the condition established in Theorem 3.4.17 is not only sufficient, but also necessary.

Importantly, morphisms in $\mathbf{LSI}_{\leq}$ can be seen as $\mathbb{R}$ -indexed collections of morphisms from $\mathbf{FdContraction}$. Then, the execution formula from $\mathbf{FdContraction}$ induces an execution formula in $\mathbf{LSI}_{\leq}$. To this end, the following proposition lets us discuss addition of $\mathbf{LSI}$ maps in terms of the index-wise addition of its component linear maps.

Proposition 3.4.20. *For any two morphisms $f, g \in \mathbf{LSI}(A, B)$, let their addition $f + g$ be the following $\mathbb{R}$ -indexed collection:*

$$f + g = \{\widehat{f}_{\omega} + \widehat{g}_{\omega}\}_{\omega \in \mathbb{R}}.$$

Then, $f + g \in \mathbf{LSI}(A, B)$.

Proof. First, realise that if we see f and g as linear shift invariant maps $\oplus_A \ell^2 \rightarrow \oplus_B \ell^2$, their pointwise addition is again a linear shift invariant map — it will commute with shift operators. The challenge is to prove that such a map is characterised by the collection $\{\widehat{f}_{\omega} + \widehat{g}_{\omega}\}_{\omega \in \mathbb{R}}$ or, more explicitly, we must show that:

$$\widehat{(f + g)}_{\omega} = \widehat{f}_{\omega} + \widehat{g}_{\omega} \tag{3.37}$$

for all $\omega \in \mathbb{R}$. Recall that for any LSI map $h: \oplus_A \ell^2 \rightarrow \oplus_B \ell^2$ each $\widehat{h}_\omega: \mathbb{C}^{|A|} \rightarrow \mathbb{C}^{|B|}$ is the linear map given by the matrix:

$$\widehat{h}_\omega = \begin{pmatrix} \mathcal{F}_d[\chi_{ba}^h](\omega) & \dots & \mathcal{F}_d[\chi_{ba'}^h](\omega) \\ \vdots & \ddots & \vdots \\ \mathcal{F}_d[\chi_{b'a}^h](\omega) & \dots & \mathcal{F}_d[\chi_{b'a'}^h](\omega) \end{pmatrix}$$

where $\chi_{ba}^h \in \ell^2$ is the unique function satisfying $(\pi_b h \iota_a)[\psi] = \chi_{ba}^h * \psi$ for all $\psi \in \ell^2$. Considering that addition of linear maps is defined pointwise and that π_b is linear:

$$(\pi_b(f + g)\iota_a)[\psi] = (\pi_b f \iota_a)[\psi] + (\pi_b g \iota_a)[\psi]$$

for all $\psi \in \ell^2$. Then, thanks to convolution distributing over addition (see Proposition 3.4.3) we have that:

$$(\pi_b(f + g)\iota_a)[\psi] = (\chi_{ba}^f * \psi) + (\chi_{ba}^g * \psi) = (\chi_{ba}^f + \chi_{ba}^g) * \psi$$

and, due to uniqueness of the characteristic function, we conclude that $\chi_{ba}^{f+g} = \chi_{ba}^f + \chi_{ba}^g$. This, along with linearity of $\mathcal{F}_d$, implies that $\mathcal{F}_d[\chi_{ba}^{f+g}] = \mathcal{F}_d[\chi_{ba}^f] + \mathcal{F}_d[\chi_{ba}^g]$ and it is then immediate that $\widehat{(f + g)}_\omega = \widehat{f}_\omega + \widehat{g}_\omega$ as required by equation (3.37), completing the proof. $\square$

Evidently, it is generally not the case that for two morphisms $f, g \in \mathbf{LSI}_{\leq}(A, B)$ their addition $f + g$ is again in $\mathbf{LSI}_{\leq}$, since addition of contractions need not result in a contraction. In the following subsection we prove that $\mathbf{LSI}_{\leq}$ is a totally traced category with the execution formula.

3.4.3 The execution formula in $\mathbf{LSI}_{\leq}$

Since morphisms in $\mathbf{LSI}$ are $\mathbb{R}$ -indexed collections of morphisms in $\mathbf{FdHilb}$ and composition and addition in $\mathbf{LSI}$ are defined index-wise, it is not surprising that $\mathbf{LSI}$ is a Σ_g -UDC. Similarly, $\mathbf{LSI}_{\leq}$ is a Σ_w -UDC.

Proposition 3.4.21. *$\mathbf{LSI}$ is a Σ_g -UDC.*

Proof. The Σ -group structure on every hom-set $\mathbf{LSI}(A, B)$ is defined in terms of the Σ -group structure on $\mathbf{FdHilb}(\mathbb{C}^{|A|}, \mathbb{C}^{|B|})$, i.e. for each family $\{f_i\}_I \in \mathbf{LSI}(A, B)^*$ define:

$$\Sigma\{f_i\}_I = \begin{cases} \{\widehat{g}_\omega\}_{\omega \in \mathbb{R}} & \text{if } \forall \omega \in \mathbb{R}, \Sigma\{\widehat{(f_i)}_\omega\}_I \simeq \widehat{g}_\omega \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Since $\mathbf{FdHilb}(\mathbb{C}^{|A|}, \mathbb{C}^{|B|})$ is a Σ -group and Σ on $\mathbf{LSI}(A, B)$ is defined index-wise, it is immediate that $\mathbf{LSI}(A, B)$ is a Σ -group. Composition in $\mathbf{LSI}$ is Σ -bilinear since both composition and addition in $\mathbf{LSI}$ are defined index-wise and composition in $\mathbf{FdHilb}$ is Σ -bilinear; thus, $\mathbf{LSI}$ is a Σ_g -enriched category. Moreover, $(\mathbf{LSI}, \oplus, \emptyset)$ is a symmetric monoidal category where $\oplus$ corresponds to disjoint union of sets on objects and, on any two morphisms $f \in \mathbf{LSI}(A, B)$ and $g \in \mathbf{LSI}(C, D)$, it acts as index-wise direct sum:

$$f \oplus g = \{\widehat{f}_\omega \oplus \widehat{g}_\omega : \mathbb{C}^{|A|} \oplus \mathbb{C}^{|C|} \rightarrow \mathbb{C}^{|B|} \oplus \mathbb{C}^{|D|}\}_{\omega \in \mathbb{R}}.$$

This is well-defined thanks to the isomorphism $\mathbb{C}^{|A \uplus B|} \cong \mathbb{C}^{|A|} \oplus \mathbb{C}^{|B|}$. The monoidal unit $\emptyset$ is clearly a zero object since $\mathbb{C}^{|\emptyset|} = \{0\}$, so that $f \in \mathbf{LSI}(\emptyset, A)$ and $g \in \mathbf{LSI}(A, \emptyset)$ are $\mathbb{R}$ -indexed collections of copies of the zero map in $\mathbf{FdHilb}$. Verifying the identity $\Sigma\{\text{id}_A \oplus 0, 0 \oplus \text{id}_B\} \simeq \text{id}_{A \uplus B}$ is conceptually simple since both $\oplus$ and Σ in $\mathbf{LSI}$ are defined index-wise on morphisms, and the identity is satisfied in $\mathbf{FdHilb}$. Consequently, $\mathbf{LSI}$ is a Σ_g -UDC, as claimed. $\square$

Corollary 3.4.22. $\mathbf{LSI}_{\leq}$ is a Σ_w -UDC.

Proof. By definition, $\mathbf{LSI}_{\leq}$ is a subcategory of $\mathbf{LSI}$ and we may define a symmetric monoidal structure on it in the same manner we did for $\mathbf{LSI}$ in the previous proposition. Moreover, we may also provide a Σ -monoid structure on each hom-set in $\mathbf{LSI}_{\leq}$ via the same index-wise definition — this time using the Σ -monoid structure of hom-sets in $\mathbf{FdContraction}$. Recall that $\mathbf{FdContraction}$ is only Σ_w -enriched, since a pair of contractions does not generally add up to a contraction; thus, the enrichment in $\mathbf{LSI}_{\leq}$ is over Σ_w as well. Consequently, the same proof strategy from the previous proposition establishes that $\mathbf{LSI}_{\leq}$ is a Σ_w -UDC. $\square$

It follows from the fact that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a totally traced category that $(\mathbf{LSI}_{\leq}, \oplus, \text{ex})$ is totally traced as well.

Theorem 3.4.23. *The category $(\mathbf{LSI}_{\leq}, \oplus, \text{ex})$ is totally traced.*

Proof. Recall that on any Σ_* -UDC the execution formula ex is defined as follows:

$$\text{ex}^U(f) = \Sigma\{f_{BA}\} \uplus \{f_{BU}f_{UU}^n f_{UA}\}_{\mathbb{N}}.$$

Since the morphisms in $\mathbf{LSI}_{\leq}$ are, by definition, $\mathbb{R}$ -indexed collections of morphisms in $\mathbf{FdContraction}$ and $\Sigma, \oplus$ and composition in $\mathbf{LSI}_{\leq}$ are all defined index-wise, we have that $\text{ex}^U(f)$ corresponds to:

$$\text{ex}^U(f) = \{\text{ex}^{\mathbb{C}^{[U]}}(\widehat{f}_\omega)\}_{\omega \in \mathbb{R}}$$

where $\text{ex}^{\mathbb{C}^{|U|}}$ is the execution formula in **FdContraction**. Theorem 3.3.15 establishes that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a totally traced category and, hence, for all $\omega \in \mathbb{R}$ the linear map $\text{ex}^{\mathbb{C}^{|U|}}(\widehat{f}_\omega)$ is a well-defined contraction. According to Theorem 3.4.17 this guarantees that the LSI map $\text{ex}^U(f)$ is itself a contraction and, hence, the execution formula is a total function in $\mathbf{LSI}_\leq$. Moreover, $(\mathbf{LSI}_\leq, \oplus, \text{ex})$ is a traced monoidal category since the execution formula in **FdContraction** is a valid trace, which guarantees that the axioms of traced monoidal categories are satisfied for every index $\omega \in \mathbb{R}$. Thus, we conclude that $(\mathbf{LSI}_\leq, \oplus, \text{ex})$ is a totally traced category, as claimed. $\square$

As discussed at the beginning of this section, the physical interpretation of the Hilbert space $\oplus_A \ell^2$ is that each vector is a quantum state that spreads over a finite set of locations A and over the discrete-time domain $\mathbb{Z}$. LSI maps $\oplus_A \ell^2 \rightarrow \oplus_B \ell^2$ are interpreted as devices with a finite number of input and output ports, labelled by the elements in A and B respectively, and whose input-output behaviour is independent of the time-step the input reaches the device. Then, $\text{ex}^U(f)$ may be interpreted as the device obtained after connecting the output ports labelled by U to the input ports with the same label.

Considering the prevalence of the notion of time — an infinite domain — in the definition of **LSI**, it is perhaps surprising that we may prove $\mathbf{LSI}_\leq$ is a traced category by means of an argument in finite dimensions, *i.e.* using **FdContraction**. The key to this result were the discrete-time Fourier transform (Definition 3.4.7), together with its convolution theorem (Theorem 3.4.9) and the unique characterisation of LSI maps (Proposition 3.4.13). In short, the Fourier transform lets us uniquely characterise time-shift invariant maps according to their action on plane waves $\gamma_\omega(t) = e^{i\omega t}$ (see Remark 3.4.8) and the convolution theorem establishes that the action of these maps at different frequencies ω do not interact with each other. Consequently, in order to describe the input-output behaviour of a time-shift invariant device, we simply need to figure out how it affects the phase and amplitude of each plane wave γ_ω for every angular frequency $\omega \in \mathbb{R}$; this is precisely what each map $\widehat{f}_\omega: \mathbb{C}^{|A|} \rightarrow \mathbb{C}^{|B|}$ represents. In some sense, the fact that $(\mathbf{LSI}_\leq, \oplus, \text{ex})$ is a traced category tells us that we may calculate the behaviour of a device with feedback by an iterative method, adding up the contribution of each of the possible paths from ports in A to ports in B . Moreover, the fact that ex is total indicates that this iterative method always converges. These results are not novel: physicists and engineers have been using these facts about LSI maps and

the Fourier transform for quite a long time; however, there is some value in providing a formal description of this situation in the language of traced monoidal categories, as it may enable computer scientists to use these notions in a more abstract setting.

The category **LSI** captures the notion of quantum processes over a discrete domain $\mathbb{Z}$; such domain has been interpreted to describe the line of time, but it could instead be interpreted to correspond to discrete one-dimensional space, or we may extend it to a domain $\mathbb{Z}^3$ so that the appropriate version of **LSI** would capture maps that are invariant with respect to translations on three-dimensional space. A natural question is whether a similar approach could be used to discuss LSI maps acting on states over a *continuous* domain $\mathbb{R}$. To do so, we would need to work with the Hilbert space L^2 but, as discussed in Remark 3.4.6, convolution in L^2 does not have a unit and, consequently, the characterisation of LSI via an $\mathbb{R}$ -indexed collection of linear maps is only valid up to approximation (see Proposition 3.4.12). In practice, physicists and engineers tend to disregard these approximations and treat them as equalities (see Remark 3.4.6); however, the objective of this section was to discuss feedback in the formal setting of traced monoidal categories and, to that extent, the case of the continuous-time domain is left as an open question.

3.5 Related work

In this section, the contributions of this chapter are put into perspective and compared to previous works in the field.

Hoshino [33]. Hoshino proved that every Σ_s -UDC is partially traced using the execution formula. The same result is established in Theorem 3.2.11 of this chapter, although the proof strategy differs from Hoshino's. In [33], Hoshino proves that every Σ_s -UDC can be faithfully embedded in a category with countable biproducts and enriched over *totally* defined strong Σ -monoids — *i.e.* every family of morphisms is summable. It is straightforward to show that these categories are totally traced using the execution formula and, thus, the faithful embedding provided by Hoshino immediately induces a partial trace on the original category as described in Proposition 3.1.3. Unfortunately, such a result cannot be reproduced in the general case of Σ_w -UDCs: there are examples of such categories that are not partially traced using the execution formula, *e.g.* $(\mathbf{FdHilb}, \oplus, \{0\})$, see Example 3.2.12. Our interest in Σ_w -enriched categories comes from the fact that the categories of quantum processes we are interested

in (e.g. **Unitary** and **LSI**) have additive inverses of morphisms; these capture destructive interference in quantum computing but are not available in a Σ_s -enrichment.

The Barr functor. The Barr functor $\mathbf{PInj} \rightarrow \mathbf{Contraction}$ originally proposed in [9] induces a Σ_s -enrichment in **Contraction** using that of **PInj**. The details of the induced Σ_s -enrichment in **Contraction** appear in Section 5.5 from [28]; in essence, families of contractions are deemed summable if and only if the domain and codomain of every pair of contractions come from orthogonal subspaces and, hence, no quantum interference may occur between them. Such a situation can be perceived as a quantum computer simulating a classical iterative loop, which is not of interest to this thesis.

Malherbe, Scott & Selinger [44]. The kernel-image trace plays an essential role in our proof that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a totally traced category. We can draw a loose analogy between the kernel-image trace of a morphism $f: A \oplus U \rightarrow B \oplus U$ and the existence of a pseudo-inverse $(\text{id} - f_{UU})^+$ which may be used to calculate $\text{Tr}_{\text{ki}} \approx f_{BA} + f_{BU}(\text{id} - f_{UU})^+ f_{UA}$. Under this analogy, the intuition behind our proof that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is totally traced loosely corresponds to the identity in the real numbers $(1 - r)^{-1} = \sum_{n=0}^{\infty} r^n$, which only holds when both sides are defined and $r < 1$. More formally, Lemma 3.2.30 identifies sufficient conditions for a morphism f to satisfy that the well-definedness of $\text{Tr}_{\text{ki}}(f)$ implies that of $\text{ex}(f)$, with their results matching. Then, we take advantage of the kernel-image trace being a valid categorical trace in **Hilb** to prove that the execution formula is a valid trace of its subcategory **FdContraction**.

Bartha [10]. Bartha had shown that both **FdIsometry** and **FdUnitary** are totally traced categories using the execution formula. Bartha's proof makes use of the kernel-image trace in a similar way as we do and, in particular, our proof that **FdContraction** is totally traced (instead of simply partially traced) relies on the approach Bartha used to prove the same result in the case of **FdIsometry** (see Proposition 3.3.8). Even though Bartha did not explicitly show that **FdContraction** is totally traced, this can be easily inferred from the result that **FdIsometry** is totally traced and, hence, we do not claim this to be a novel result. Instead, the approach we have used to reach this result is novel: whereas Bartha relies on matrix iteration theories, our approach relies on the more general theory of Hausdorff abelian groups. Moreover, our approach is self-contained, whereas Bartha defers the details of a key

Chapter 4

Weakly measured while loops

This chapter reproduces the contents of a publication due to Andres-Martinez and Heunen [5]; the text and notation has been revised to better fit this thesis. The focus of this chapter is on classical control flow of quantum programs and, hence, measurements are fundamental. As such, the framework of completely positive weak trace reducing maps (CPTR) will be used (see Section 1.5.1 for a brief introduction).

A while loop tests a termination condition on every iteration but, on a quantum computer, such measurements perturb the evolution of the algorithm. In this chapter we define a while loop primitive using weak measurements, offering a trade-off between the perturbation caused and the amount of information gained per iteration. This trade-off is adjusted with a parameter set by the programmer. We provide sufficient conditions that let us determine, with arbitrarily high probability, a worst-case estimate of the number of iterations the loop will run for. As an example, we solve Grover’s search problem using a while loop and prove the quadratic quantum speed-up is maintained.

This chapter is structured as follows: Section 4.1 gives a brief introduction to weak measurements and defines the notion of κ -measurement; Section 4.2 contains the main contribution of this chapter: the proposal of κ -while loops and the study of its properties; Section 4.3 shows that Grover’s algorithm may be implemented using a κ -while loop, and Section 4.4 concludes the chapter with discussion of related work and open questions.

4.1 Weak measurements (*Preamble*)

Roughly speaking, a weak measurement is “a measurement which gives very little information about the system on average, but also disturbs the state very little” [14]. For instance, the field of quantum feedback control uses weak measurements (often, continuous measurements) to monitor a state; the stream of measurement outcomes is used to control the strength of a Hamiltonian that corrects the system (see [58] for a survey). Our approach is inspired by these ideas, contextualised for their application to algorithm design. We restrict ourselves to the discrete-time setting and define a particular kind of parametrised measurement, the κ -measurement, that behaves as a weak measurement when κ is small.

Let H be a Hilbert space and let B be an orthonormal basis of H ; we wish to apply a measurement to test whether a state $\psi \in H$ satisfies a predicate $Q: B \rightarrow \{0, 1\}$. Assume the existence of a unitary $O_Q: H \otimes \mathbb{C}^2 \rightarrow H \otimes \mathbb{C}^2$ acting as the *oracle* of predicate Q :

$$O_Q|x, p\rangle = |x, p \oplus Q(x)\rangle.$$

Fix a value of parameter $\kappa \in [0, 1]$ and let $P = \text{span}\{\perp, \top\}$ be an auxiliary space known as the *probe*. We define a unitary $E_{\kappa, Q}$ as follows:

$$\begin{aligned} E_{\kappa, Q} &= (O_Q^\dagger \otimes \text{id}_P) (\text{id}_H \otimes \Lambda(R_\kappa)) (O_Q \otimes \text{id}_P) \\ \Lambda(R_\kappa) &= |0\rangle\langle 0| \otimes \text{id}_P + |1\rangle\langle 1| \otimes R_\kappa \\ R_\kappa &= \begin{pmatrix} \sqrt{1-\kappa} & \sqrt{\kappa} \\ \sqrt{\kappa} & -\sqrt{1-\kappa} \end{pmatrix}, \end{aligned}$$

The purpose of R_κ is to map $|\perp\rangle$ to $\alpha|\perp\rangle + \beta|\top\rangle$ so that $|\beta|^2 = \kappa$.¹ In particular, if $\kappa = 0$ then $R_0 = \text{id}$ and if $\kappa = 1$ then $R_1|\perp\rangle = |\top\rangle$. $E_{\kappa, Q}$ is a quantumly controlled version of R_κ where the outcome from the oracle O_Q acts as the control. Notice that the auxiliary qubit the oracle O_Q acts upon is restored to its initial state by $O_Q^\dagger$ so it may be reused and we need not keep track of it. We initialise it to $|0\rangle$ and omit it in further discussions.

Definition 4.1.1. A κ -measurement of predicate Q may be applied on any density matrix $\rho \in B(H)$ by the following procedure:

- apply the unitary $E_{\kappa, Q}: H \otimes P \rightarrow H \otimes P$ on the state $\rho \otimes |\perp\rangle\langle\perp|$,

¹Such a behaviour is not unique to this definition of R_κ . For instance, $Z(\theta)R_\kappa Z(\theta')$ for any θ and θ' is an equally valid choice for this unitary, where $Z(\theta)$ is a Z -rotation of angle θ .

- apply a measurement to discern between the orthogonal subspaces $H \otimes \text{span}\{\perp\}$ and $H \otimes \text{span}\{\top\}$.

Since $E_{\kappa,Q}$ entangles the probe with the result of the oracle, the outcome of this measurement provides some information about the original state in H . If the outcome of the κ -measurement is $\perp$ and $\kappa < 1$, we obtain no definitive information on whether the state in H satisfies Q or not: for any state $\rho \in B(H)$, the probability of outcome $\top$ is

$$p_{\top}(\rho) = \kappa \cdot p_Q(\rho)$$

where $p_Q(\rho)$ is the probability of predicate Q being satisfied by ρ ,

$$p_Q(\rho) = \text{tr} \left((\text{id}_H \otimes \langle 1|) O_Q (\rho \otimes |0\rangle\langle 0|) O_Q^\dagger (\text{id}_H \otimes |1\rangle) \right). \quad (4.1)$$

The smaller κ is, the less likely it is that we read outcome $\top$. On the other hand, when the outcome of the κ -measurement is $\top$, no matter the value of κ we are *certain* that the state left in space H satisfies predicate Q . In contrast, the map $\mathcal{W}_{\perp} : B(H) \rightarrow B(H)$ corresponding to outcome $\perp$ does *not* project the state to the subspace of H where Q is unsatisfiable:

$$\mathcal{W}_{\perp}(\rho) = \frac{W_{\perp} (\rho \otimes |\perp\rangle\langle\perp|) W_{\perp}^\dagger}{1 - p_{\top}(\rho)} \quad (4.2)$$

where $W_{\perp} = (\text{id}_H \otimes |\perp\rangle\langle\perp|) E_{\kappa,Q}$.

Intuitively, a κ -measurement models the behaviour of a device with internal state space P used to measure a property Q of a system H , with κ indicating how strongly the probing device couples with the system.

Remark 4.1.2. After applying a measurement on a pure state, we may describe the result as a mixed state on the C^* -algebra $B(H \otimes P)$. However, in doing so are omitting the information we have obtained from the classical outcome of the measurement: we in fact know whether the state is in $H \otimes \text{span}\{\perp\}$ or in $H \otimes \text{span}\{\top\}$. For our purposes, it is more elucidating to describe the two possible outcomes separately. Notice that when the state in H prior to measurement is a pure state both possible outcomes yield a pure state; this fact will be used in Section 4.3 to simplify our discussions.

4.2 The κ -while loop

This section contains the main contribution of this chapter: we define κ -while loops and introduce the properties of $\mathcal{A}$ -guarantee and robustness which let us estimate the

worst-case runtime of certain algorithms using κ -while loops. In short, a κ -while loop is a classically controlled while loop where the test of the termination condition is realised by a κ -measurement.

4.2.1 Motivation

Fix a predicate Q , and let $\mathcal{C}$ be a completely positive (weak) trace reducing (CPTR) map acting on $B(H)$; we will refer to $\mathcal{C}$ as the *body* of the loop.

Definition 4.2.1. Let $\sigma \in B(H)$ be an arbitrary density matrix. Define a predicate $\mathcal{A}_{Q,\sigma}: \mathbb{N} \rightarrow \mathbb{B}$ as follows:

$$\mathcal{A}_{Q,\sigma}(n) \iff p_Q(\mathcal{C}^n(\sigma)) > \frac{1}{2}$$

where p_Q is given in (4.1). If $\mathcal{A}_{Q,\sigma}(n)$ is satisfied, we say n is an $\mathcal{A}$ -iteration — read as *active iteration*.

If we can somehow identify an $\mathcal{A}$ -iteration m , we may use a simple approach to find a state satisfying Q : apply $\mathcal{C}^m(\sigma)$ and then perform a projective measurement; if the outcome does not satisfy Q , repeat the process from the initial state σ . Thanks to $\mathcal{A}_{Q,\sigma}(m)$ being satisfied, the probability of succeeding at some point within k restarts is $1 - 1/2^k$ so the probability of success quickly approaches 1 as k increases. This is an efficient approach whenever a small $\mathcal{A}$ -iteration m is known. In fact, this is how the standard Grover's algorithm works, choosing an iteration m where p_Q is maximised.

However, this measure-restart strategy can only be implemented if we already know when $\mathcal{A}$ -iterations will occur. The distribution of $\mathcal{A}$ -iterations may become unpredictable as soon as some randomness is introduced in the body of the loop. For instance, consider the standard Grover's algorithm being implemented on a faulty machine where, without notice, the quantum memory may reset to its initial state. If such an event occurs mid-computation the evolution is effectively restarted, but the algorithm — oblivious to the memory reset — continues for only the remaining fixed number of iterations. In contrast, a version of Grover's algorithm that uses a κ -while loop (such as the one we describe in Section 4.3) will, by definition, keep iterating until it succeeds to find the target state. Although a contrived example, this illustrates how a while loop could provide reliability against unpredictable behaviour. More realistic scenarios that would lead to an unpredictable distribution of $\mathcal{A}$ -iterations may come from algorithms that, by design, explore the state space in an unpredictable manner;

<pre> while $!_{\kappa} Q[\rho]$ do $\rho \leftarrow \mathcal{C}(\rho)$ end </pre>	<pre> $q \leftarrow \perp\rangle\langle\perp$ while $M[q] = \perp$ do $\rho \leftarrow \mathcal{C}(\rho)$ $\rho, q \leftarrow E_{\kappa, Q}(\rho \otimes q)$ end </pre>
---	--

Figure 4.1: *Left*: the syntax we use to represent a κ -while loop; $\kappa \in [0, 1]$ is a parameter set by the programmer and Q is the predicate to be measured. *Right*: the pseudocode that implements the κ -while loop on a programming language with classical control flow, following the notation from [54]; $E_{\kappa, Q}$ is defined in Section 4.1 and the condition $M[q] = \perp$ indicates that q is measured to distinguish between $H \otimes \text{span}\{\perp\}$ and $H \otimes \text{span}\{\top\}$, with the condition being satisfied if q collapses to the former.

Section 4.4.2 proposes the field of quantum walk-based algorithms as an area where such examples may perhaps be found.

In this section we present the concept of κ -while loops as an abstract quantum programming construct. In Section 4.3 we present a version of Grover’s algorithm that uses a κ -while loop and maintains the quadratic quantum speed-up. In the future, we hope to apply κ -while loops to practical problems where the distribution of $\mathcal{A}$ -iterations cannot be predicted. To this end, we provide sufficient conditions that let us determine, with arbitrarily high probability, a worst-case estimate of the number of iterations the loop will run for. Remarkably, these conditions do not require us to know the precise distribution of $\mathcal{A}$ -iterations, but only a guarantee of their proportion throughout the algorithm in the worst-case scenario.

4.2.2 Definition and properties

In Figure 4.1 we propose the syntax for the κ -while loop and its implementation in a quantum programming language with classical control flow. The syntax of the κ -while loop is meant to be read as “repeat $\mathcal{C}$ while it is *not certain* that Q is satisfied”. On each iteration, the value of the state ρ will be updated to $\mathcal{C}(\rho)$, followed by a κ -measurement of predicate Q (see Section 4.1). If the outcome of the κ -measurement is $\perp$, the loop keeps iterating; the state ρ becomes $\mathcal{W}_{\perp}(\rho)$ where $\mathcal{W}_{\perp}$ is given in (4.2). Otherwise, outcome $\top$ causes the loop to halt and we succeed in obtaining a state ρ that satisfies Q .

Definition 4.2.2. An isotone function $f: \mathbb{N} \rightarrow \mathbb{N}$ is a $\mathcal{A}_{Q,\sigma}$ -guarantee if for all $n \in \mathbb{N}$:

$$n \leq |\{k \in \mathbb{N} \mid k \leq f(n), \mathcal{A}_{Q,\sigma}(k)\}| \quad (4.3)$$

If such a function f exists we are promised that there will be at least n active iterations within the first $f(n)$ applications of $\mathcal{C}$. In principle, this need not be a tight bound. For instance, we may only know that within the first thousand iterations there are at least ten $\mathcal{A}$ -iterations; a valid $\mathcal{A}_{Q,\sigma}$ -guarantee in this case would be a function f such that $f(1) = f(2) = \dots = f(10) = 1000$. This gives us little information about when any of these $\mathcal{A}$ -iterations actually occur.

Definition 4.2.3. The evolution induced by a CPTR map $\mathcal{C}$ on a state σ is said to be ε -robust to κ -measurements of predicate Q if there is an isotone function $g: \mathbb{N} \rightarrow \mathbb{N}$ such that:

$$\forall n \in \mathbb{N}, \exists m \leq g(n): |p_Q(\mathcal{C}^n(\sigma)) - p_Q((\mathcal{W}_\perp \mathcal{C})^m(\sigma))| \leq \varepsilon \quad (4.4)$$

where $\varepsilon < 1/2$ and $\mathcal{W}_\perp$ is given in (4.2). The function g is said to be a *witness* of the robustness.

In practice, if the evolution is ε -robust this means that the weakly-measured evolution will provide a similar curve of the success probability p_Q throughout its iterations, but certain intervals may span a larger number of iterations. Then, the witness function g provides a bound on how much slower the weakly-measured evolution may be.

Lemma 4.2.4. If f is a $\mathcal{A}_{Q,\sigma}$ -guarantee and g witnesses that $\mathcal{C}$ is ε -robust to κ -measurements of predicate Q , then $g \circ f$ is a $\mathcal{A}'_{Q,\sigma}$ -guarantee where $\mathcal{A}'_{Q,\sigma}$ is the predicate satisfying:

$$\mathcal{A}'_{Q,\sigma}(n) \iff p_Q((\mathcal{W}_\perp \mathcal{C})^n(\sigma)) > \frac{1}{2} - \varepsilon$$

Proof. Since f is a $\mathcal{A}_{Q,\sigma}$ -guarantee, at least n instances of $\mathcal{A}_{Q,\sigma}$ -iterations will occur within the first $f(n)$ applications of $\mathcal{C}$. If we switch to weakly measured iterations $\mathcal{W}_\perp \mathcal{C}$, robustness tells us that within the first $gf(n)$ iterations of the loop we will find those n active iterations again, but the probability p_Q might differ by ε , so it is at least $p_Q > 1/2 - \varepsilon$. Thus, for every $n \in \mathbb{N}$ it follows that:

$$n \leq |\{k \in \mathbb{N} \mid k \leq gf(n), \mathcal{A}'_{Q,\sigma}(k)\}|$$

This concludes the proof. □

Lemma 4.2.4 guarantees that at least n of the first $gf(n)$ iterations of the κ -while loop will be $\mathcal{A}'$ -iterations. For each of these $\mathcal{A}'$ -iterations, the probability of outcome $\top$ is at least $\kappa(1/2 - \varepsilon)$ since $p_\top = \kappa \cdot p_Q$. Within N instances of $\mathcal{A}'$ -iterations the probability of loop termination is:

$$P_{succ} > 1 - (1 - \kappa(1/2 - \varepsilon))^N$$

Since for all $x \in (0, 1)$ we have that $(1 - x)^{\frac{1}{x}} < \frac{1}{e}$, if the algorithm is allowed to run for at least $N = \frac{2}{\kappa(1-2\varepsilon)} \mathcal{A}'$ -iterations, the probability of success will be:

$$P_{succ} > 1 - \frac{1}{e} > \frac{1}{2}$$

Therefore, with probability higher than $1/2$, our κ -while loop will halt within its first

$$T = gf\left(\frac{2}{\kappa(1-2\varepsilon)}\right) \quad (4.5)$$

iterations. Thus, equation (4.5) estimates the median of the number of iterations the κ -while loop runs for before halting. More generally, for any $c \in \mathbb{N}$, the probability of success after cN instances of $\mathcal{A}'$ -iterations is greater than $1 - 1/e^c$. Since $1/e^c$ quickly approaches 0 as c increases, we may say that, with arbitrarily high probability, the loop halts successfully within $T_c = gf(cN)$ iterations for small c . Hence, we may claim that the time complexity of the loop is asymptotically bounded by $gf(cN)$ or, in other words, its time complexity is

$$\mathcal{O}\left(gf\left(\frac{2c}{\kappa(1-2\varepsilon)}\right)\right).$$

Hence, if we manage to choose appropriate κ , ε , f and g , we can prove statements about quantum speed-ups — under the assumption that the time complexity of the best classical algorithm is known. Notice that if we do not manage to find witness functions f and g to prove certain quantum speed-up this does not rule out its existence.

Remark 4.2.5. Definition 4.2.2 can be weakened so that instead of f being a ‘deterministic’ guarantee of $\mathcal{A}_{Q,\sigma}$, we only impose that $f(n)$ satisfies (4.3) with probability higher than η . In such a case, after T_c iterations we can achieve a success probability arbitrarily close to η .

4.3 An example: Grover's algorithm

In Grover's search problem [26], we are given an unsorted set of elements B , about which we know no structure or heuristics, and a predicate $\chi: B \rightarrow \{0, 1\}$ that satisfies

```

input :  $\chi, |\psi\rangle$ 
output :  $q$ 
begin
   $q \leftarrow |\psi\rangle$ 
  while  $!_{\kappa} \chi[q]$  do
     $q \leftarrow G(q)$ 
  end
end

```

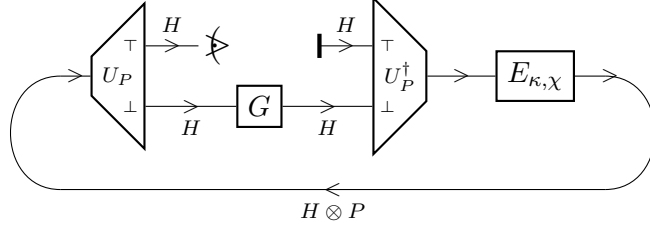


Figure 4.2: *Left*: pseudocode describing our algorithm. The while loop is controlled by a κ -measurement as described in Section 4.2.2. *Right*: the algorithm's information flow; $E_{\kappa, \chi}$ is given in Section 4.1, U_P applies the canonical isomorphism $H \otimes P \cong H \oplus H$, separating with respect to the orthogonal basis $\{|\perp\rangle, |\top\rangle\}$ of P .

$\chi(\star) = 1$ for only one element $\star \in B$. We are tasked with finding this marked element $\star$. The standard Grover's algorithm defines an iteration operator G using an oracle of χ and applies it a fixed number of times $K \approx \sqrt{|B|}$ on an initial state $|\psi\rangle$. Afterwards, a measurement on the basis B is applied, finding the marked element with high probability. Thus, the algorithm achieves a quadratic quantum speed-up: it requires $\mathcal{O}(\sqrt{|B|})$ iterations whereas a classical algorithm would require $\mathcal{O}(|B|)$ iterations since there is no available heuristic to guide the search. In this section we discuss a different approach to Grover's search problem where a κ -while loop is used instead.

The algorithm's pseudocode is given in Figure 4.2. When the κ -while loop halts, we are *certain* that the state is $|\star\rangle$. We do not fix the number of times G is applied; instead, we fix the measurement strength κ . We will see that for $\kappa \approx |B|^{-1/2}$ the loop terminates within $c\sqrt{|B|}$ iterations with arbitrarily high probability, where c is a small constant. Thus, the quadratic quantum speed-up of Grover's algorithm is preserved.

It is important to remark that, in the case of Grover's iterator G , we can easily predict when the $\mathcal{A}$ -iterations occur (see Section 4.2.1). Therefore, κ -while loops do not provide an algorithmic advantage on this problem. Instead, we present Grover's search problem as a simple proof of concept showing how classically controlled while loops may be to monitor a quantum algorithm without destroying the quantum speed-up. This approach to Grover's problem was first proposed by Mizel [46]. This section reformulates Mizel's results in the broader framework of Section 4.2.2.

4.3.1 Standard Grover's algorithm (*Preamble*)

We first give a brief description of the standard Grover's algorithm and introduce our notation. The standard Grover's algorithm acts on a quantum state in $H = \text{span } B$, starting from the uniform superposition state:

$$|\psi\rangle = \frac{1}{\sqrt{|B|}} \sum_{b \in B} |b\rangle. \quad (4.6)$$

Let $|\psi_1\rangle = |\star\rangle$ be the target state and

$$|\psi_0\rangle = \frac{1}{\sqrt{|B|-1}} \left(|\psi\rangle - \sqrt{|B|} \cdot |\psi_1\rangle \right)$$

For every angle $a \in [0, 2\pi)$ define the state $|a\rangle$ as follows:

$$|a\rangle = \cos a |\psi_0\rangle + \sin a |\psi_1\rangle. \quad (4.7)$$

Let $\alpha = \arcsin \langle \star | \psi \rangle = \arcsin |B|^{-1/2}$; it is customary to assume that $|B|$ is large enough so that the approximation $\alpha \approx |B|^{-1/2}$ is reasonable. Notice that $|\alpha\rangle = |\psi\rangle$.

For every state $|\varphi\rangle \in H$ let:

$$S_\varphi = 2|\varphi\rangle\langle\varphi| - \text{id}_H$$

Grover's iteration is given by the operator $G = S_\psi S_{\psi_0}$ where S_{ψ_0} may be implemented using an oracle of χ .

On a state $|a\rangle$, the action of Grover's iteration G is:

$$G|a\rangle = S_\psi S_{\psi_0}|a\rangle = S_\psi|-a\rangle = |a + 2\alpha\rangle \quad (4.8)$$

and after k iterations:

$$G^k|\psi\rangle = \cos(\alpha + 2k\alpha)|\psi_0\rangle + \sin(\alpha + 2k\alpha)|\psi_1\rangle$$

The standard Grover's algorithm applies G a total of $K = \lfloor (\pi\sqrt{|B|})/4 \rfloor$ times on $|\psi\rangle$ so that the amplitude of $|\psi_1\rangle$ is maximised. Then, the state is measured on the basis B , finding the marked element $\star$ with high probability.

4.3.2 While loop approach

In this section we discuss the algorithm given in Figure 4.2. To show that it retains the quadratic quantum speed-up, we provide an $\mathcal{A}$ -guarantee and prove that G is robust to κ -measurements of χ , as described in Section 4.2.2. Our framework is general enough

to deal with mixed states and CPTR maps but, in the case of Grover's algorithm, the body of the κ -while loop G is unitary and both the initial state $|\psi\rangle$ and the target $|\star\rangle$ are pure states. Therefore, as discussed in Remark 4.1.2, all states involved in our analysis will be pure states.

Lemma 4.3.1. *The function $f: \mathbb{N} \rightarrow \mathbb{N}$ given by $f(n) = 2n + K$ where $K = \lfloor \frac{\pi\sqrt{|B|}}{4} \rfloor$ is a $\mathcal{A}_{\chi,|\psi\rangle}$ -guarantee.*

Proof. By definition (4.7) of $|a\rangle$,

$$p_\chi(|a\rangle) = |\langle \star | a \rangle|^2 = \sin^2 a.$$

Whenever $a \in (\frac{\pi}{4}, \frac{3\pi}{4})$ or $a \in (\frac{5\pi}{4}, \frac{7\pi}{4})$, we have $p_\chi(|a\rangle) > \frac{1}{2}$. Moreover, $G|a\rangle = |a + 2\alpha\rangle$ according to (4.8), so the angle is increased a constant amount on each iteration. Thus, within $m \in \mathbb{N}$ iterations we know that the number of $\mathcal{A}$ -iterations is at least $\frac{m-K}{2}$ where K (defined in the claim) is the number of applications of G it takes to traverse a $\frac{\pi}{2}$ angle, which is the longest interval of consecutive iterations not satisfying $\mathcal{A}_{\chi,|\psi\rangle}$. Assuming the worst case scenario we find that within $m = 2n + K$ iterations it is guaranteed that n of them are $\mathcal{A}_{\chi,|\psi\rangle}$ -iterations, i.e. for all $n \in \mathbb{N}$

$$n \leq |\{k \in \mathbb{N} \mid k \leq f(n), \mathcal{A}_{\chi,|\psi\rangle}(k)\}|$$

thus, proving the claim. $\square$

Consider applying a κ -measurement at iteration n : if the outcome is $\top$, the state becomes $|\star\rangle$, otherwise it suffers a collapse towards $|\psi_0\rangle$ determined by the projector $W_\perp$ from Section 4.1:

$$\begin{aligned} W_\perp |a, \perp\rangle &= \cos a W_\perp |\psi_0, \perp\rangle + \sin a W_\perp |\psi_1, \perp\rangle \\ &= \cos a |\psi_0, \perp\rangle + \sin a (\text{id}_H \otimes |\perp\rangle\langle\perp|) E_{\kappa, \chi} |\psi_1, \perp\rangle \\ &= \cos a |\psi_0, \perp\rangle + \sin a \sqrt{1 - \kappa} |\psi_1, \perp\rangle \end{aligned} \quad (4.9)$$

Thus, the output is still in a superposition between $|\psi_0\rangle$ and $|\psi_1\rangle$. Notice that $W_\perp$ returns an unnormalised state; the actual CPTP map $\mathcal{W}_\perp$ from (4.2) takes care of the normalization. Then, we may describe the action of $\mathcal{W}_\perp$ as $|a\rangle \xrightarrow{\mathcal{W}_\perp} |a - \theta(a, \kappa)\rangle$ on every angle $a \in [0, 2\pi)$ where the value of $\theta(a, \kappa)$ may be calculated from (4.9) following the geometric construction of Figure 4.3. We now find an upper bound of $\theta(a, \kappa)$; this will be used to prove robustness of G . To reduce clutter, we use the shorthand:

$$\xi = \sqrt{1 - \kappa} \quad (4.10)$$

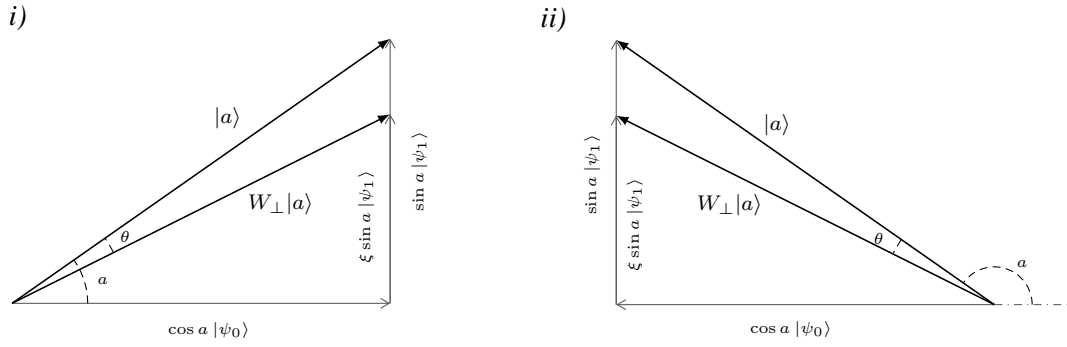


Figure 4.3: Geometric relation between the angle a before weak measurement and the offset $\theta(a, \kappa)$ after measuring outcome $\perp$. The construction is provided when *i*) $a \in [0, \frac{\pi}{2})$ and when *ii*) $a \in [\frac{\pi}{2}, \pi)$.

Lemma 4.3.2. *For any κ and angle a*

$$|\theta(a, \kappa)| \leq \arcsin\left(\frac{1 - \xi}{1 + \xi}\right).$$

Proof. Suppose $a \in [0, \frac{\pi}{2})$ and use the properties of sines on the triangle from Figure 4.3i to obtain:

$$\frac{\sin(a - \theta)}{\xi \sin a} = \frac{\sin(\pi/2 - a + \theta)}{\cos a}$$

This simplifies as follows:

$$\begin{aligned} \frac{\sin(a - \theta)}{\xi \sin a} &= \frac{\cos(a - \theta)}{\cos a} \\ \iff \tan(a - \theta) &= \xi \tan a \\ \iff \frac{\tan a - \tan \theta}{1 + \tan a \tan \theta} &= \xi \tan a \end{aligned}$$

Solving for θ gives:

$$\theta(a, \kappa) = \arctan\left(\frac{(1 - \xi) \tan a}{1 + \xi \tan^2 a}\right) \quad (4.11)$$

If $a \in [\frac{\pi}{2}, \pi)$ instead, a similar analysis yields:

$$\theta(a, \kappa) = -\arctan\left(\frac{(1 - \xi) \tan a}{1 + \xi \tan^2 a}\right) \quad (4.12)$$

which only differs from equation (4.11) in the sign. If a is in the third quadrant, then we obtain equation (4.11) and, if a is in the fourth quadrant, we get (4.12). These sign changes are convenient: the geometric analysis in Figure 4.3 shows that in the first (and third) quadrant, the resulting angle is $a - |\theta|$, whereas in the second (and fourth)

quadrant it is $a + |\theta|$. Thus the resulting angle is $a - \theta$ regardless of which quadrant a lies in.

Next, we determine the maximum value of $\theta(a, \kappa)$. To do so, fix κ and find the critical points of θ as a function on a . These happen where the derivative

$$\frac{d\theta}{da} = 1 - \frac{\xi}{\cos^2 a + \xi^2 \sin^2 a}$$

vanishes. Critical points occur periodically (once in every quadrant), but all of them reach the same absolute value. The critical point within the first quadrant happens at:

$$a = \arccos \left(\sqrt{\frac{\xi}{\xi + 1}} \right)$$

When applied to equation (4.11) this identity gives a tight upper bound:

$$|\theta(a, \kappa)| \leq \arctan \left(\frac{1 - \xi}{2\sqrt{\xi}} \right)$$

Using the equality $\sin(\arctan(x)) = \frac{x}{\sqrt{x^2+1}}$, the fact that $\xi \in [0, 1]$ and some basic algebra we reach:

$$\sin|\theta(a, \kappa)| \leq \frac{1 - \xi}{1 + \xi}$$

thus, proving the claim. □

Corollary 4.3.3. *For every angle $a \in [0, 2\pi)$ the bound*

$$|\theta(a, \kappa)| \leq \arcsin \kappa$$

holds.

Proof. By definition, $\kappa \in [0, 1]$. Using this fact along with (4.10) and simple algebra we can prove that:

$$0 \leq \frac{1 - \xi}{1 + \xi} \leq \kappa.$$

Then, the corollary follows from Lemma 4.3.2. □

According to these results, if $\kappa \approx 1$ the collapse $\theta(a, \kappa)$ can be up to $\frac{\pi}{2}$, thus always sending the state back to $|\psi_0\rangle$, preventing a gradual evolution of the angle. In such a case, the algorithm would lose its quantum speed-up. Interestingly, Corollary 4.3.3 shows we can keep $\theta(a, \kappa)$ small by bounding κ , so that the action of G overcomes the effect of the collapse.

Lemma 4.3.4. *The evolution of the unitary map G on the uniform superposition $|\psi\rangle$ is ε -robust to κ -measurements of χ for*

$$\kappa \leq \frac{1}{\sqrt{|B|}} \quad \text{and} \quad \varepsilon = \sin 3\alpha$$

witnessed by $g(n) = 2n$.

Proof. Let $\{a_n\}$ be the following sequence of angles:

$$a_{n+1} = a_n + 2\alpha$$

$$a_0 = \alpha$$

Recall from Section 4.3.1 that $G^n|\psi\rangle = |a_n\rangle$. Similarly, we define a sequence $\{b_n\}$ satisfying $(\mathcal{W}_\perp G)^n|\psi\rangle = |b_n\rangle$:

$$\begin{aligned} b_{n+1} &= b_n - \theta(b_n, \kappa) + 2\alpha \\ b_0 &= \alpha. \end{aligned} \tag{4.13}$$

Remember that $\sin \alpha = |B|^{-1/2}$, so the bound imposed on κ in the claim can be rephrased as $\kappa \leq \sin \alpha$. Thanks to Corollary 4.3.3, this implies $|\theta(b_n, \kappa)| \leq \alpha$ on every iteration. Then $\alpha \leq b_{n+1} - b_n \leq 3\alpha$ for every $n \in \mathbb{N}$, whereas $a_{n+1} - a_n = 2\alpha$. Therefore, for some $m \leq 2n$ we have that $b_m \leq a_n \leq b_{m+1}$ and $a_n - b_m \leq 3\alpha$. Since $p_\chi(G^n|\psi\rangle) = p_\chi(|a_n\rangle) = \sin^2 a_n$ and $p_\chi((\mathcal{W}_\perp G)^m|\psi\rangle) = p_\chi(|b_m\rangle) = \sin^2 b_m$, it follows that:

$$|p_\chi(G^n|\psi\rangle) - p_\chi((\mathcal{W}_\perp G)^m|\psi\rangle)| = |\sin^2 a_n - \sin^2 b_m|.$$

If we now restrict to $a_n \in [0, \frac{\pi}{2})$ and assume that α is small enough so that $a_n - 3\alpha \geq 0$ we find that $\sin^2 a_n - \sin^2 b_m \leq \sin^2 a_n - \sin^2(a_n - 3\alpha)$; then, using the identity $\sin^2 x - \sin^2(x - y) = \sin(y) \sin(2x - y)$ (which holds for all $x, y \in \mathbb{R}$) we conclude that:

$$\sin^2 a_n - \sin^2 b_m \leq \sin 3\alpha - \sin(2a_n - 3\alpha) \leq \sin 3\alpha.$$

A similar argument follows if $a_n \in [\frac{\pi}{2}, \pi)$ or if a_n is in the third or fourth quadrant so that, for every $n \in \mathbb{N}$:

$$\exists m \leq g(n): \quad |p_\chi(G^n|\psi\rangle) - p_\chi((\mathcal{W}_\perp G)^m|\psi\rangle)| \leq \sin 3\alpha$$

thus, proving the claim. □

From the general result of Lemma 4.2.4 and the discussion following it along with the assumption that α is small — so that $\sin 3\alpha < 1/2$ — we conclude that, with arbitrarily high probability, the algorithm in Figure 4.2 will succeed in finding the marked element $|\star\rangle$ within

$$T_c = gf \left(\frac{2c}{\kappa(1 - 2\varepsilon)} \right)$$

iterations for small $c \in \mathbb{N}$. Combining this with Lemmas 4.3.1 and 4.3.4 we find that, for $\kappa \leq |B|^{-1/2}$, the number of iterations is within $\mathcal{O}(\sqrt{|B|})$.

Remark 4.3.5. Our analysis yields an arguably large constant factor

$$T_c \approx (8c + \frac{\pi}{2})\sqrt{|B|}.$$

However, it is important to remark that this factor is an overestimation due to the simplifications applied when finding f and g in this section. With these simplifications we intended to prioritise clarity of our proof strategy. Tighter bounds would yield a more accurate factor. In fact, Figure 4.4 suggests the average number of iterations before success is approximately $2\sqrt{|B|}$.

Moreover, our choice of $\kappa = |B|^{-1/2}$ is not optimal. To prove Lemma 4.3.4 we only needed that the collapse on each iteration was smaller than the increment of the angle due to G , *i.e.* for every angle a we required that $\theta(a, \kappa) < 2\alpha$. If we set $\kappa = 5|B|^{-1/2}$ and use the bound of $\theta(a, \kappa)$ given by Lemma 4.3.2, we can verify that for an instance of $|B| = 10^6$ we have the bound $\theta(a, \kappa) < 1.3\alpha$, so the collapse is appropriately bounded below 2α . In this particular case, numerical analysis shows that the average number of iterations required is slightly smaller than $K = \frac{\pi}{4}\sqrt{|B|}$ where $\lfloor K \rfloor$ is the number of iterations the standard Grover's algorithm runs for. The optimal value of κ depends on the parameters of the problem — *i.e.* it may be different for different values of $|B|$. We conjecture that such an optimal value of κ exists for every instance of Grover's problem so that our approach and the standard one coincide in their expected number of iterations.

The realisation that for a fine-tuned value of κ our algorithm would perform as well as the standard Grover's algorithm was first discussed in a recent paper [20]. The paper provides an automatic procedure based on differentiable programming that looks for optimal values of the parameters of a quantum program. The authors discuss our algorithm as a use case of their approach and, for small values of $|B|$, they find optimal values for κ satisfying the conjecture above.

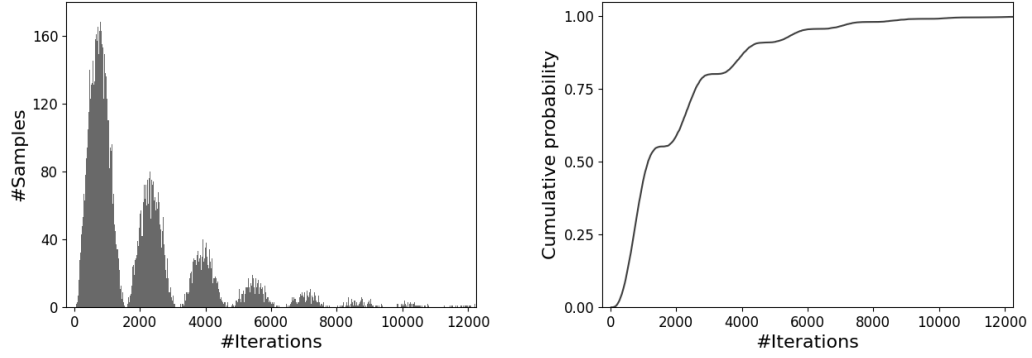


Figure 4.4: Distribution of the number of iterations before success for parameters $|B| = 10^6$ and $\kappa = |B|^{-1/2}$; *left*: histogram, *right*: cumulative probability distribution. Drawn from 10000 samples, obtained by sampling the success probability $p_{\top} = \kappa \sin^2 b_n$ where b_n is given by (4.13). The median is approximately 1000 iterations, and mean 2000 iterations, *i.e.* approximately $\frac{1}{\kappa}$ and $\frac{2}{\kappa}$ respectively.

Figure 4.4 helps us visualise the behaviour of the algorithm: as the angle b_n monotonically increases throughout the iterations, the instantaneous probability of success $p_{\top} = \kappa \sin^2 b_n$ changes periodically. The peaks of the histogram correspond to intervals of $\mathcal{A}$ -iterations where the probability of success approaches its maximum (that is, $p_{\top} \approx \kappa$); these alternate with troughs where the probability of halting approaches 0. These periodic stages of unlikely termination are the cause of the plateaus in the second plot from Figure 4.4 where the cumulative probability of halting — *i.e.* for each $n \in \mathbb{N}$, the probability of halting before n iterations — is graphed. Apart from these plateaus, the curve resembles that of the cumulative probability of a geometric distribution, which is to be expected since these model experiments that “continue until success”. The data points used to draw the figures were obtained by sampling from the success probability $p_{\top} = \kappa \sin^2 b_n$ throughout iterations until success, where b_n is given by (4.13); no simulation of the quantum algorithm is required since we are not running the computation itself.

4.4 Discussion

The κ -while loop is a programming construct that offers a promising abstraction: a way to ‘peek’ at quantum states while computing without sacrificing asymptotic quantum speed-up. The key challenge to its use is the need to strike a balance — by tuning

the measurement strength κ — so that the collapse per iteration is low enough and the information gained is sufficient. The body of a κ -while loop may be any CPTR map, hence κ -while loops may be nested inside each other. As with classical while loops, termination implies the satisfaction of the predicate, which is a useful feature for the analysis of program correctness. We emphasise that these κ -while loops can be realised in any quantum programming language with classical control flow, as indicated in Figure 4.1. More precisely, κ -while loops are a particular instance of the classically controlled while loops described in [55]. Hence, we expect it would be immediate to apply to κ -while loops any findings from the literature on classical control flow of quantum programs; for instance, verification of program correctness [21], study of loop termination [39, 55] and semantics [49, 54]. We conclude this chapter discussing previous works and proposing new avenues of research.

4.4.1 Related work

The key component of the κ -while loop is a weak measurement parametrised by κ . Weak measurements are not new to quantum computer scientists, as they are at the heart of quantum feedback control [58]. Our κ -while loop can be seen as an example of quantum feedback control where the weak measurements are applied at discrete time steps. Discrete time feedback control has been used to protect a single qubit from decoherence [12, 23], while similar notions of weak measurement (over continuous time) have been proposed to monitor and drive complex evolutions [41]. However, weak measurements are rarely used in the design of quantum algorithms; Mizel’s work [46] being the only case we are aware of. By defining the κ -while loop programming construct we are attempting to introduce weak measurements to algorithm designers and programming language experts in a language that is familiar to them.

We have shown that a κ -while loop may be used to implement Grover’s algorithm; however, there is no algorithmic advantage with respect to the standard approach. The reason is that the distribution of $\mathcal{A}$ -iterations throughout Grover’s algorithm is easy to predict, and thus we can fix the number of iterations we should run it for in advance, instead of using a κ -while loop. Our approach trivially extends to the more general setting of amplitude amplification originally proposed in [13].

Mizel [46] proposed a version of Grover’s algorithm that is, in essence, the same as ours. We learned about Mizel’s work while writing the first version of our publication [5]. Our work can be seen as a generalisation of Mizel’s where our study of

Grover’s algorithm is meant as a proof of concept; our main contribution is the proposal of a programming construct — the κ -while loop — and the introduction of tools to study the worst-case complexity of algorithms using it. Mizel presents their algorithm as a fixed point routine, where a mixed state gradually converges to the target state. However, this fixed point behaviour is an artefact of disregarding the outcome of the weak measurement (see Remark 4.1.2): we know the loop will eventually halt and, when it does, the result will be the target state; the fixed point behaviour Mizel observes is that of the cumulative probability of halting converging to 1 as the number of iterations tends to infinity (see Figure 4.4) and, thus, it is an statement about the stochastic behaviour of multiple runs of the algorithm. In contrast, we have focused on describing the evolution of the quantum state in a single run, and we have shown that the state keeps evolving at approximately the same rate throughout the algorithm (*i.e.* the angle increases an amount between α and 3α per iteration) until a $\top$ measurement outcome occurs, in which case the state collapses onto the target subspace. The field of quantum trajectories [14] focuses on understanding these kind of stochastic dynamics in the presence of weak measurements.

It is worth mentioning that algorithms for Grover’s search using a fixed point approach do exist [27, 57]. In these, no measurement is applied during execution; instead, each iteration applies a unitary parametrised by a value that is gradually reduced throughout the algorithm. Intuitively, each iteration moves the state closer to the target, but each time the step is smaller to avoid overshooting. The drawback of this fixed point approach is that, unless we can implement a circuit where the unitary’s parameter can be changed during runtime, each iteration requires a different circuit.

4.4.2 Applications beyond quantum search

We have presented a κ -while loop version of Grover’s algorithm as a proof of concept. However, the pragmatic value of κ -while loops cannot be confirmed unless we find examples of quantum algorithms where the distribution of $\mathcal{A}$ -iterations is unknown or it would be costly to predict. We believe such examples may be found in the field of quantum walk based algorithms. The literature on quantum walks provides multiple results where quantum computers exhibit an advantage in the study of Markov processes [51, 4]. The unitary evolution of these quantum walks is defined in terms of the transition matrix of the Markov process, which is often only required to be symmetric and ergodic. Thus, the literature from this field offers a diverse class of ex-

amples of quantum algorithms whose evolution may be arbitrarily complex. We argue there may be Markov processes whose quantum walk can be proven to be robust to κ -measurements (as per Definition 4.2.3) while, at the same time, being complex enough to prevent an accurate estimation of when a desired state would be reached. The argument goes as follows: on one hand, the choice of κ may be determined by a lower bound of the rate at which the walker traverses the graph (so that κ -measurements are not so strong that they would prevent the walker from reaching certain regions of the graph). This rate may be estimated as the infimum of local rates, using notions such as conductance and effective resistances [11]. On the other hand, in order to predict *when* the walker reaches a particular state, we would need to take into account the global behaviour of the walk, which may be a more daunting task. Thus, in such situations choosing the value of the measurement strength κ may be an easier task than that of deciding the number of iterations in advance.

Another possible avenue of research is the study of continuously measured while loops. The notion of weak measurement used in this chapter is discrete, but continuous measurements have been studied extensively in the literature of quantum feedback control [35]. We can entertain the idea of a while loop whose body describes an infinitesimal step of the evolution — for instance, by providing a Hamiltonian — thus extending κ -while loops to the continuous-time setting. This may be valuable from an experimentalist perspective; for instance, it hints at implementations of algorithms (for instance, Grover’s search) where the probe is continuously measured throughout the execution, thus simplifying control: we no longer need to know when each iteration finishes and the next one starts.

Chapter 5

Final remarks

The driving motivation of this thesis has been the study of control flow in quantum programs, with particular focus on unbounded iterative loops. We have studied both the case of quantum control flow and the case of classical control flow. Along the way we have presented multiple theoretical results whose relevance was discussed within their corresponding chapters. In this final chapter we wish to highlight the two contributions of major pragmatic value.

Chapter 3 culminates with Theorem 3.4.23, establishing that $(\mathbf{LSI}_{\leq}, \oplus, \text{ex})$ is a totally traced category. The morphisms in $\mathbf{LSI}_{\leq}$ describe quantum processes over discrete time and the execution formula in this category represents quantum iteration. This result opens up the possibility of providing categorical semantics for quantum programming languages using *quantum* control flow. Indeed, in Section 5.1 we sketch a toy programming language with quantum control flow and we provide its denotational semantics in the category $\mathbf{LSI}_{\leq}$.

In Chapter 4 we have proposed a novel programming primitive: the κ -while loop. These κ -while loops are an instance of *classical* control flow since the decision-making process is dependent on the outcome of a measurement. This primitive may be introduced into any quantum programming language supporting classical control; we discuss how this may be achieved in Section 5.2 and provide denotational semantics in CPTR for the κ -while loop.

5.1 A toy language with quantum control flow

In this section we sketch the bare bones of a programming language $\mathbf{qWhile}$ supporting quantum control flow and provide its denotational semantics using the results

from Chapter 3. Previous works [56, 15, 17] have proposed programming languages and mathematical frameworks aiming to describe quantum computation with quantum control. The shortcomings of the approach proposed by Ying, Yu and Feng [56] were discussed in Section 1.1.1, whereas Section 3.5 discussed the connection between our work and that of [15]. The authors of [17] proposed the PBS-calculus, a diagrammatic language used to describe and reason about quantum computation with quantum control flow; at the end of this section we compare $\mathbf{qWhile}$ with their approach.

Definition 5.1.1. We use finite sets A, B to specify the input and output types of programs $P: A \rightarrow B$ in $\mathbf{qWhile}$. Every program $P: A \rightarrow B$ in $\mathbf{qWhile}$ is inductively defined using the following syntax:

$$U \mid \Delta_t \mid P; P' \mid P \oplus P' \mid \text{do } P \text{ while } X$$

- there is a *unitary primitive* $U: A \rightarrow A$ for every finite set A and every unitary $U: \mathbb{C}^{|A|} \rightarrow \mathbb{C}^{|A|}$;
- there is a *delay primitive* $\Delta_t: \{\bullet\} \rightarrow \{\bullet\}$ for every $t \in \mathbb{N}$ and every element $\bullet$;
- there is a *sequential combinator* yielding $P; P': A \rightarrow C$ for every two programs $P: A \rightarrow B$ and $P': B \rightarrow C$;
- there is a *parallel combinator* yielding $P \oplus P': A \uplus A' \rightarrow B \uplus B'$ for any two programs $P: A \rightarrow B$ and $P': A' \rightarrow B'$;
- for every program $P: A \uplus X \rightarrow B \uplus X$ there is a *while loop combinator* yielding $(\text{do } P \text{ while } X): A \rightarrow B$.

We choose to write $(\text{do } P \text{ while } X)$ instead of the more usual syntax $(\text{while } X \text{ do } P)$ since its semantics are easier to describe. The semantics of $(\text{do } P \text{ while } X)$ will be given by the execution formula where the shortest execution path is $P_{BA}: A \rightarrow B$ which can be understood as a single application of P . Thus, we want our syntax to suggest that at least one iteration of P will always be applied, hence, we ‘do’ P before we test the loop condition.

The denotational semantics of $\mathbf{qWhile}$ is given by a mapping $\llbracket - \rrbracket$ from programs $P: A \rightarrow B$ to morphisms $\llbracket P \rrbracket \in \mathbf{LSI}_{\leq}(A, B)$. Recall from Section 3.4 that a morphism $f: A \rightarrow B$ in $\mathbf{LSI}_{\leq}$ corresponds to a linear shift invariant contraction $\oplus_A \ell^2 \rightarrow \oplus_B \ell^2$. According to Definition 3.4.18, a morphism $f \in \mathbf{LSI}_{\leq}(A, B)$ is represented as an $\mathbb{R}$ -indexed collection of contractions:

$$f = \{\hat{f}_{\omega} \in \mathbf{FdContraction}(\mathbb{C}^{|A|}, \mathbb{C}^{|B|})\}_{\omega \in \mathbb{R}}.$$

Definition 5.1.2. The denotational semantics of a program $P: A \rightarrow B$ in $\mathbf{qWhile}$ is the morphism $\llbracket P \rrbracket \in \mathbf{LSI}_{\leq}(A, B)$ inductively defined as follows:

- for each *unitary primitive* $U: A \rightarrow A$, let $\llbracket U \rrbracket$ be the $\mathbb{R}$ -indexed collection where $\widehat{\llbracket U \rrbracket}_{\omega} = U$ for all $\omega \in \mathbb{R}$;
- for each *delay primitive* $\Delta_t: \{\bullet\} \rightarrow \{\bullet\}$, let $\llbracket \Delta_t \rrbracket$ be the $\mathbb{R}$ -indexed collection where $\widehat{\llbracket \Delta_t \rrbracket}_{\omega} = e^{-i\omega t}$ for all $\omega \in \mathbb{R}$;
- for each program of the form $P; P'$, let $\llbracket P; P' \rrbracket$ be the $\mathbb{R}$ -indexed collection where $\widehat{\llbracket P; P' \rrbracket}_{\omega} = \widehat{\llbracket P' \rrbracket}_{\omega} \circ \widehat{\llbracket P \rrbracket}_{\omega}$;
- for each program of the form $P \oplus P'$, let $\llbracket P \oplus P' \rrbracket$ be the $\mathbb{R}$ -indexed collection where $\widehat{\llbracket P \oplus P' \rrbracket}_{\omega} = \widehat{\llbracket P \rrbracket}_{\omega} \oplus \widehat{\llbracket P' \rrbracket}_{\omega}$;
- for each program of the form $(\text{do } P \text{ while } X)$ where $P: A \uplus X \rightarrow B \uplus X$, let

$$\llbracket \text{do } P \text{ while } X \rrbracket_{\omega} = \text{ex}^X(\llbracket P \rrbracket_{\omega})$$

where ex is the execution formula in $\mathbf{LSI}_{\leq}$.

The definition of $\llbracket - \rrbracket$ on each syntax primitive follows from the discussion on linear shift invariant maps in Section 3.4. In particular, recall that the elements of the $\mathbb{R}$ -indexed collection representing a linear time-shift invariant map f may be understood as the action — via multiplication — of f on inputs that are periodic signals of angular frequency $\omega \in \mathbb{R}$ (see Remark 3.4.8). Since unitary primitives are instantaneous, these act in the same manner on all input signals, whereas a delay primitive simply changes the phase of the signal. The correctness of these definitions can be checked by obtaining the characteristic function $\chi^U = U \cdot \delta$ and $\chi_t^{\Delta} = S_t[\delta]$ of these LSI maps (see Proposition 3.4.11), then applying the discrete-time Fourier transform to them (Definition 3.4.7). We established in Section 3.4 that, thanks to the convolution theorem 3.4.9, composition of LSI maps can be described in terms of index-wise composition of their corresponding $\mathbb{R}$ -indexed collections; a similar result trivially follows for parallel composition (*i.e.* monoidal product). Finally, we provide semantics for while loops using the execution formula; well-definedness is guaranteed since $(\mathbf{LSI}_{\leq}, \oplus, \text{ex})$ is a totally traced category, as established in Theorem 3.4.23.

Comparison with PBS-calculus [17]. The authors of the PBS-calculus consider two separate state spaces: one being the internal state (holding ‘data’) and another — referred to as the ‘polarisation’ — which determines the execution path the computation

will take. In the case of $\mathbf{qWhile}$ we only capture the ‘polarisation’ space since the state space assigned to each input port $a \in A$ of a program $P: A \rightarrow B$ is the ℓ^2 space of square-summable functions $\mathbb{Z} \rightarrow \mathbb{C}$, *i.e.* we only capture amplitude over time on each port $a \in A$. It would not be conceptually difficult to extend $\mathbf{qWhile}$ so that the inputs and outputs ports may carry an ‘internal state’; it is a matter of increasing dimensions. However, there is a fundamental conceptual difference between $\mathbf{qWhile}$ and the PBS-calculus: the PBS-calculus does not support any primitive that changes the polarisation state during computation. This restriction reduces the expressivity of loops dramatically: the variables used in conditional statements are, in fact, constants defined at the program’s input. The toy programming language $\mathbf{qWhile}$ does not exhibit this restriction.

Notice that the introduction of the time delay primitive Δ_t to the syntax of $\mathbf{qWhile}$ is not necessary for the denotational semantics to be well-defined. In fact, if we remove Δ_t from the syntax we would be able to define the semantics of $\mathbf{qWhile}$ within $(\mathbf{FdContraction}, \oplus, \text{ex})$ — or even $\mathbf{FdUnitary}$ — which we have shown to be a totally traced category as well (Theorem 3.3.15). However, in such a case the semantics would suggest that the different execution paths of the program may interact with each other as if all of them produced their output in the same instant — this is not a physically sound proposal. Thus, the delay primitive Δ_t is added for pragmatic reasons rather than theoretical limitations. An interesting consequence is that we may perceive $\mathbf{qWhile}$ as a language for manipulating wave functions since states are signals rather than qubits. However, recall that $\mathbf{LSI}_{\leq}$ only captures processes over *discrete* time, whereas wave functions are defined over continuous time.

5.2 The κ -while loop in existing programming languages

The κ -while loop primitive may be introduced into any quantum programming language supporting classical control flow and standard while loops. Figure 5.1 shows how this may be achieved:¹ a κ -while loop is nothing more than a standard while loop controlled by a κ -measurement, so it suffices to unpack the definition of κ -measurement (Definition 4.1.1) within the body of the loop. This may be done by a parser that scans the code and replaces any occurrence of a κ -while loop with the equivalent code from Figure 5.1.

On a similar note, the semantics of the κ -while loop may also be defined in terms

¹Figure 5.1 is a copy of Figure 4.1, reproduced here for the reader’s convenience.

$$\begin{array}{ll}
\text{while } !_{\kappa} Q[\rho] \text{ do} & q \leftarrow |\perp\rangle\langle\perp| \\
\quad \rho \leftarrow \mathcal{C}(\rho) & \text{while } M[q] = \perp \text{ do} \\
\text{end} & \quad \rho \leftarrow \mathcal{C}(\rho) \\
& \quad \rho, q \leftarrow E_{\kappa, Q}(\rho \otimes q) \\
& \text{end}
\end{array}$$

Figure 5.1: *Left*: the syntax we use to represent a κ -while loop; $\kappa \in [0, 1]$ is a parameter set by the programmer and Q is the predicate to be measured. *Right*: the pseudocode that implements the κ -while loop on a programming language with classical control flow, following the notation from [54]; $E_{\kappa, Q}$ is defined in Section 4.1 and the condition $M[q] = \perp$ indicates that q is measured to distinguish between $H \otimes \text{span}\{\perp\}$ and $H \otimes \text{span}\{\top\}$, with the condition being satisfied if q collapses to the former.

of the semantics of the base language. In fact, it is as simple as evaluating the semantics of the equivalent code provided in the right hand side of Figure 5.1 which, being comprised only of primitives of the base programming language, should already have well-defined semantics. Nevertheless, and for the purpose of completeness, we construct the morphism in **CPTR** that ought to describe the denotational semantics of the κ -while loop. Let Q be the halting predicate and let κ be its measurement strength; we may construct a function

$$\text{Wk}_{\kappa, Q}: \mathbf{CPTR}(B(H), B(H)) \rightarrow \mathbf{CPTR}(B(H) \oplus B(H), B(H) \oplus B(H))$$

that maps a morphism $\mathcal{C} \in \mathbf{CPTR}(B(H), B(H))$ describing the body of a κ -while loop to its corresponding weakly measured counterpart. To do so, use the isomorphism $h: H \otimes \text{span}\{\perp, \top\} \rightarrow H \oplus H$ and the functor $F: \mathbf{FdContraction} \rightarrow \mathbf{CPTR}$ given in Definition 3.3.18 along with the canonical inclusion $\theta: B(H) \oplus B(H) \rightarrow B(H \oplus H)$ and its left inverse $\phi: B(H \oplus H) \rightarrow B(H) \oplus B(H)$ (see Section 3.3.3).² Then, for every $\mathcal{C} \in \mathbf{CPTR}(B(H), B(H))$ we may define $\text{Wk}_{\kappa, Q}(\mathcal{C})$ as follows:

$$\text{Wk}_{\kappa, Q}(\mathcal{C}) = \phi \circ F(h \circ E_{\kappa, Q}) \circ (\mathcal{C} \otimes \text{id}) \circ F(h^{-1}) \circ \theta$$

where $E_{\kappa, Q}$ is a unitary defined in Section 4.1. Since **CPTR** is a Σ_s -UDC it has canonical quasi-injections $\iota_{\perp}, \iota_{\top}: B(H) \rightarrow B(H) \oplus B(H)$ and quasi-projections $\pi_{\perp}, \pi_{\top}: B(H) \oplus B(H) \rightarrow B(H)$ distinguishing the measurement outcomes $\perp$ and

²Notice that θ may be understood as *state preparation* and ϕ as *measurement* [6].

$\top$. Then, we may define $\text{Wk}_{\kappa,Q}$ in terms of its unique decomposition:

$$\text{Wk}_{\kappa,Q}(\mathcal{C}) = \begin{pmatrix} \pi_{\top} \circ \text{Wk}_{\kappa,Q}(\mathcal{C}) \circ \iota_{\top} & \pi_{\top} \circ \text{Wk}_{\kappa,Q}(\mathcal{C}) \circ \iota_{\perp} \\ \pi_{\perp} \circ \text{Wk}_{\kappa,Q}(\mathcal{C}) \circ \iota_{\top} & \pi_{\perp} \circ \text{Wk}_{\kappa,Q}(\mathcal{C}) \circ \iota_{\perp} \end{pmatrix}$$

so that $\text{ex}^{B(H)}(\text{Wk}_{\kappa,Q})(\mathcal{C})$ corresponds to the morphism in **CPTR** describing the κ -while loop of body $\mathcal{C}$ and halting predicate Q , where the execution formula ex in **CPTR** is the one discussed in Section 3.2.1.

5.3 Conclusion

With this thesis we are attempting to put forward new perspectives on the field of control flow of quantum programs and, in particular, the study of unbounded iteration. On one hand, we have revisited Bartha’s result [10] that quantum processes with instantaneous coherent quantum feedback are well-defined: more formally, that $(\mathbf{FdContraction}, \oplus, \text{ex})$ is a totally traced category. A physical interpretation of this result is that, instead of assuming inputs and outputs come in the form of particles interacting with a device, these are described in terms of plane waves (see Remark 3.4.8) which are, in some sense, everywhere at once; thus, feedback in such a situation does not require a notion of ‘delay’ to be well-defined. Following this intuition, we have used the Fourier transform to decompose any input signal to its plane wave components and used the properties of time-shift invariant maps to prove the novel result that $(\mathbf{LSI}_{\leq}, \oplus, \text{ex})$ is totally traced. Other novel technical results on category theory and, in particular, on categories of Σ -monoids and UDCs have been introduced while building towards our objective. On the other hand, we have introduced a novel programming primitive — the κ -while loop — that in certain cases may be used to classically control the termination condition of a quantum subroutine without sacrificing the quantum speed-up.

Quantum computer scientists tend to dismiss unbounded iteration in quantum algorithms as an uninteresting field of work: in the case of classical control flow due to the assumption that testing a termination condition on every iteration would destroy any achievable quantum speed-up and, in the case of quantum control flow, due to the technical obstacles that interference and the possibility of infinitely many execution paths would entail — these obstacles were discussed in Section 1.1.1. However, prior to this thesis there already were examples of both classically controlled unbounded loops — such as Mizel’s proposal of a measurement-controlled Grover algorithm [26] — and

instances of quantumly controlled unbounded loops — in the form of a wide variety of quantum-walk based algorithms. In this thesis we have established that unbounded iteration in quantum computing is mathematically sound. It is my hope that further study on this field will yield quantum programming languages supporting quantum control flow and new algorithms that make use of unbounded iteration.

Bibliography

- [1] Samson Abramsky. Retracing some paths in process algebra. In *Proceedings 7th International Conference on Concurrency Theory*, pages 1–17. Springer, 1996.
- [2] Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani. Quantum walks on graphs. In *Proceedings 33rd annual ACM symposium on Theory of computing*, pages 50–59, 2001.
- [3] Andris Ambainis. Quantum walk algorithm for element distinctness. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31, 2004.
- [4] Andris Ambainis, András Gilyén, Stacey Jeffery, and Martins Kokainis. Quadratic speedup for finding marked vertices by quantum walks. In *Proceedings 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 412–424, 2020.
- [5] Pablo Andrés-Martínez and Chris Heunen. Weakly measured while loops: peeking at quantum states. *Quantum Science and Technology*, 7(2):025007, feb 2022.
- [6] Pablo Andrés-Martínez, Chris Heunen, and Robin Kaarsgaard. Universal properties of partial quantum maps. In *Proceedings 19th International Workshop on Quantum Physics and Logic (QPL)*, 2022.
- [7] Jean-Pierre Aubin. *Applied functional analysis*. John Wiley & Sons, second edition, 2000.
- [8] Costin Bădescu and Prakash Panangaden. Quantum alternation: Prospects and problems. In *Proceedings 12th International Workshop on Quantum Physics and Logic (QPL)*, volume 195 of *EPTCS*, pages 33–42, 2015.
- [9] Michael Barr. Algebraically compact functors. *Journal of Pure and Applied Algebra*, 82(3):211–231, 1992.

- [10] Miklós Bartha. Quantum Turing automata. In *Proceedings 8th International Workshop on Developments in Computational Models*, volume 143, page 17–31. Open Publishing Association, 2014.
- [11] Aleksandrs Belovs. Quantum walks and electric networks. *Preprint arXiv:1302.3143*, 2013.
- [12] Agata M. Brańczyk, Paulo E. M. F. Mendonça, Alexei Gilchrist, Andrew C. Doherty, and Stephen D. Bartlett. Quantum control of a single qubit. *Physical Review A*, 75(1), 2007.
- [13] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.
- [14] Todd A. Brun. A simple model of quantum trajectories. *American Journal of Physics*, 70(7):719–737, 2002.
- [15] Titouan Carrette, Marc de Visme, and Simon Perdrix. Graphical language with delayed trace: Picturing quantum computing with finite memory. In *Proceedings 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE Computer Society, 2021.
- [16] Andrew M. Childs. Universal computation by quantum walk. *Physical Review Letters*, 102:180501, 2009.
- [17] Alexandre Clément and Simon Perdrix. PBS-Calculus: A Graphical Language for Coherent Control of Quantum Computations. In *Proceedings 45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170, pages 24:1–24:14, 2020.
- [18] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017.
- [19] B.L. Douglas and J.B. Wang. Efficient quantum circuit implementation of quantum walks. *Physical Review A*, 79(5):052335, 2009.
- [20] Wang Fang, Mingsheng Ying, and Xiaodi Wu. Differentiable quantum programming with unbounded loops. To appear, 2021.

- [21] Yuan Feng and Mingsheng Ying. Quantum hoare logic with classical variables. *ACM Transactions on Quantum Computing*, 2(4), 2021.
- [22] D.J.H. Garling. Tensor products of topological abelian groups. *Journal für die reine und angewandte Mathematik*, (223):164–182, 1966.
- [23] G. G. Gillett, R. B. Dalton, B. P. Lanyon, M. P. Almeida, M. Barbieri, G. J. Pryde, J. L. O’Brien, K. J. Resch, S. D. Bartlett, and A. G. White. Experimental feedback control of quantum systems using weak measurements. *Physical Review Letters*, 104(8):080503, 2010.
- [24] Helge Glöckner. Tensor products in the category of topological vector spaces are not associative. *Commentationes Mathematicae Universitatis Carolinae*, 45(4):607–614, 2004.
- [25] Lawrence M. Graves. *The theory of functions of real variables*. McGraw-Hill, 1946.
- [26] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings 28th annual ACM symposium on Theory of computing*, pages 212–219, 1996.
- [27] Lov K. Grover. Fixed-point quantum search. *Physical Review Letters*, 95:150501, 2005.
- [28] Esfandiar Haghverdi. Unique decomposition categories, geometry of interaction and combinatory logic. *Mathematical Structures in Computer Science*, 10(2):205–230, 2000.
- [29] Esfandiar Haghverdi and Philip Scott. Towards a typed geometry of interaction. *Mathematical Structures in Computer Science*, 20(3):473–521, 2010.
- [30] Paul R. Halmos. Normal dilations and extensions of operators. *Summa Brasilensis Mathematica*, 2:125–134, 1950.
- [31] Chris Heunen and Jamie Vicary. *Categories for Quantum Theory: an introduction*. Oxford University Press, 2019.
- [32] Denis Higgs. Σ -groups as convergence groups. *Mathematische Nachrichten*, 137(1):101–112, 1988.

- [33] Naohiko Hoshino. A representation theorem for unique decomposition categories. In *Proceedings 28th Conference on the Mathematical Foundations of Programming Semantics (MFPS)*, volume 286, pages 213–227, 2012.
- [34] Bart Jacobs and Jorik Mandemaker. Coreflections in algebraic quantum logic. *Foundations of physics*, 42(7):932–958, 2012.
- [35] Kurt Jacobs and Daniel A. Steck. A straightforward introduction to continuous quantum measurement. *Contemporary Physics*, 47(5):279–303, 2006.
- [36] André Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3):447–468, 1996.
- [37] John L. Kelley. *General topology*. Van Nostrand, New York, 1955.
- [38] Tom Leinster. *Basic Category Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2014.
- [39] Yangjia Li and Mingsheng Ying. Algorithmic analysis of termination problems for quantum programs. In *Proceedings 45th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*, 2018.
- [40] Noah Linden and Sandru Popescu. The halting problem for quantum computers. *Preprint arXiv quant-ph/9806054*, 1998.
- [41] Seth Lloyd and Jean-Jacques E. Slotine. Quantum feedback with weak measurements. *Physical Review A*, 62:012307, 2000.
- [42] Neil B. Lovett, Sally Cooper, Matthew Everitt, Matthew Trevers, and Viv Kendon. Universal quantum computation using the discrete-time quantum walk. *Physical Review A*, 81:042330, 2010.
- [43] Octavio Malherbe. *Categorical models of computation: partially traced categories and presheaf models of quantum computation*. 2013. PhD thesis.
- [44] Octavio Malherbe, Philip J. Scott, and Peter Selinger. Partially traced categories. *Journal of Pure and Applied Algebra*, 216:2563–2585, 2012.
- [45] Ernest G. Manes and Michael A. Arbib. *Algebraic Approaches to Program Semantics*. Springer New York, 1986.

- [46] Ari Mizel. Critically damped quantum search. *Physical Review Letters*, 102:150501, 2009.
- [47] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):1–8, 2016.
- [48] James R. Munkres. *Topology*. Featured Titles for Topology. Prentice Hall, Incorporated, 2000.
- [49] Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.
- [50] Béla Sz.-Nagy, Ciprian Foias, Hari Bercovici, and László Kérchy. *Harmonic Analysis of Operators on Hilbert Space*. Springer New York, 2010.
- [51] Mario Szegedy. Quantum speed-up of markov chain based algorithms. In *Proceedings 45th Annual IEEE symposium on foundations of computer science*, pages 32–41. IEEE, 2004.
- [52] Martin Varbanov and Todd A. Brun. Quantum scattering theory on graphs with tails. *Physiscal Review A*, 80:052330, 2009.
- [53] Hassler Whitney. Tensor products of Abelian groups. *Duke Mathematical Journal*, 4(3):495 – 528, 1938.
- [54] Mingsheng Ying. *Foundations of Quantum Programming*. Morgan Kaufmann, 2016.
- [55] Mingsheng Ying and Yuan Feng. Quantum loop programs. *Acta Informatica*, 47(4):221–250, 2010.
- [56] Mingsheng Ying, Nengkun Yu, and Yuan Feng. Defining quantum control flow. *Preprint arXiv:1209.4379*, 2012.
- [57] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. Fixed-point quantum search with an optimal number of queries. *Physiscal Review Letters*, 113:210501, 2014.
- [58] Jing Zhang, Yu xi Liu, Re-Bing Wu, Kurt Jacobs, and Franco Nori. Quantum feedback: theory, experiments, and applications. *Physics Reports*, 679:1–60, 2017.