

The Way of the Dagger

Martti Karvonen



Doctor of Philosophy
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2018

(Graduation date: July 2019)

Abstract

A dagger category is a category equipped with a functorial way of reversing morphisms, i.e. a contravariant involutive identity-on-objects endofunctor. Dagger categories with additional structure have been studied under different names in categorical quantum mechanics, algebraic field theory and homological algebra, amongst others. In this thesis we study the dagger in its own right and show how basic category theory adapts to dagger categories.

We develop a notion of a dagger limit that we show is suitable in the following ways: it subsumes special cases known from the literature; dagger limits are unique up to unitary isomorphism; a wide class of dagger limits can be built from a small selection of them; dagger limits of a fixed shape can be phrased as dagger adjoints to a diagonal functor; dagger limits can be built from ordinary limits in the presence of polar decomposition; dagger limits commute with dagger colimits in many cases.

Using cofree dagger categories, the theory of dagger limits can be leveraged to provide an enrichment-free understanding of limit-colimit coincidences in ordinary category theory. We formalize the concept of an ambilimit, and show that it captures known cases. As a special case, we show how to define biproducts up to isomorphism in an arbitrary category without assuming any enrichment. Moreover, the limit-colimit coincidence from domain theory can be generalized to the unenriched setting and we show that, under suitable assumptions, a wide class of endofunctors has canonical fixed points.

The theory of monads on dagger categories works best when all structure respects the dagger: the monad and adjunctions should preserve the dagger, and the monad and its algebras should satisfy the so-called Frobenius law. Then any monad resolves as an adjunction, with extremal solutions given by the categories of Kleisli and Frobenius-Eilenberg-Moore algebras, which again have a dagger.

We use dagger categories to study reversible computing. Specifically, we model reversible effects by adapting Hughes' arrows to dagger arrows and inverse arrows. This captures several fundamental reversible effects, including serialization and mutable store computations. Whereas arrows are monoids in the category of profunctors, dagger arrows are involutive monoids in the category of profunctors, and inverse arrows satisfy certain additional properties. These semantics inform the design of functional reversible programs supporting side-effects.

Lay Summary

This thesis is primarily about the mathematics of so-called dagger categories. For now, think of a category as a collection of systems A, B, C, \dots and processes between them. Processes can be done after each other, and for any system A there should be a trivial process id_A that amounts to doing nothing. In a dagger category there is an additional piece of structure – any process $f: A \rightarrow B$ has a counterpart process $f^\dagger: B \rightarrow A$ going in the opposite direction. One might think of f^\dagger as some kind of a reversal of the process f , and this motivates the rules this reversal is supposed to obey, namely

- reversing the trivial process of any system A results in the trivial process, i.e. $\text{id}_A^\dagger = \text{id}_A$
- reversing a process twice results in the original process, i.e. $f^{\dagger\dagger} = f$
- if one reverses the process “ g after f ”, one might as well have reversed each process individually and done them after each other in the opposite order. More succinctly, $(g \circ f)^\dagger = f^\dagger \circ g^\dagger$

Any mathematical structure obeying these rules is a dagger category. For a concrete example, let us consider the alphabet. Letters can be written after one another, resulting in strings such as “asdf” and “cat”. Strings can be composed by writing them one after the other, so “world” composed after “hello” becomes ‘helloworld’. The empty string is a neutral element with respect to composition, i.e. appending or prepending it to a string doesn’t change the string in question. What makes this into a dagger category is the fact that strings can be reversed. Reversing the empty string results in an empty string. Reversing a string twice results in the original string. Reversing a composite string is the same thing as composing the reversals in the opposite order. Hence the rules above are indeed obeyed by strings of letters.

Dagger categories arise independently both in physics and computation, and also at their intersection in quantum computing. However, so far a systematic study of dagger categories has been missing, and this thesis fills the gap. There are various mathematical questions and notions people study in the context of ordinary categories, such as (co)limits, which consider well-behaved ways of building new systems from old ones, or monads and arrows, which can be used to enlarge the collection of processes available. This thesis shows how to understand such topics in the context of dagger categories.

Acknowledgements

First and foremost I need to thank my first supervisor Chris Heunen for his guidance throughout the years. Besides helping in choosing a topic, nudging me in the right direction when I got stuck, giving advice on literature to read and advising with numerous smaller issues that come about in academic life, he has helped me to grow into a researcher who can hopefully make all these decisions alone. Without him you would be reading an empty thesis¹.

I must also thank my second supervisor Tom Leinster. His intuition, whether pertaining to topics described in this thesis or to mathematics in general, has not only been helpful but also a delight to observe. I hope that I have been able to absorb even a fraction of his way of thinking.

I am very thankful for my examiners Peter Selinger and David Jordan. The time they invested into reading this thesis with care resulted in insightful comments and feedback that has greatly improved this thesis.

I wish to thank Gordon Plotkin for very helpful comments during my yearly reviews and on an earlier draft. It is truly a pleasure to see how a computer scientist thinks about category theory. Similarly, I must thank Robin Kaarsgaard not only for his helpful comments but also for the work we have done together – hopefully this was only a start. To the extent that the thesis falls under theoretical computer science, it is because of Gordon and Robin.

I am indebted for Jamie Vicary both for coining the name “way of the dagger” to describe the general philosophy when working with dagger categories, and for letting me use it as a title. More widely, I am thankful for the whole community centred around the conference Quantum Physics and Logic – it is a great research community to be a part of, and a great source of friends and collaborators.

Before moving to friends and family, I wish to thank various organizations that have helped me. I wish to thank the Osk. Huttunen foundation for financially supporting me during my PhD. The money that Väisälä foundation kindly let me use as a travel grant was very useful for letting me become a part of an international scientific community and I learned a lot on the way. COST Action IC1405 made possible a trip to Copenhagen, without which this thesis would be shorter. My host institute LFCS was not only a great academic community, but also helped me with a funding gap at the start and some travel money during the later years.

¹A variant of [144]

Life in Edinburgh would have been much duller without Toms, Kima and their friends. Climbing with Orfeas has provided me with a much needed counterbalance to research. My summer breaks in Finland were mostly timed so that I could visit KKT, so I have to thank everyone involved for that particular getaway and the support therein. I must also thank Tuomo, Sakari and IHRA more generally for their sustained friendship.

I am grateful for Pentti, Pirjo and Pekka, not only for the usual things loving families provide, but also for being crucial for my intellectual development during my formative years and supportive once I got interested in mathematics. All of my siblings, half or otherwise, have been crucial to me growing up to be the person I am.

There is no way words can express the gratitude I have for Esma, so I will not even try.

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.

(Matti Karvonen)

Contents

1	Introduction	1
1.1	Why dagger categories	1
1.2	Historical Background	2
1.3	Outline, results and content available elsewhere	5
1.4	Prerequisites	6
1.5	On notation	7
2	Background	9
2.1	Dagger categories	9
2.2	Graphical calculus	16
3	Equivalences and adjunctions	21
3.1	Dagger equivalences	21
3.2	Dagger adjunctions	25
4	Limits	29
4.1	Introduction	29
4.2	Dagger limits	30
4.3	Uniqueness up to unitary isomorphism	37
4.4	Completeness	43
4.5	Global dagger limits	50
4.6	Dagger adjoint functors	53
4.7	Polar decomposition	55
4.8	Commutativity of limits and colimits	64
5	Ordinary limit-colimit coincidences via dagger limits	73
5.1	Ambilimits	73
5.2	Biproducts without zero morphisms	75

5.3	Limit-colimit coincidence from domain theory	80
5.4	Fixed points	82
5.5	Traces	99
5.6	A language for traced ω -continuous rig categories	102
6	Monads	107
6.1	Introduction	107
6.2	Frobenius monads	108
6.3	Algebras	110
6.4	Formal monads on dagger categories	115
6.5	Strength	118
6.6	Closure	124
6.7	Limits of algebras	128
6.8	Monadicity	130
7	Arrows	137
7.1	Introduction	137
7.2	Inverse categories	139
7.3	Arrows as an interface for reversible effects	141
7.4	Inverse arrows, categorically	151
7.5	Applications and related work	156
	Bibliography	159
	Index of symbols	173
	Index of concepts	175

Chapter 1

Introduction

1.1 Why dagger categories

In fundamental physical theories, the time evolution of a system is in principle reversible. That is, if the clock was run backwards¹, everything would still comply with the laws of physics. But what is the nature of this “time-reversal” itself? In a sense, this thesis exists in order to study such reversal operations in an abstract setting.

Such reversals occur not only in physics but also in computation. In a reversible computer, each computation step is logically reversible by design. There are many more phenomena that are *weakly reversible*, in the sense that any process $f: A \rightarrow B$ is associated with a corresponding counterpart process $f^\dagger: B \rightarrow A$ going the other way. For instance, in a nondeterministic computer program, for any state of the system there are in principle several possible states to enter next. Such a computer program can naturally be ran in the other direction: given a state of system, just go to one of the possible preceding states. However, doing a computation step back and forth in this manner need not result in nothing happening, hence “weak” reversibility.

One way of studying a concept is to study it together with its environment. Hence physicists study time-evolution in the context of the theory they are working on and say a computer scientist might study reversible computing with all the tools needed for the task. Another, complementary way of understanding a concept is to try to isolate it and abstract away the unnecessary details, studying fundamental properties of the concept on its own. This latter route is the one taken in this thesis, at least for the most part.

First, we abstract away from a particular theoretical context by using the language

¹and charge conjugation & parity transformation are applied as well, see the CPT theorem.

of category theory. Then “weak reversibility” is modelled by a dagger category: a way of associating, to every morphism $f: A \rightarrow B$ a morphism $f^\dagger: B \rightarrow A$ going in the opposite direction. If one thinks of the category in question as modelling physical processes, then the dagger of f might correspond to running the evolution described by f in reverse. Similarly, if the category is modelling reversible computation, a dagger of a computation f would then correspond to running the computation backwards. However, one should not always think of f^\dagger as something that undoes f , but rather as a canonical counterpart of f . For instance, the binary relation “ x is the parent of y ” has a dagger “ y is the offspring of x ”, but the composite relation “ x is the parent of an offspring of y ” is not the identity relation.

Dagger categories have been studied in several schools of thought as discussed in the following section. Usually, the dagger plays a minor part, being merely one of several interlocking pieces of structure assumed. We deviate from tradition by studying the dagger in isolation, and thus studying piece-by-piece, what “cooperation with the dagger” means.

From the point of view of a mathematician, the need to study dagger categories is obvious – if the theory of dagger categories has not been worked out, how sure can we be that we have the correct viewpoint when studying categories where the dagger is merely one of several parts of the structure? Maybe there is an underlying unity behind various kinds of dagger categories which can only be found by going through the foundational topics of category theory with a dagger at hand.

1.2 Historical Background

The tradition from which I enter dagger categories is called categorical quantum mechanics (see e.g. [3, 4] for the founding papers and [38] or the upcoming [80] for a textbook), and indeed it is within this context that the name “dagger categories” came up [129]. In this tradition, one usually uses categories that are either dagger compact or at least dagger monoidal and are often enriched in commutative monoids i.e. sums of morphisms make sense. Such categories are then used to illuminate the structure of quantum computing protocols and results used in quantum computing – it turns out that a surprising amount of quantum computing is not about finite-dimensional complex linear algebra but works in any dagger compact category.

However, this is not the first school of thought to use categories with a dagger. In fact, categories of relations [27, 123] and categories with involution [29, 30] were

already studied in the 60s in the context of homological algebra [110]. This resulted in the notion of a regular category [58, 145, 17], which has sparked considerable amounts of attention (e.g. [99, 98] and [42, 34] in the enriched setting). In this setting the hom-sets are partially ordered [104] like in the standard category of sets and binary relations. See e.g. [49, 32] or [25, Chapter 2] for a modern treatment of the topic.

Another use of dagger categories, perhaps closer to categorical quantum mechanics, comes from the study of operator algebras [52, 68], unitary group representations [45] and algebraic quantum field theory [64]. In these settings, one typically assumes the sets of morphisms to be at least complex vector spaces, or maybe even Banach spaces.

Finally, dagger categories come up in the study of reversible computing [54, 26, 89, 33]. Here one often uses inverse categories [96], which are closely related to inverse semigroups [107]. Once again, one often wants the categories in question to have considerable additional structure, e.g. DCPO-enrichment [93] or various kinds of monoidal products [53].

In each of these schools of thought, the dagger is merely a small piece of the structure that is assumed, but dagger categories have not been studied on their own right – it is almost as if mathematicians had studied particular kinds of enriched categories without ever developing ordinary unenriched category theory. This thesis is an attempt to rectify this omission. The main philosophy when working with dagger categories, called “the way of the dagger” by Jamie Vicary, is that all structure in sight should cooperate with the dagger. What this means for basic notions such as (monoidal) products and isomorphisms has been known for a while, but the dagger counterparts for many standard categorical notions have either been unstudied or only understood in special cases. We fix this by discussing the dagger versions of the topics one would expect to meet in a typical introductory textbook on category theory: equivalences, adjunctions, monads, limits and the like.

Given the power of (enriched) category theory to generalize and unify, one would expect this to fall under already existing theory and to be trivial, or at the very least trivially trivial, to borrow a phrase from Freyd. However, this turns out not to be the case: dagger categories are not enriched categories in any obvious and interesting sense. Similarly, so-called formal category theory [56, 137], i.e. working in a 2-category, possibly with additional structure such as Yoneda structures [138, 141] or proarrow equipments [152, 153] fails to apply for two reasons:

- (i) The 2-category **DagCat** of dagger categories, dagger functors and natural trans-

formations is not just a 2-category, it is in fact a *dagger 2-category*, see Definition 2.1.8 below. This means that the problem is merely shifted a level up: to understand dagger category theory formally, one needs to understand dagger 2-categories.

- (ii) Moreover, even when studying dagger categories, one often wants to also use non-dagger categories. For instance, dagger equalizers are an important kind of a dagger limit, and require stepping outside of **DagCat**.

The main issue, directly resulting in (i) above, is that unlike in ordinary category theory, isomorphic objects are not always similar enough. In a dagger category, the correct notion of an isomorphism is given by a unitary f satisfying $f^{-1} = f^\dagger$. This issue, perhaps pathological for the ordinary category theorist², is at the heart of why dagger category theory differs from ordinary category theory. Indeed, homotopy type theory treats dagger categories on a different footing than ordinary categories [143, 9.7].

What is the upshot of trying to unify such a disparate collection of categorical structures under dagger category theory? Does this unification not resemble the “mustard-watch” of Girard [124]? First and foremost, most of the basic notions such as limits and monads turn out to sit naturally at the level of unenriched dagger categories – when one moves to enriched dagger categories new weights are of course available, but having a clear understanding of the unenriched notion is essential. Understanding clearly the basic categorical notions at the level of dagger categories illuminates what is to be added when one works with dagger categories with additional structure.

Moreover, the results and viewpoints developed can be applied to various domains at hand. In particular, the last two chapters of the thesis discuss topics of interest in ordinary theoretical computer science in the context of dagger categories. Perhaps surprisingly, dagger category theory provides applications also to ordinary category theory: various limit-colimit coincidences can be understood in a new light once one understands dagger limits clearly. A more detailed description of the results is provided in the next section.

²People even wonder whether dagger categories are “evil” in a technical sense. See e.g. the MO-discussion [113] or the nLab-entry [118, section on \dagger -categories]

1.3 Outline, results and content available elsewhere

Chapter 2 develops the background, defining the concepts needed throughout the rest. Apart from a few minor observations, it contains no original results.

Chapter 3 discusses the concepts of equivalences and adjunctions between dagger categories. The main results about equivalences limit the extent to which dagger categories fail to satisfy the “principle of equivalence”[113, 118].

Chapter 4 is a core theoretical chapter of the thesis. We develop a notion of limit for dagger categories, that we show is suitable in the following ways: it subsumes special cases known from the literature; dagger limits are unique up to unitary isomorphism; a wide class of dagger limits can be built from a small selection of them; dagger limits of a fixed shape can be phrased as dagger adjoints to a diagonal functor; dagger limits can be built from ordinary limits in the presence of polar decomposition; dagger limits commute with dagger colimits in many cases. The chapter is based on

C. Heunen and M. Karvonen. Limits in dagger categories. *arXiv:1803.06651*, 2018

which is under peer review.

Chapter 5 applies the viewpoint of the previous chapter to ordinary category theory, providing a way of understanding various cases of limits coinciding with colimits without requiring enrichment. In particular, biproducts can be defined without assuming the existence of zero morphisms. This is studied in Section 5.2 which is based on

M. Karvonen. Biproducts without zero morphisms. *arXiv:1801.06488*, 2018

which is under peer review. Moreover, the limit-colimit coincidence from domain theory can also be understood without enrichment, as shown in Section 5.3. We study this unenriched notion further in Section 5.4, showing that, under suitable assumptions, polynomial functors over a rig category have canonical fixed points. This provides a starting point for interpreting recursive data types in them, and we show how to interpret a language in such categories. We also discuss a dagger variant of the theory.

Chapter 6 is another core theoretical chapter, discussing the notion of a monad on a dagger category. If the monad and its algebras satisfy the so-called Frobenius

law, everything works as expected: dagger adjunctions induce monads and every monad can be resolved as a dagger adjunction, with extremal solutions given by the categories of Kleisli and Frobenius-Eilenberg-Moore algebras. We characterize the Frobenius law as a coherence property between dagger and closure, and characterize strong monads as being induced by Frobenius monoids. Finally, we discuss limits in categories of algebras and prove a monadicity theorem. Apart from the last two topics, which are new, the chapter is based on the conference proceedings paper

C. Heunen and M. Karvonen. Reversible monadic computing. In *Mathematical Foundations of Programming Semantics (MFPS)*, volume 319 of *Electronic Notes in Theoretical Computer Science*, pages 217–237, 2015

and the extended journal version

C. Heunen and M. Karvonen. Monads on dagger categories. *Theory and Applications of Categories*, 31(35):1016–1043, 2016

Chapter 7 discusses arrows, a generalization of monads used to model side-effects in functional programming. We adapt the theory for dagger and inverse categories and discuss several examples, and characterize arrows categorically. This chapter is based on the conference paper

C. Heunen, R. Kaarsgaard, and M. Karvonen. Reversible effects as inverse arrows. In *Mathematical Foundations of Programming Semantics (MFPS)*, volume 341 of *Electronic Notes in Theoretical Computer Science*, pages 179–199. Elsevier, 2018

This same paper has been used by the coauthor Robin Kaarsgaard in his thesis [92, Section C.2].

1.4 Prerequisites

To understand this thesis, an understanding of basic category theory ((monoidal) categories, functors, natural transformations, limits and colimits and adjunctions) is essential and can be obtained from any standard textbook [111, 14, 24, 25]. To appreciate section 6.4, one needs to know 2-category theory at the level of formal theory of monads [137].

Outside of ordinary category theory, our main sources of intuition come from quantum mechanics, quantum computing and reversible computing. However, the reader does not strictly speaking need to be familiar with any of these fields. On occasion some basic knowledge of Hilbert spaces, say at the level of early chapters of one of [31, 66, 122] will be useful. Similarly, in chapter 5 some rudimentary facts about Banach spaces are used. Early chapters of any book on functional analysis should suffice for this. As an example we suggest the textbook [67] for the categorically minded reader.

1.5 On notation

We use a bolded font for naming categories. For example, the category of sets is denoted by **Set** and a generic category by **C**. Objects in a category are usually written with upper case letters A, B, C, \dots or X, Y, Z, \dots , and morphisms are usually written by lower case letters f, g, h, \dots . Functors are denoted $F, G, H \dots$ and natural transformations are denoted by Greek letters. The main exceptions to this are cones and cocones of diagrams $D: \mathbf{J} \rightarrow \mathbf{C}$: we often denote a cone $\Delta L \rightarrow D$ by a generic morphism $l_A: L \rightarrow D(A)$ instead of naming the cone itself as l or using the more cumbersome (but formal) notation $(l_A)_{A \in \mathbf{J}}$ for indexed families. Whether l_A refers to the whole cone or a particular leg of it should be clear from context. The notation $[\mathbf{C}, \mathbf{D}]$ refers to the category of dagger functors $\mathbf{C} \rightarrow \mathbf{D}$, which is itself a dagger category – ordinary functor categories are referred to using **Cat**(**C**, **D**)

The set of natural numbers includes 0 and is denoted by \mathbb{N} . The one-object category induced by the monoid $(\mathbb{N}, +)$ is also denoted by \mathbb{N} , whereas when we view the ordered set (\mathbb{N}, \leq) as a category we denote it by ω . Other notational conventions are either explained as they appear or clear given the context. We refer a reader confused by notation to the Index of symbols. If instead of notation the reader is confused by a word, we suggest starting from the Index of concepts.

Chapter 2

Background

2.1 Dagger categories

This section sets the scene by discussing dagger categories themselves. Dagger categories can behave rather differently than ordinary categories, see e.g. [143, 9.7]. We start by establishing terminology and setting conventions.

Definition 2.1.1. , A *dagger* is a contravariant involutive identity-on-objects functor. Explicitly a dagger is a functor $\dagger: \mathbf{C}^{\text{op}} \rightarrow \mathbf{C}$ satisfying $A^\dagger = A$ on objects and $f^{\dagger\dagger} = f$ on morphisms. A *dagger category* is a category equipped with a dagger.

Example 2.1.2. Examples of dagger categories abound. We first give some concrete examples, and will later add more abstract constructions.

- Any monoid M equipped with an involutive homomorphism $f: M^{\text{op}} \rightarrow M$ may be regarded as a one-object dagger category with $x^\dagger = f(x)$. For example: the complex numbers with conjugation, either under addition or multiplication. Or: the algebra of (complex) n -by- n matrices with conjugate transpose.
- The category **Hilb** of (complex) Hilbert spaces and bounded linear maps is a dagger category, taking the dagger of $f: A \rightarrow B$ to be its adjoint, i.e. the unique morphism satisfying $\langle f(a), b \rangle = \langle a, f^\dagger(b) \rangle$ for all $a \in A$ and $b \in B$. The full subcategory of finite-dimensional Hilbert spaces **FHilb** is a dagger category too.
- **Hilb** and **FHilb** have wide subcategories **Hilb** $_{\leq 1}$ and **FHilb** $_{\leq 1}$ respectively, where one restricts morphisms to non-expansive linear maps, i.e. maps f satisfying $\|fx\| \leq \|x\|$ for all x . The dagger is inherited from **Hilb**.

- The category **DStoch** has finite sets as objects. A morphism $A \rightarrow B$ is a matrix $f: A \times B \rightarrow [0, 1]$ that is doubly stochastic, i.e. satisfies $\sum_{a \in A} f(a, b) = 1$ for all $b \in B$ and $\sum_{b \in B} f(a, b) = 1$ for all $a \in A$. This becomes a dagger category with $f^\dagger(b, a) = f(a, b)$.
- In the category **Rel** with sets as objects and relations $R \subseteq A \times B$ as morphisms $A \rightarrow B$, composition is given by $S \circ R = \{(a, c) \mid \exists b \in B: (a, b) \in R, (b, c) \in S\}$. This becomes a dagger category with $R^\dagger = \{(b, a) \mid (a, b) \in R\}$. The full subcategory **FinRel** of finite sets is also a dagger category.
- **Rel** and **FinRel** have wide subcategories **PInj** and **FPInj** respectively, where one restricts morphisms to partial injections. The dagger is inherited from **Rel**.
- If **C** is a category with pullbacks, the category **Span(C)** of spans is defined as follows. Objects are the same as those of **C**. A morphism $A \rightarrow B$ in **Span(C)** is a span $A \leftarrow S \rightarrow B$ of morphisms in **C**, where two spans $A \leftarrow S \rightarrow B$ and $A \leftarrow S' \rightarrow B$ are identified when there is an isomorphism $S \simeq S'$ making both triangles commute. Composition is given by pullback. The category **Span(C)** has a dagger, given by $(A \leftarrow S \rightarrow B)^\dagger = (B \leftarrow S \rightarrow A)$.
- Any groupoid has a canonical dagger with $f^\dagger = f^{-1}$. Of course, a given groupoid might have other daggers as well. For example, the core of **Hilb** inherits a dagger from **Hilb** that is different from the canonical one it has as a groupoid.

Dagger categories are *self-dual* in a strong sense. Consequently dagger categories behave differently than categories on a fundamental level. Intuitively speaking, categories are built from objects \bullet and morphisms $\bullet \rightarrow \bullet$ whereas dagger categories are built from \bullet and $\bullet \rightrightarrows \bullet$. The result is a theory that is essentially directionless. The guiding principle when working with dagger categories, also known as ‘the way of the dagger’, is that all structure in sight should cooperate with the dagger. This amounts to defining dagger versions of various ordinary notions in category theory. For basic notions this is clear or at least has been already achieved, whereas for more complicated notions it takes some effort to ensure that the dagger notions are truly well-behaved. A lot of the terminology for the dagger counterparts of ordinary notions comes from the category **Hilb**.

Definition 2.1.3. A morphism $f: A \rightarrow B$ in a dagger category is:

- *unitary* (or a *dagger isomorphism*) if $f^\dagger f = \text{id}_A$ and $f f^\dagger = \text{id}_B$;

- an *isometry* (or a *dagger monomorphism*) if $f^\dagger f = \text{id}_A$;
- a *coisometry* (or a *dagger epimorphism*) if $f f^\dagger = \text{id}_B$;
- a *partial isometry* if $f = f f^\dagger f$;

Moreover, an endomorphism $f: A \rightarrow A$ is

- *self-adjoint* if $f = f^\dagger$;
- a *projection* (or a *dagger idempotent*) if $f = f^\dagger = f f$;
- *positive* if $f = g^\dagger g$ for some $g: A \rightarrow B$.

A *dagger subobject* is a subobject that can be represented by a dagger monomorphism. A dagger category is *unitary* when objects that are isomorphic are also unitarily isomorphic.

A morphism f is dagger monic iff f^\dagger is dagger epic. The other concepts introduced above are self-adjoint, i.e. a morphism f is unitary/a partial isometry/self-adjoint/a projection/positive iff f^\dagger is. Notice that dagger monomorphisms are split monomorphisms, and dagger epimorphisms are split epimorphisms. A morphism is dagger monic (epic) if and only if it is both a partial isometry and monic (epic). In diagrams, we will depict partial isometries as \dashrightarrow , dagger monomorphisms as $\dashv\rightarrow$, dagger epimorphisms as $\rightarrow\vdash$, and dagger isomorphisms as $\dashv\rightarrow\vdash$. If f is a partial isometry, then both $f^\dagger f$ and $f f^\dagger$ are projections.

The definition of a dagger subobject might seem odd: two monomorphisms $m: M \hookrightarrow A$ and $n: N \hookrightarrow A$ are considered to represent the same subobject when there is an isomorphism $f: M \rightarrow N$ such that $m = n f$, whereas one might expect that dagger monics m and n represent the same *dagger* subobject if f can be chosen to be unitary. Hence there seems to be a possibility that two dagger monics representing the same subobject might nevertheless represent different dagger subobjects. However, this can not happen: if m, n are dagger monic and f is an isomorphism satisfying $m = n f$, then f is in fact unitary. This is because $m = n f$ implies $n^\dagger m = n^\dagger n f = f$ and $m f^{-1} = n$ implies $f^{-1} = m^\dagger n$ so that $f^{-1} = f^\dagger$.

Example 2.1.4. An *inverse category* is a dagger category in which every morphism is a partial isometry, and positive morphisms commute [96]: $f f^\dagger f = f$ and $f^\dagger f g^\dagger g = g^\dagger g f^\dagger f$ for all morphisms $f: A \rightarrow B$ and $g: A \rightarrow C$. A prototypical inverse category is **Pinj**, the category of sets and partial injections, which is a subcategory of **Rel**.

Remark 2.1.5. Partial isometries need not be closed under composition [72, 5.4]. The same holds for self-adjoint morphisms, projections and positive morphisms. However, given partial isometries $p: A \rightarrow B$ and $q: B \rightarrow C$, their composition qp is a partial isometry whenever the projections $q^\dagger q$ and pp^\dagger commute. Similarly, a morphism f factors through a partial isometry p with the same codomain if and only if $pp^\dagger f = f$.

Definition 2.1.6. A *dagger functor* is a functor $F: \mathbf{C} \rightarrow \mathbf{D}$ between dagger categories satisfying $F(f^\dagger) = F(f)^\dagger$. Denote the category of small dagger categories and dagger functors by **DagCat**.

For a dagger functor F we can and will write Ff^\dagger with no ambiguity. There is no need to go further and define ‘dagger natural transformations’: if $\sigma: F \rightarrow G$ is a natural transformation between dagger functors, then taking daggers componentwise defines a natural transformation $\sigma^\dagger: G \rightarrow F^1$.

Example 2.1.7. If \mathbf{C} and \mathbf{D} are dagger categories, then the category $[\mathbf{C}, \mathbf{D}]$ of dagger functors $\mathbf{C} \rightarrow \mathbf{D}$ and natural transformation is again a dagger category. In particular, taking \mathbf{C} to be a group G and $\mathbf{D} = \mathbf{Hilb}$, this shows that the category of unitary representations of G and intertwiners is a dagger category.

Example 2.1.7 in fact makes **DagCat** into a *dagger 2-category*, as in the following definition. Notice that products of (ordinary) categories actually provide the category **DagCat** with products, so that the following definition makes sense.

Definition 2.1.8. A *dagger 2-category* is a category enriched in **DagCat**, and a *dagger 2-functor* is a **DagCat**-enriched functor.

Concretely, a dagger 2-category is a 2-category where the 2-cells have a dagger that cooperates with vertical and horizontal composition, in the sense that the equations

$$\begin{aligned} (\sigma^\dagger)^\dagger &= \sigma \\ (\tau \circ \sigma)^\dagger &= \sigma^\dagger \circ \tau^\dagger \\ (\sigma * \tau)^\dagger &= (\sigma^\dagger) * (\tau^\dagger) \end{aligned}$$

hold, where \circ denotes vertical composition and $*$ denotes horizontal composition. Dagger 2-functors are 2-functors that preserve this dagger. Strictly speaking, this defines *strict* dagger 2-categories. Defining dagger bicategories is straightforward but not necessary for our purposes.

¹This no longer holds for natural transformations between arbitrary functors into a dagger category, though, see Definition 2.1.14

The forgetful functor $\mathbf{DagCat} \rightarrow \mathbf{Cat}$, has both adjoints. We recall the definitions of free and cofree dagger categories [70, 3.1.17,3.1.19], which will be used in Examples 4.2.3 and 4.2.7 and in the proof of Theorem 5.1.2.

Proposition 2.1.9. *The forgetful functor $\mathbf{DagCat} \rightarrow \mathbf{Cat}$ has a left adjoint $\mathbf{ZigZag}(-)$. The objects of $\mathbf{Zigzag}(\mathbf{C})$ are the same as in \mathbf{C} , and a morphism $A \rightarrow B$ is an alternating sequence of morphisms $A \rightarrow C_1 \leftarrow \cdots \rightarrow C_n \leftarrow B$ from \mathbf{C} , subject to the identifications $(A \xrightarrow{f} C \xleftarrow{\text{id}} C \xrightarrow{g} D \xleftarrow{h} B) = (A \xrightarrow{g \circ f} D \xleftarrow{h} B)$ and $(A \xrightarrow{f} C \xleftarrow{g} D \xrightarrow{\text{id}} D \xleftarrow{h} B) = (A \xrightarrow{f} C \xrightarrow{g \circ h} B)$; composition is given by juxtaposition. The category $\mathbf{Zigzag}(\mathbf{C})$ has a dagger $(f_1, \dots, f_n)^\dagger = (f_n, \dots, f_1)$.*

This adjunction induces a monad on \mathbf{Cat} , and it is easy to see that \mathbf{DagCat} is the category of algebras for this monad.

Proposition 2.1.10. *The forgetful functor $\mathbf{DagCat} \rightarrow \mathbf{Cat}$ has a right adjoint $(-)\rightleftharpoons$, which sends a category \mathbf{C} to the full subcategory of $\mathbf{C}^{\text{op}} \times \mathbf{C}$ with objects of the form (A, A) , and sends a functor F to the restriction of $F^{\text{op}} \times F$. The dagger on $\mathbf{C}\rightleftharpoons$ is given by $(f, g)^\dagger = (g, f)$.*

The first adjunction induces a monad and the second a comonad on \mathbf{Cat} . Hence one might wonder what are the algebras of the monad and coalgebras of the comonad. Both of these categories in fact coincide with \mathbf{DagCat} . Recall that a functor $U: \mathbf{D} \rightarrow \mathbf{C}$ is monadic if there is a monad T on \mathbf{D} and an equivalence of categories $\mathbf{D} \rightarrow \mathbf{C}^T$ such that U is isomorphic to the composite $\mathbf{D} \rightarrow \mathbf{C}^T \rightarrow \mathbf{C}$. Beck's monadicity theorem [19] characterizes monadic functors.

Theorem 2.1.11. *The forgetful functor $U: \mathbf{DagCat} \rightarrow \mathbf{Cat}$ is both monadic and comonadic.*

One could show this directly as follows:

- (i) By the universal property of $\mathbf{ZigZag}(\mathbf{C})$, any dagger on \mathbf{C} induces a map $a: \mathbf{ZigZag}(\mathbf{C}) \rightarrow \mathbf{C}$ such that $a \circ \eta: \mathbf{C} \rightarrow \mathbf{ZigZag}(\mathbf{C}) \rightarrow \mathbf{C}$ equals the identity on \mathbf{C} . The map a can be shown to make \mathbf{C} into an algebra for the monad, and dagger functors induce algebra homomorphisms.
- (ii) Conversely, an algebra structure $a: \mathbf{ZigZag}(\mathbf{C}) \rightarrow \mathbf{C}$ induces a dagger on \mathbf{C} : the dagger on \mathbf{C} can be defined as the composite $a \circ \dagger \circ \eta$, where the \dagger in the middle refers to the dagger on $\mathbf{ZigZag}(\mathbf{C})$. Algebra homomorphisms induce dagger functors.

- (iii) Similarly, by the universal property of $\mathbf{C}_{\rightleftharpoons}$ any dagger on \mathbf{C} gives rise to a map $a: \mathbf{C} \rightarrow \mathbf{C}_{\rightleftharpoons}$ such that $\varepsilon \circ a = \text{id}_{\mathbf{C}}$, where ε is the counit. This gives a coalgebra structure on \mathbf{C} and dagger functors are homomorphisms of the corresponding coalgebras.
- (iv) If $a: \mathbf{C} \rightarrow \mathbf{C}_{\rightleftharpoons}$ is a coalgebra structure on \mathbf{C} , then the composite map $\varepsilon \circ \dagger \circ a$ is a dagger on \mathbf{C} , where the \dagger in the middle refers to the dagger on dagger on $\mathbf{C}_{\rightleftharpoons}$. Homomorphisms of coalgebras are dagger functors between the corresponding dagger categories.
- (v) Constructions (i) and (ii) ((iii) and (iv)) are mutually inverse.

However, we avoid some of the tedious calculations involved by using Beck's monadicity theorem instead. In fact, the following sufficient condition is enough for us:

Theorem 2.1.12. *A functor $U: \mathbf{D} \rightarrow \mathbf{C}$ is monadic if*

- (i) *U has a left adjoint*
- (ii) *U reflects isomorphisms*
- (iii) *\mathbf{D} has and U preserves coequalizers.*

The usual monadicity theorem gives necessary and sufficient conditions for a functor to be monadic (see e.g. [25, Theorem 4.4.4.]). To deduce Theorem 2.1.12 from it, one just notes that conditions (i) and (ii) are as in the usual statement and condition (iii) is stronger than the usual requirement that concerns so-called U -split coequalizers.

Proof of Theorem 2.1.11. To see that U is comonadic, note first that it has a right adjoint by Proposition 2.1.10. Moreover, U clearly reflects isomorphisms: if $F: \mathbf{C} \rightarrow \mathbf{D}$ is a dagger functor that is an isomorphism of the underlying categories, then F^{-1} is also a dagger functor. Hence to use (the dual of) Theorem 2.1.12, it remains to show that **DagCat** has all equalizers and U preserves them. If $F, G: \mathbf{C} \rightarrow \mathbf{D}$ are parallel dagger functors, then the equalizer E of UF, UG is the subcategory of \mathbf{C} containing those objects A and those morphisms f satisfying $FA = GA$ and $Ff = Gf$. Since F and G are dagger functors, if $F(f) = G(f)$, then also $F(f^\dagger) = G(f^\dagger)$ holds. Hence the subcategory E is closed under the dagger of \mathbf{C} , and this induces a dagger on E . It is trivial to show that this gives an equalizer in **DagCat**, and U preserves it since it is a right adjoint by Proposition 2.1.9.

Next we prove that U is monadic. U has a left adjoint by Proposition 2.1.9 and we know that U reflects isomorphisms. Hence it remains to show that **DagCat** has coequalizers and that U preserves them. Preservation is clear since U is a left adjoint. The existence of coequalizers could be shown directly by e.g. using the concrete description of them in **Cat** from [20], but follows readily from U being comonadic: categories of algebras are known to have any limits present in the underlying category [25, Proposition 4.4.1] so by duality **DagCat** has all colimits that **Cat** has, and hence in particular it has coequalizers. \square

Corollary 2.1.13. ***DagCat** is complete and cocomplete.*

Proof. Categories of algebras have all limits present in the underlying category [25, Proposition 4.4.1], so monadicity of $U: \mathbf{DagCat} \rightarrow \mathbf{Cat}$ implies that **DagCat** has all limits that **Cat** has. Since U is also comonadic, **DagCat** has all colimits that **Cat** has. Hence the result follows from **Cat** being complete and cocomplete. \square

Neither of the adjoints of U can be defined to operate on 2-cells, and indeed the forgetful 2-functor $: \mathbf{DagCat} \rightarrow \mathbf{Cat}$ has no 2-adjoints. To see this, note that a right 2-adjoint R to U would, depending on the definition, result in a natural family of isomorphisms/equivalences $\mathbf{DagCat}(C, RD) \cong \mathbf{Cat}(UC, D)$ – but the left hand side is always a dagger category, whereas the right hand side might fail to be self-dual: for an easy example, pick $C = \bullet$ and $D = \bullet \rightarrow \bullet$. The non-existence of a left 2-adjoint is proved similarly.

Definition 2.1.14. Consider a natural transformation $\sigma: F \Rightarrow G$ where F, G are arbitrary functors with codomain a dagger category \mathbf{C} . Taking daggers componentwise generally does not yield a natural transformation $G \Rightarrow F$. When it does, we call σ *adjointable*. For small \mathbf{J} , we denote the dagger category of functors $\mathbf{J} \rightarrow \mathbf{C}$ and adjointable natural transformations by $[\mathbf{J}, \mathbf{C}]_{\dagger}$.

A natural transformation $\sigma: F \Rightarrow G$ is adjointable if and only if σ defines a natural transformation $\dagger \circ F \Rightarrow \dagger \circ G$

Lemma 2.1.15. *If G is a dagger functor and $\sigma: F \Rightarrow G$ is a natural transformation that is pointwise dagger monic, then F is a dagger functor as well.*

Proof. As σ is pointwise dagger monic and natural, the following diagram commutes

for any f :

$$\begin{array}{ccc} FA & \xrightarrow{F(f)} & FB \\ \sigma \downarrow & & \uparrow \sigma^\dagger \\ GA & \xrightarrow{G(f)} & GB \end{array}$$

Taking the dagger of this diagram, and replacing f with f^\dagger , respectively, results in two commuting diagrams:

$$\begin{array}{ccc} FB & \xrightarrow{F(f)^\dagger} & FA \\ \sigma \downarrow & & \uparrow \sigma^\dagger \\ GB & \xrightarrow{G(f)^\dagger} & GA \end{array} \qquad \begin{array}{ccc} FB & \xrightarrow{F(f^\dagger)} & FA \\ \sigma \downarrow & & \uparrow \sigma^\dagger \\ GB & \xrightarrow{G(f^\dagger)} & GA \end{array}$$

Because G is a dagger functor, the bottom arrows in both diagrams are equal. Therefore top arrows are equal too, making F a dagger functor. \square

We end with a useful folklore result²:

Lemma 2.1.16. *Let $F: \mathbf{C} \rightarrow \mathbf{D}$ be a full and faithful functor. Then any dagger on \mathbf{D} induces a unique dagger on \mathbf{C} such that F is a dagger functor.*

Proof. Let \dagger be a dagger on \mathbf{D} . Given a morphism $f: A \rightarrow B$ in \mathbf{C} , we define its dagger to be the unique map $B \rightarrow A$ that F maps to $F(f)^\dagger$. It is easy to check that this assignment gives a dagger on \mathbf{C} and as F is faithful the uniqueness condition is satisfied. \square

2.2 Graphical calculus


Many proofs in chapter 6 are most easily presented in graphical form. This section briefly overviews the graphical calculus that governs monoidal (dagger) categories, such as the category $[\mathbf{C}, \mathbf{C}]$ where our monads will live. For more information, see [131].

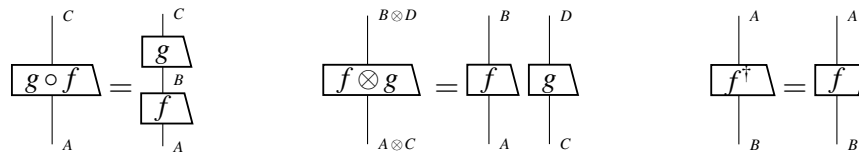
Definition 2.2.1. A (symmetric) monoidal dagger category is a dagger category that is also a (symmetric) monoidal category, satisfying $(f \otimes g)^\dagger = f^\dagger \otimes g^\dagger$ for all morphisms f and g , whose coherence maps $\lambda: I \otimes A \rightarrow A$, $\rho: A \otimes I \rightarrow A$, and $\alpha: (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$ (and $\sigma: A \otimes B \rightarrow B \otimes A$) are unitary.

²Peter Selinger mentions this fact at [118]

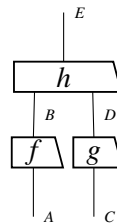
Example 2.2.2. Many monoidal structures on dagger categories make them monoidal dagger categories:

- The dagger category **Rel** is a monoidal dagger category under cartesian product.
- The dagger category **(F)Hilb** is a monoidal dagger category under tensor product.
- For any dagger category **C**, the dagger category $[\mathbf{C}, \mathbf{C}]$ of dagger functors $\mathbf{C} \rightarrow \mathbf{C}$ is a monoidal dagger category under composition of functors.
- Any monoidal groupoid is a monoidal dagger category under $f^\dagger = f^{-1}$.

There is a sound and complete graphical calculus for such categories, that represents a morphism $f: A \rightarrow B$ as , and composition, tensor product, and dagger as follows.



Using these building blocks one can then draw more complicated diagrams such as



denoting a morphism $A \otimes B \rightarrow E$. Tensor product, composition and the dagger extend to larger diagrams straightforwardly: tensor product of diagrams is depicted by juxtaposition, composition by wiring the outputs of one diagram to the inputs of the other³, and the dagger of a diagram amounts to flipping the diagram upside down.

Distinguished morphisms are often depicted with special diagrams instead of generic boxes. For example, the identity $A \rightarrow A$ and the swap map of symmetric monoidal dagger categories are drawn as:



³for this to be defined the outputs must match the inputs

whereas the (identity on) the monoidal unit object I is drawn as the empty picture:

The following definition gives another example: the unit and multiplication of a monoid get a special diagram.

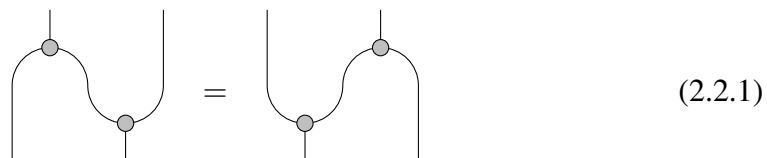
Definition 2.2.3. A *monoid* in a monoidal category is an object A with morphisms $\mu: A \otimes A \rightarrow A$ and $\eta: I \rightarrow A$, satisfying the following equations.



A *comonoid* in a monoidal category is a monoid in the opposite category; an object A with morphisms $\nu: A \rightarrow A \otimes A$ and $\epsilon: A \rightarrow I$ satisfying the duals of the above equations. A monoid in a symmetric monoidal category is *commutative* if it satisfies:



A monoid in a monoidal dagger category is a *dagger Frobenius monoid* if it satisfies the following *Frobenius law*.



The Frobenius law might look mysterious, but will turn out to be precisely the right property to make monads respect daggers. Section 6.6 below will formally justify it as a coherence property between closure and the dagger. For now we illustrate that the Frobenius law corresponds to natural mathematical structures in example categories.

Example 2.2.4. See [73, 79, 147] for more information on the following examples.

- Let \mathbf{G} be a small groupoid, and G its set of objects. The assignments

$$\{*\} \mapsto \{\text{id}_A \mid A \in G\} \quad (f, g) \mapsto \begin{cases} \{f \circ g\} & \text{if } f \circ g \text{ is defined} \\ \emptyset & \text{otherwise} \end{cases}$$

define a dagger Frobenius monoid in **Rel** on the set of morphisms of **G**. Conversely, any dagger Frobenius monoid in **Rel** is of this form.

- Let **G** be a finite groupoid, and G its set of objects. The assignments

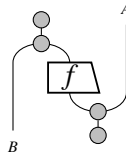
$$1 \mapsto \sum_{A \in G} \text{id}_A \qquad f \otimes g \mapsto \begin{cases} f \circ g & \text{if } f \circ g \text{ is defined} \\ 0 & \text{otherwise} \end{cases}$$

define a dagger Frobenius monoid in **(F)Hilb** on the Hilbert space of which the morphisms of **G** form an orthonormal basis. Conversely, any dagger Frobenius monoid in **(F)Hilb** is of this form.

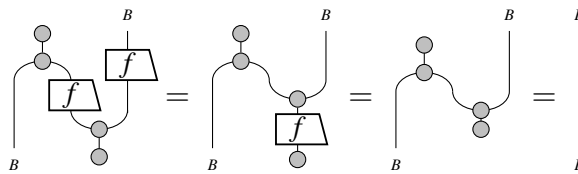
The following lemma exemplifies graphical reasoning. Recall that a (co)monoid homomorphism is a morphism f between (co)monoids satisfying $f \circ \flat = \flat$ and $f \circ \flat' = \flat' \circ (f \otimes f)$ ($\flat \circ f = \flat$ and $\flat' \circ f = (f \otimes f) \circ \flat'$).

Lemma 2.2.5. *A monoid homomorphism between dagger Frobenius monoids in a monoidal dagger category, that is also a comonoid homomorphism, is an isomorphism.*

Proof. Construct an inverse to $A \xrightarrow{f} B$ as follows:



The composite with f gives the identity in one direction:



The third equality uses the Frobenius law (2.2.1) and the unit law. The other composite is the identity by a similar argument. □

Chapter 3

Equivalences and adjunctions

3.1 Dagger equivalences

Given an equivalence of categories $F: \mathbf{C} \rightleftarrows \mathbf{D}: G$ and a dagger on \mathbf{C} , one would like to be able to make \mathbf{D} into a dagger category in such a manner that F and G would be an equivalence of dagger categories. Unfortunately, this isn't possible in general: while Lemma 2.1.16 gives a dagger on \mathbf{D} so that G is a dagger functor, F can fail to be a dagger functor. Selinger [132] gave the following argument to this effect. Consider the forgetful functor $F: \mathbf{FHilb} \rightarrow \mathbf{FVect}$. Since F is full, faithful, and essentially surjective on objects, it is part of an equivalence of categories. Equip the same vector space V with two different inner products, and consider the map f between them defined by $v \mapsto v$. Then f is not unitary, but $F(f) = \text{id}_V$ is unitary with respect to any dagger structure on \mathbf{FVect} . Thus there is no dagger on \mathbf{FVect} such that F would be a dagger functor.

While this argument shows that arbitrary equivalences cannot be lifted to equivalences in the 2-category \mathbf{DagCat} , one can argue that this endeavour is misguided in the first place: if one takes “the way of the dagger” seriously, we see that \mathbf{DagCat} isn't just a 2-category, but it is a *dagger* 2-category, so the notion of equivalence should be different – one should require the relevant natural isomorphisms to be unitary as in the definition below.

Definition 3.1.1. A dagger equivalence in a dagger 2-category consists of 1-cells

$$F: \mathbf{C} \rightleftarrows \mathbf{D}: G$$

and *unitary* isomorphisms $\eta: \text{id}_{\mathbf{C}} \rightarrow GF$ and $\varepsilon: FG \rightarrow \text{id}_{\mathbf{D}}$, i.e. η and ε are invertible 2-cells satisfying $\eta^\dagger = \eta^{-1}$ and $\varepsilon^\dagger = \varepsilon^{-1}$.

Specifically, a dagger equivalence between dagger categories \mathbf{C} and \mathbf{D} consists of dagger functors $F: \mathbf{C} \rightleftarrows \mathbf{D}: G$ and unitary natural transformations $\eta: \text{id}_{\mathbf{C}} \rightarrow GF$ and $\varepsilon: FG \rightarrow \text{id}_{\mathbf{D}}$.

Now, if (\mathbf{C}, \dagger) is a dagger category and $F: \mathbf{C} \rightleftarrows \mathbf{D}: G, \eta: \text{id}_{\mathbf{C}} \rightarrow GF, \varepsilon: FG \rightarrow \text{id}_{\mathbf{D}}$ is an arbitrary equivalence in \mathbf{Cat} , one would not expect it to lift to a dagger equivalence – at the very least, η and $G\varepsilon$ should be unitary. Failing to lift to a equivalence in the underlying 2-category of \mathbf{DagCat} , while a common occurrence, is less interesting as that is not the right notion of equivalence for dagger categories. However, these two necessary conditions turns out to be sufficient. Before the proof, we give the two lemmas.

Lemma 3.1.2. *If $F, G: \mathbf{C} \rightarrow \mathbf{D}$ are functors that are unitarily isomorphic, then F preserves the dagger iff G does.*

Proof. Assume that F preserves the dagger and let $\sigma: F \rightarrow G$ be a unitary isomorphism. As σ is unitary, the following diagrams commute:

$$\begin{array}{ccc} GB & \xrightarrow{(Gf)^\dagger} & GA \\ \sigma^\dagger \downarrow & & \uparrow \sigma \\ FB & \xrightarrow{(Ff)^\dagger} & FA \end{array} \qquad \begin{array}{ccc} GB & \xrightarrow{G(f^\dagger)} & GA \\ \sigma^\dagger \downarrow & & \uparrow \sigma \\ FB & \xrightarrow{F(f^\dagger)} & FA \end{array}$$

As F is a dagger functor, the bottom paths in both diagrams are equal, and thus the top paths are as wells, showing that G is a dagger functor. The other direction follows by symmetry. \square

Theorem 3.1.3. *Let (\mathbf{C}, \dagger) be a dagger category and $F: \mathbf{C} \rightleftarrows \mathbf{D}: G, \eta: \text{id}_{\mathbf{C}} \rightarrow GF, \varepsilon: FG \rightarrow \text{id}_{\mathbf{D}}$ be an equivalence in \mathbf{Cat} such that η and $G\varepsilon$ are unitary. Then there is a unique dagger on \mathbf{D} such that F, G, η, ε forms a dagger equivalence.*

Proof. There is a unique dagger on D such that G preserves the dagger. For this dagger, ε is clearly unitary. It remains to show that F preserves the dagger. By Lemma 3.1.2 both FG and GF preserve the dagger. Furthermore $\varepsilon F: FGF \rightarrow F$ is unitary, so that it suffices to prove that FGF preserves the dagger. We calculate:

$$FGF(f^\dagger) = F((GFf)^\dagger) = FG((Ff)^\dagger) = (FGFf)^\dagger$$

\square

For adjoint equivalences the situation is even cleaner since the condition that $G\varepsilon$ is unitary is redundant: in other words exactly one half of the equivalence needs to cooperate with the dagger.

Corollary 3.1.4. *Let (\mathbf{C}, \dagger) be a dagger category and $F: \mathbf{C} \rightleftarrows \mathbf{D}: G$, $\eta: \text{id}_{\mathbf{C}} \rightarrow GF$, $\varepsilon: FG \rightarrow \text{id}_{\mathbf{D}}$ be an adjoint equivalence in \mathbf{Cat} such that η is unitary. Then there is a unique dagger on \mathbf{D} such that $(F, G, \eta, \varepsilon)$ forms a dagger equivalence.*

Proof. By Theorem 3.1.3 it suffices to show that $G\varepsilon$ is unitary. Since the equivalence in question is an adjoint equivalence we have $G\varepsilon \circ \eta G = \text{id}$ whence $G\varepsilon$ is the inverse of a unitary isomorphism and hence unitary itself. \square

As with standard equivalences, one can only work with one half of a dagger equivalence. This is made precise by the following Lemma which we cite from [148].

Lemma 3.1.5 (Vicary). *If $F: \mathbf{C} \rightarrow \mathbf{D}$ is a dagger functor that is full, faithful and unitarily essentially surjective, then it forms part of a dagger equivalence.*

Proof. See [148, Lemma 5.1]. \square

Corollary 3.1.6. *If $F: \mathbf{C} \rightarrow (\mathbf{D}, \dagger)$ is full, faithful and unitarily essentially surjective, then there is a unique dagger on \mathbf{C} such that F forms part of a dagger equivalence.*

Proof. Follows immediately from Lemmas (2.1.16) and (3.1.5). \square

A way to construct the other half $G: \mathbf{FVect} \rightarrow \mathbf{FHilb}$ of the equivalence is to choose an inner product for each vector space. For such G , $FG = \text{id}_{\mathbf{FVect}}$, and thus $G\varepsilon$ is unitary. As there is no way to make this into a dagger equivalence, the Theorem 3.1.3 implies that for such G the isomorphism $\eta: \text{id}_{\mathbf{FHilb}} \rightarrow GF$ is never unitary at every component.

The problem comes from the forgetful functor, but we can replace it with a naturally isomorphic one to get a dagger equivalence. Thus, if one cares about functors only up to natural isomorphism, there is nothing wrong with the equivalence $\mathbf{FHilb} \rightarrow \mathbf{FVect}$. Before proving this, we first state some useful definitions and facts about them.

Definition 3.1.7. Write \cong_{\dagger} for the equivalence relation on objects that is generated by unitaries. A dagger category is *dagger skeletal* if each \cong_{\dagger} -equivalence class is a singleton. A *dagger skeleton* of a dagger category \mathbf{C} is a dagger skeletal subcategory that is dagger equivalent to \mathbf{C} .

Assuming enough choice it is not difficult to prove the following.

Theorem 3.1.8. *Every dagger category has a dagger skeleton, which is also unique up to dagger isomorphism. Two dagger categories are dagger equivalent iff they have dagger isomorphic skeletons.*

Theorem 3.1.9. *Let (\mathbf{C}, \dagger) be a dagger category and $F: \mathbf{C} \rightleftarrows \mathbf{D}: G$ an equivalence in \mathbf{Cat} such that $A \cong_{\dagger} GF(A)$ for every A . Then one can replace G and F with naturally isomorphic functors and lift the resulting equivalence into a dagger equivalence.*

Proof. We begin by reducing to the case in which \mathbf{C} is dagger skeletal. Let \mathbf{C}' be the dagger skeleton of \mathbf{C} and let $I: \mathbf{C}' \rightleftarrows \mathbf{C}: J$ be a dagger equivalence. Now $FI: \mathbf{C}' \rightleftarrows \mathbf{D}: GJ$ is an equivalence satisfying the assumption. If one can replace FI and JG with isomorphic functors F' and G' lifting to a dagger equivalence, then $F'J: \mathbf{C}' \rightleftarrows \mathbf{D}: IG'$ is also a dagger equivalence, and furthermore $F'J \cong FIJ \cong F$ and $IG' \cong IJG \cong G$, proving the claim.

Thus we may assume that \mathbf{C} is dagger skeletal. As F defines G up to natural isomorphism, it suffices to produce a new functor G' so that the unit and counit satisfy the requirements of Theorem 3.1.3. As \mathbf{C} is dagger skeletal, $GFA = A$ for each A . Define $G'(A) = G(A)$ for objects. Now, choose for each A an isomorphism $\epsilon_A: FG'(A) \rightarrow A$ so that $\epsilon_{FG'(A)}: FG'FG'(A) = FG'(A) \rightarrow FG'(A)$ equals id for each A . Define $G'(f)$ to be the unique map $G'(A) \rightarrow G'(B)$ that F maps to $\epsilon_B^{-1} \circ f \circ \epsilon_A: FG'(A) \rightarrow A \rightarrow B \rightarrow FG'(B)$. It is easy to check that G' is functorial, $G'F = \text{id}_{\mathbf{C}}$ and that $\epsilon: FG'A \rightarrow A$ is a natural transformation satisfying $G'\epsilon = \text{id}$. Thus by Theorem 3.1.3 there is a dagger on \mathbf{D} such that F and G' form a dagger equivalence. \square

Corollary 3.1.10. *Let (\mathbf{C}, \dagger) be a unitary dagger category and $F: \mathbf{C} \rightleftarrows \mathbf{D}: G$ an equivalence in \mathbf{Cat} . Then one can replace G and F with naturally isomorphic functors and lift the resulting equivalence into a dagger equivalence.*

Note that the requirement $A \cong_{\dagger} GF(A)$ for every A is strictly weaker than the existence of a unitary natural transformation $\text{id} \rightarrow GF$, as witnessed by the forgetful functor $\mathbf{FHilb} \rightarrow \mathbf{FVect}$. Naively, one might think that even this assumption can be dropped: $A \cong_{\dagger} B$ implies $A \cong B$, so that by modifying G and F up to isomorphism, it might be possible to tweak any equivalence to one satisfying this condition. However, this can fail in situations when $A \cong B$ doesn't imply $A \cong_{\dagger} B$. For example, let \mathbf{C} be the category with two uniquely isomorphic objects, and consider the dagger category $\mathbf{ZigZag}(\mathbf{C})$ from Proposition 2.1.9. The two objects of this category are isomorphic, but not unitarily isomorphic. Thus $\mathbf{ZigZag}(\mathbf{C})$ is equivalent in \mathbf{Cat} to some one-object category

\mathbf{D} , but there is no dagger equivalence between $\mathbf{ZigZag}(\mathbf{C})$ and a dagger category with a single object.

3.2 Dagger adjunctions

This section considers adjunctions that respect daggers.

Definition 3.2.1. A *dagger adjunction* is an adjunction between dagger categories where both functors are dagger functors.

Note that the previous definition did not need to specify left and right adjoints, because the dagger makes the adjunction go both ways. If $F: \mathbf{C} \rightarrow \mathbf{D}$ and $G: \mathbf{D} \rightarrow \mathbf{C}$ are dagger adjoints, say $F \dashv G$ with natural bijection $\theta: \mathbf{D}(FA, B) \rightarrow \mathbf{C}(A, GB)$, then $f \mapsto \theta(f^\dagger)^\dagger$ is a natural bijection $\mathbf{D}(A, FB) \rightarrow \mathbf{C}(GA, B)$, whence also $G \dashv F$. Hence if F and G form a dagger adjunction, we will call G the dagger adjoint of F and vice versa.

For example, a dagger category \mathbf{C} has a zero object if and only if the unique dagger functor $\mathbf{C} \rightarrow \mathbf{1}$ has a dagger adjoint. Here, a *zero object* is one that is both initial and terminal, and hence induces zero maps between any two objects. This is the nullary version of the following example: a product $A \xleftarrow{p_A} A \times B \xrightarrow{p_B} B$ is a *dagger biproduct* when $p_A \circ p_A^\dagger = \text{id}$, $p_A \circ p_B^\dagger = 0$, $p_B \circ p_A^\dagger = 0$, $p_B \circ p_B^\dagger = \text{id}$.

Example 3.2.2. A dagger category \mathbf{C} with zero object has binary dagger biproducts if and only if the diagonal functor $\mathbf{C} \rightarrow \mathbf{C} \times \mathbf{C}$ has a dagger adjoint $(-)\oplus(-)$ with dagger epic counit $(p_A, p_B): (A \oplus B, A \oplus B) \rightarrow (A, B)$, i.e. $(p_A \circ p_A^\dagger, p_B \circ p_B^\dagger) = (\text{id}_A, \text{id}_B)$ for $A, B \in \mathbf{C}$.

Proof. The implication from left to right is routine. For the other direction, a right adjoint to the diagonal is well-known to fix binary products [111, V.5]. If it additionally preserves daggers the product is also a coproduct, so it remains to check that the required equations governing p_A and p_B are satisfied. By naturality, the diagram

$$\begin{array}{ccccc}
 A & \xrightarrow{p_A^\dagger} & A \oplus B & \xrightarrow{p_B} & B \\
 \text{id} \downarrow & & \downarrow \text{id} \oplus 0 & & \downarrow 0 \\
 A & \xrightarrow{p_A^\dagger} & A \oplus B & \xrightarrow{p_B} & B
 \end{array}$$

commutes, so that $p_B \circ p_A^\dagger = 0$. By symmetry $p_A \circ p_B^\dagger = 0$, and the remaining equations hold by assumption. \square

Here is a more involved example of a dagger adjunction.

Example 3.2.3. The monoids $(\mathbb{N}, +)$ and $(\mathbb{Z}, +)$ become one-object dagger categories under the trivial dagger $k \mapsto k$. The inclusion $\mathbb{N} \hookrightarrow \mathbb{Z}$ is a dagger functor. It induces a dagger functor $F: [\mathbb{Z}, \mathbf{FHilb}] \rightarrow [\mathbb{N}, \mathbf{FHilb}]$, which has a dagger adjoint G .

Proof. An object of $[\mathbb{Z}, \mathbf{FHilb}]$ is a self-adjoint isomorphism $T: A \rightarrow A$ on a finite-dimensional Hilbert space A , whereas an object of $[\mathbb{N}, \mathbf{FHilb}]$ is just a self-adjoint morphism $T: A \rightarrow A$ in \mathbf{FHilb} . To define G on objects, notice that a self-adjoint morphism $T: A \rightarrow A$ restricts to a self-adjoint surjection from $\ker(T)^\perp = \overline{\text{Im } T}$ to itself, and by finite-dimensionality of A hence to a self-adjoint isomorphism $G(T)$ on $\text{Im } T$.

On a morphism $f: T \rightarrow S$ in $[\mathbb{N}, \mathbf{FHilb}]$, define Gf to be the restriction of f to $\text{Im } T$. To see this is well-defined, i.e. the right diagram below commutes if the left one does,

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ T \downarrow & & \downarrow S \\ A & \xrightarrow{f} & B \end{array} \quad \Longrightarrow \quad \begin{array}{ccc} \text{Im } T & \xrightarrow{G(f)} & \text{Im } S \\ T \downarrow & & \downarrow S \\ \text{Im } T & \xrightarrow{G(f)} & \text{Im } S \end{array}$$

observe that if $b \in \text{Im } T$ then $b = T(a)$ for some $a \in A$, so that $f(b) = fT(a) = Sf(a)$, and hence $f(b) \in \text{Im } S$. This definition of G is easily seen to be dagger functorial.

To prove that F and G are dagger adjoint, it suffices to define a natural transformation $\eta: \text{id} \rightarrow F \circ G$, because $G \circ F$ is just the identity. Define η_T to be the projection $A \rightarrow \text{Im } T$, which is a well-defined morphism in $[\mathbb{N}, \mathbf{FHilb}]$:

$$\begin{array}{ccc} A & \xrightarrow{\eta_T} & \text{Im } T \\ T \downarrow & & \downarrow T \\ A & \xrightarrow{\eta_T} & \text{Im } T. \end{array}$$

Naturality of η boils down to commutativity of

$$\begin{array}{ccc} A & \xrightarrow{\eta_T} & \text{Im } T \\ f \downarrow & & \downarrow f \\ B & \xrightarrow{\eta_S} & \text{Im } S \end{array}$$

which is easy to verify. \square

There are variations on the previous example. For example, $(n, m) \mapsto (m, n)$ induces daggers on $\mathbb{N} \times \mathbb{N}$ and $\mathbb{Z} \times \mathbb{Z}$. A dagger functor $\mathbb{N} \times \mathbb{N} \rightarrow \mathbf{FHilb}$ corresponds to a choice of a normal map, which again restricts to a normal isomorphism on its image. This defines a dagger adjoint to the inclusion $[\mathbb{N} \times \mathbb{N}, \mathbf{FHilb}] \rightarrow [\mathbb{Z} \times \mathbb{Z}, \mathbf{FHilb}]$.

Recall that $F: \mathbf{C} \rightarrow \mathbf{D}$ is a *Frobenius functor* when it has a left adjoint G that is simultaneously right adjoint. This is also called an *ambidextrous adjunction* [106].

Proposition 3.2.4. *If F is a Frobenius functor with adjoint G , then F_{\rightrightarrows} and G_{\leftrightsquigarrow} as in Proposition 2.1.10 are dagger adjoint.*

Proof. If $F: \mathbf{C} \rightarrow \mathbf{D}$ there is a natural bijection

$$\begin{aligned} \mathbf{C}_{\rightrightarrows}((A, A), G_{\leftrightsquigarrow}(B, B)) &= \mathbf{C}(GB, A) \times \mathbf{C}(A, GB) \\ &\cong \mathbf{D}(B, FA) \times \mathbf{C}(FA, B) \\ &= \mathbf{D}_{\leftrightsquigarrow}(F_{\rightrightarrows}(A, A), (B, B)) \end{aligned}$$

because $G \dashv F \dashv G$. \square

Since ordinary adjoints are defined up to unique isomorphism, so are dagger adjoints. One might hope for more, i.e. that for dagger adjoints the unique isomorphism is always unitary. However, this turns out to be false, essentially because universal arrows are not unique up to unitary isomorphism. Recall that if F and F' are left adjoints of G with units η and η' respectively, the unique isomorphism $\sigma: F \rightarrow F'$ compatible with the adjunction is the one making the triangle

$$\begin{array}{ccc} A & \xrightarrow{\eta_A} & GFA \\ & \searrow \eta'_A & \downarrow G(\sigma_A) \\ & & GF'A \end{array}$$

Now it is easy to produce a dagger adjunction where σ is not unitary: set $F = G = \text{id}_{\mathbf{FHilb}}$, and $\eta = \text{id}$ and $\eta' = 2\text{id}$, then there is a unique natural isomorphism $\sigma: \text{id}_{\mathbf{FHilb}} \rightarrow \text{id}_{\mathbf{FHilb}}$ making the triangle

$$\begin{array}{ccc} H & \xrightarrow{\text{id}} & H \\ & \searrow 2\text{id} & \downarrow \sigma_H \\ & & H \end{array}$$

commute, but this isomorphism is not unitary. Note that for dagger equivalences similar issues cannot arise: if η and η' are unitary, then so is $G(\sigma)$ and hence σ .

Chapter 4

Limits

4.1 Introduction

This chapter studies limits in dagger categories. If $l_A: L \rightarrow D(A)$ is a limit for a diagram $D: \mathbf{J} \rightarrow \mathbf{C}$, then $l_A^\dagger: D(A) \rightarrow L$ is a colimit for $\dagger \circ D: \mathbf{J}^{\text{op}} \rightarrow \mathbf{C}$, so L has two universal properties; they should be compatible with each other. Moreover, a *dagger limit* should be unique not just up to mere isomorphism but up to unitary isomorphism.

In Section 4.2 we define the notion of dagger limit that subsumes all known examples. Section 4.3 shows how dagger limits are unique up to unitary isomorphism. Section 4.4 deals with completeness. If a dagger category has ‘too many’ dagger limits, it degenerates (showcasing how dagger category theory can be quite different than ordinary category theory). A more useful notion of ‘dagger completeness’ is defined, and shown to be equivalent to having dagger equalizers, dagger products, and dagger intersections. Section 4.5 formulates dagger limits in terms of an adjoint to a diagonal functor, and Section 4.6 attempts a dagger version of an adjoint functor theorem. Section 4.7 makes precise the idea that polar decomposition turns ordinary limits into dagger limits. Finally, Section 4.8 proves that dagger limits commute with dagger colimits in a wide range of situations.

To end this introduction, let us discuss earlier attempts at defining dagger limits [148]. That work defines a notion of a dagger limit for diagrams $D: \mathbf{J} \rightarrow \mathbf{C}$ where \mathbf{J} has finitely many objects and \mathbf{C} is a dagger category enriched in commutative monoids. We dispense with both requirements. Proposition 4.4.14 shows that when these requirements are satisfied the two notions agree for a wide class of diagrams. However, they do not always agree, as discussed in Example 4.4.15.

4.2 Dagger limits

Before we define dagger limits formally, let us look at few examples to motivate the definition. If $\mathbf{1}$ is the terminal category, then a functor $D: \mathbf{1} \rightarrow \mathbf{Hilb}$ is given by a choice of an object H . Now, a limit of D is given by an object H' and an isomorphism $f: H' \rightarrow H$. However, for a limit of D to qualify as the dagger limit of D , one would expect the isomorphism f to be unitary. However, for some diagrams it seems as if one needs to make choices: a limit of $2 \times -: \mathbb{C} \rightleftharpoons \mathbb{C}: -/2$ is given by an isomorphism to one (and hence to both) copies of \mathbb{C} . For a dagger limit of the same diagram, we can require one of the isomorphisms to be unitary but not both – which copy of \mathbb{C} one should prefer?

We will discuss more instructive examples below, but already it seems like dagger limits have something to do with “normalization”, and one can conceivably choose to normalize at different locations. To keep track of these choices, we build them in the definition below, which is the basic object of study in this chapter. The rest of this section illustrates it.

Definition 4.2.1. Let \mathbf{C} be a dagger category and \mathbf{J} a category. A class Ω of objects of \mathbf{J} is *weakly initial* if for every object B of \mathbf{J} there is a morphism $f: A \rightarrow B$ with $A \in \Omega$, i.e. if every object of \mathbf{J} can be reached from Ω . Let $D: \mathbf{J} \rightarrow \mathbf{C}$ be a diagram and let $\Omega \subseteq \mathbf{J}$ be weakly initial. A *dagger limit of (D, Ω)* is a limit L of D whose cone $l_A: L \rightarrow D(A)$ satisfies the following two properties:

normalization l_A is a partial isometry for every $A \in \Omega$;

independence the projections on L induced by these partial isometries commute, i.e.

$$l_A^\dagger l_A l_B^\dagger l_B = l_B^\dagger l_B l_A^\dagger l_A \text{ for all } A, B \in \Omega.$$

A dagger limit of D is a dagger limit of (D, Ω) , for some weakly initial Ω . If L is a dagger limit of (D, Ω) , we will also write $L = \text{dlim}^\Omega D$. For a fixed \mathbf{J} and Ω , if (D, Ω) has a dagger limit in \mathbf{C} for every D , we will say that \mathbf{C} has all (\mathbf{J}, Ω) -shaped limits.

Note that if L is a dagger limit of (D, Ω) and $\Psi \subset \Omega$ is weakly initial, L is also a dagger limit of (D, Ψ) . Moreover, if L is a dagger limit of (D, Ψ) and (D, Ω) also has a dagger limit, Theorem 4.3.1 will imply that L is also a dagger limit of (D, Ω) .

Example 4.2.2. Definition 4.2.1 subsumes various concrete dagger limits from the literature:

- A terminal object is a limit of the unique functor $\emptyset \rightarrow \mathbf{C}$. As the empty category has no objects, being a dagger limit of $\emptyset \rightarrow \mathbf{C}$ says nothing more than being terminal. In a dagger category, any terminal object is automatically a zero object.
- A *dagger product* of objects A and B in a dagger category with a zero object is traditionally defined [129] to be a product $A \times B$ with projections $p_A: A \times B \rightarrow A$ and $p_B: A \times B \rightarrow B$ satisfying $p_A p_A^\dagger = \text{id}_A$, $p_B p_B^\dagger = \text{id}_B$, $p_A p_B^\dagger = 0$, and $p_B p_A^\dagger = 0$. This is precisely a dagger limit of (D, Ω) , where \mathbf{J} is the discrete category on two objects that D sends to A and B , and Ω necessarily consists of both objects of \mathbf{J} ; see also Example 4.3.2 below.

- A *dagger equalizer* of morphisms $f, g: A \rightarrow B$ in a dagger category is traditionally defined [148] to be an equalizer $e: E \rightarrow A$ that is dagger monic. This is precisely a dagger limit, where $\mathbf{J} = \bullet \rightrightarrows \bullet$ which D sends to f and g , and Ω consists of only the first object, which gets sent to A .

This example justifies why Definition 4.2.1 cannot require Ω to be all of \mathbf{J} in general, otherwise there would be many pairs f, g that have a dagger equalizer in the traditional sense but not in the sense of Definition 4.2.1.

- A *dagger kernel* of a morphism $f: A \rightarrow B$ in a dagger category with a zero object is traditionally defined [74] to be a kernel $k: K \rightarrow A$ that is dagger monic. As a special case of a dagger equalizer it is a dagger limit.
- A *dagger intersection* of dagger monomorphisms $f_i: A_i \rightarrow B$ in a dagger category is traditionally defined [148] to be a (wide) pullback P such that each leg $p_i: P \rightarrow A_i$ of the cone is dagger monic. This is precisely a dagger limit, where Ω consists of all the objects of \mathbf{J} getting mapped to A_i . Since pullback of monics are monic, each p_i is not only a partial isometry but also a monomorphism, and hence a dagger monomorphism.
- If $p: A \rightarrow A$ is a projection, a *dagger splitting* of p is a dagger monic $i: I \rightarrow A$ such that $p = ii^\dagger$ [130]. A dagger splitting of p can be seen as the dagger limit of the diagram generated by p . More precisely, we can take $\Omega = \{A\}$, by definition i is a partial isometry, and if $l: L \rightarrow A$ is another limit, then $m = i^\dagger l$ is the unique map satisfying $l = im$. Conversely, suppose that $l: L \rightarrow A$ is a dagger limit. Then l is a partial isometry, and so the cone l factors through itself via both $l^\dagger l$ and id_L ; but since mediating maps are unique these must be equal, and so l is

dagger monic. Similarly, because p is idempotent, p gives a cone, which factors through l . This implies $ll^\dagger p = p$. Taking daggers we see that $p = pll^\dagger = l^\dagger$ since $pl = l$. We say that \mathbf{C} has dagger splittings of projections if every projection has a dagger splitting.

Example 4.2.3. Let \mathbf{J} be a dagger category, and $D: \mathbf{J} \rightarrow \mathbf{C}$ a dagger functor. Any leg $l_A: L \rightarrow D(A)$ of a cone is a partial isometry if and only if any other leg $l_B: L \rightarrow D(B)$ in the same connected component is. To see this, fix a morphism $f: A \rightarrow B$ in \mathbf{J} , and assume that l_A is a partial isometry.

$$\begin{aligned}
 l_B l_B^\dagger l_B &= D(f) l_A l_A^\dagger D(f)^\dagger D(f) l_A && \text{as } L \text{ is a cone} \\
 &= D(f) l_A l_A^\dagger D(f^\dagger f) l_A && \text{as } D \text{ is a dagger functor} \\
 &= D(f) l_A l_A^\dagger l_A && \text{as } L \text{ is a cone} \\
 &= D(f) l_A && \text{as } l_A \text{ is a partial isometry} \\
 &= l_B && \text{as } L \text{ is a cone}
 \end{aligned}$$

Similarly $l_A^\dagger l_A = l_B^\dagger l_B$ for any objects A and B in the same connected component of \mathbf{J} . These two facts imply that whenever D is a dagger functor, the choice of the parameter Ω doesn't matter when speaking about dagger limits, as the resulting equations are equivalent. Hence whenever D is a dagger functor we omit Ω , and call a dagger limit of D a *dagger-shaped limit*. In particular, whenever every dagger functor $\mathbf{J} \rightarrow \mathbf{C}$ has a dagger limit, we will say that \mathbf{C} has \mathbf{J} -shaped limits. If \mathbf{C} has \mathbf{J} -shaped limits for every small dagger category \mathbf{J} , we will say that \mathbf{C} has dagger-shaped limits. If we wish to say something about dagger limits of functors $\mathbf{J} \rightarrow \mathbf{C}$ that don't necessarily preserve the dagger, we will make it clear by not omitting Ω .

- Any discrete category has a unique dagger, which is always preserved by maps into dagger categories. Thus dagger products can be seen as dagger-shaped limits.
- Any dagger splitting of a projection is a dagger-shaped limit, as in Example 4.2.2.
- We say that a dagger category has *dagger split infima of projections* if, whenever \mathcal{P} is a family of projections on a single object A , it has an infimum admitting a dagger splitting, i.e. a dagger subobject $K \rightarrow A$ such that the induced projection is the infimum of \mathcal{P} . Limits of projections can be defined as dagger-shaped limits: consider the monoid freely generated by a set of idempotents; the dagger

on the monoid fixes those idempotents, and reverses words in them. However, we prefer to think of them instead in terms of the partial order on projections. It is not hard to show that the dagger intersection of a family of dagger monics $m_i: A_i \rightarrow A$ coincides with the dagger limit of the projections $m_i m_i^\dagger: A \rightarrow A_i \rightarrow A$.

- We say that a dagger category has *dagger stabilizers* when it has all **ZigZag**(**E**)-shaped limits, where **E** is the equalizer shape and **ZigZag**(**E**) is the free dagger category on **E** from Proposition 2.1.9. Concretely, a dagger functor with domain **ZigZag**(**E**) is uniquely determined by where it sends E , i.e. by a choice of a parallel morphisms $f, g: A \rightrightarrows B$ in the target category **C**. A cone for such a functor consists of an object X with maps $p_A: X \rightarrow A$ and p_B satisfying $f p_A = g p_A = p_B$ and $f^\dagger p_B = g^\dagger p_B = p_A$. A dagger stabilizer of f and g is then a terminal such cone that also satisfies normalization and independence. Hence the dagger stabilizer of f and g is *not* in general a dagger equalizer. For example, the (dagger) kernel of a linear map $f: A \rightarrow B$ in **FHilb** can be computed as the equalizer of f and 0, whereas the dagger stabilizer of f and 0 is always 0.

Example 4.2.4. One can also understand (anti)symmetrized tensor powers as dagger limits in a monoidal dagger category. The n -th symmetric tensor power of X , denoted by $\text{Sym}^n X$, if it exists, is the dagger limit of the functor $D_X: S_n \rightarrow \mathbf{C}$, where S_n is the symmetric group on n elements, D_X sends the unique object to $X^{\otimes n}$ and a permutation on n elements to the corresponding permutation on $X^{\otimes n}$ built from the symmetry. Symmetric tensor powers are in fact dagger-shaped limits.

If $f: X^{\otimes n} \rightarrow X^{\otimes n}$ is a symmetry, i.e. self-adjoint and unitary, then one can define the f -symmetrized tensor power as the dagger limit of the following diagram. Define **J** by setting its objects to correspond elements of the group S_n . For any $g \in S_n \setminus \{1\}$, **J** has two morphisms $1 \rightarrow g$ and no other non-identity morphisms exist. Define $D: \mathbf{J} \rightarrow \mathbf{C}$ by setting $D(-) = X^{\otimes n}$ on objects, and let one of the two morphisms $1 \rightarrow g$ be mapped to the permutation $X^{\otimes n} \rightarrow X^{\otimes n}$ corresponding to g and the other one mapped to $f^{\text{sgn}(g)}$ where $\text{sgn}(g) \in \{0, 1\}$ is the sign of the permutation g . Then the f -symmetrized tensor product, denoted by $\text{Sym}_f^n X$, is the dagger limit of (D, \mathbf{J}) . When $\mathbf{C} = \mathbf{FHilb}$ and $f = -1$, this corresponds to antisymmetrized tensor powers, and when $f = 1$ one obtains an equivalent description of symmetric tensor powers. More generally, one might let f vary with g , obtaining a tensor power where some transpositions are symmetric and some are asymmetric. Note that each morphism in the image of D is unitary.

Recall that a dagger category is *connected* if every hom-set is inhabited.

Proposition 4.2.5. , *The dagger categories **Rel**, **FinRel**, **PInj**, and **Span(FinSet)** have **J**-shaped limits for any small connected dagger category **J**.*

Proof. Let $D: \mathbf{J} \rightarrow \mathbf{Rel}$ be a dagger functor; we will construct a dagger limit. Write G for the (undirected multi-)graph with vertices $V = \coprod_{A \in \mathbf{J}} D(A)$ and edges $E = \coprod_{f \in \mathbf{J}} D(f)$. Call a vertex $a \in D(A) \subseteq V$ a D -endpoint if $D(f)a = \emptyset$ for some $f: A \rightarrow B$ in \mathbf{J} . Set

$$L = \{X \in \mathcal{P}(V) \mid X \text{ is a path component of } G \text{ with no } D\text{-endpoints}\},$$

and define $l_A: L \rightarrow D(A)$ by $l_A(X) = X \cap D(A)$. We will prove that this is a dagger limit of D , starting with normalization and independence. First we show that $l_A(X) \neq \emptyset$ for any $A \in \mathbf{J}$ and $X \in L$. Pick an element $b \in X$, say $b \in D(B)$, and choose some $f: B \rightarrow A$. Since b is not a D -endpoint, $D(f)b$ is nonempty and contained in $X \cap A = l_A(X)$. Because path components of a graph are disjoint, $l_A^\dagger l_A(X) = X$ for all X . Hence $l_A^\dagger l_A = \text{id}_L$ for all A , establishing normalization and independence.

Next we verify that l_A forms a cone. Path components are closed under taking neighbours, so $D(f)l_A(X) \subseteq D_B(X)$. To see the other inclusion, let $b \in D_B(X) = X \cap B$. Again b isn't a D -endpoint, so $D(f)^\dagger(b) = D(f^\dagger)(b) \subseteq l_A(X)$ is nonempty. Any element of $D(f)^\dagger(b)$ is related to b by $D(f)$. Hence $b \in D(f)l_A(X)$, so $D(f)l_A = l_B$ as desired.

Finally, we verify that the cone l_A is limiting. Let $R_A: Y \rightarrow D(A)$ be any cone. If $x \in D(A)$ is a D -endpoint, say $D(f)x = \emptyset$, then

$$x \notin D(f^\dagger)D(f)D(A) = D(f^\dagger f)D(A) \supseteq D(f^\dagger f)R_A(y) = R_A(y).$$

Hence no $R_A(y)$ contains D -endpoints. Moreover, if $D(f)R_A(y) = R_B(y)$ for all f , then the set $\coprod_{A \in \mathbf{J}} (R_A(y)) \subseteq V$ is closed under taking neighbours in G . As it contains no D -endpoints, it is a union of a set of connected components of G without D -endpoints. Mapping y to this set of connected components, i.e. to a subset of L , defines the unique relation $R: Y \rightarrow L$ satisfying $R_A = l_A \circ R$ for all $A \in \mathbf{J}$.

The same construction works for **FinRel**: one merely needs to check that L is finite whenever each $D(A)$ is. In fact L is finite if at least one $D(A)$ is, since $l_A^\dagger l_A = \text{id}_L$ implies that the function $L \rightarrow \mathcal{P}(D(A))$ corresponding to l_A is injective.

Now consider a dagger functor $D: \mathbf{J} \rightarrow \mathbf{PInj}$ and set

$$L = \{(x_A)_{A \in \mathbf{J}} \in \prod_{A \in \mathbf{J}} D(A) \mid D(f)x_A = x_B \text{ for every } f: A \rightarrow B\},$$

$$l_A((x_A)_{A \in \mathbf{J}}) = x_A.$$

It is easy to verify that this forms a dagger limit.

Dagger limits of a dagger functor $D: \mathbf{J} \rightarrow \mathbf{Span}(\mathbf{FinSet})$ resemble the case of \mathbf{PInj} more than $(\mathbf{Fin})\mathbf{Rel}$. Think of (the isomorphism class of) a span $A \leftarrow \bullet \rightarrow B$ of finite sets as a matrix $R: A \times B \rightarrow \mathbb{N}$ with natural number entries, so that a morphism $f: A \rightarrow B$ in \mathbf{J} maps to $D(f): D(A) \times D(B) \rightarrow \mathbb{N}$. Set

$$L = \{(x_A)_{A \in \mathbf{J}} \in \prod_{A \in \mathbf{J}} D(A) \mid D(f)(x_A, z) = \delta_{z, x_B} \text{ for every } f: A \rightarrow B\},$$

$$l_A((x_A)_{A \in \mathbf{J}}, z) = \delta_{z, x_A}.$$

It is easy to see l_A forms a cone. To see that it is limiting, let $R_A: Y \rightarrow D(A)$ be any cone, and pick $y \in Y$. Consider $x_A \in R_A$ such that $R_A(y, x_A) = n \neq 0$. We will show that if $B \in \mathbf{J}$ and $f: A \rightarrow B$, then $D(f)(x_A, z) = \delta_{z, x_B}$ for some unique x_B , so that x_A extends to a unique family $(x_A)_{A \in \mathbf{J}} \in L$. Consider an arbitrary $f: A \rightarrow B$. Now $D(f^\dagger f)R_A = R_A$, so there has to be some $x_B \in B$ with $D(f)(x_A, x_B) > 0$. If there were several such x_B or if $D(f)(x_A, x_B) > 1$, then $D(f^\dagger f)R_A(y, x_A) > n$, which is a contradiction. Hence x_A extends uniquely to a family $(x_A)_{A \in \mathbf{J}} \in L$. Moreover, if $R_B = D(f)R_A$, then $R_B(y, x_B) = n$ as well. Hence we can define $R: Y \rightarrow L$ by setting $R(y, (x_A)_{A \in \mathbf{J}}) = R_A(y, x_A)$. Now R satisfies $l_B R = R_B$ for each B and it is clearly unique as such. \square

Note that this theorem fails for $\mathbf{Span}(\mathbf{Set})$, since idempotents do not always split. For instance, the idempotent $1 \leftarrow \mathbb{N} \rightarrow 1$ does not admit a splitting. We leave open the question of characterizing exactly which categories of spans or relations admit connected dagger-shaped limits.

The following example illustrates the name ‘independence axiom’ in Definition 4.2.1.

Example 4.2.6. When working in \mathbf{FHilb} , consider \mathbb{C}^2 as the sum of two non-orthogonal lines, e.g. the ones spanned by $|0\rangle$ and $|+\rangle$. Projections to these two lines will give rise to two maps $p_1, p_2: \mathbb{C}^2 \rightarrow \mathbb{C}$ making (\mathbb{C}^2, p_1, p_2) into a categorical product. Moreover, p_1 and p_2 are partial isometries so that the normalization axiom is satisfied. However, the independence axiom fails, and indeed, (\mathbb{C}^2, p_1, p_2) fails to be a dagger product. In other words, the limit structure (\mathbb{C}^2, p_1, p_2) and the colimit structure $(\mathbb{C}^2, p_1^\dagger, p_2^\dagger)$ are not compatible.

Example 4.2.7. Sometimes in ordinary category theory an object is both a limit and a colimit “in a compatible way” to a pair of related diagrams in \mathbf{C} . Usually this is formulated in terms of a canonical morphism from the colimit to the limit being an isomorphism, but in some cases one can instead formulate them as dagger limits in

\mathbf{C}_{\Leftarrow} , the cofree dagger category from proposition 2.1.10. Moreover, if \mathbf{C} has zero morphisms¹, then dagger limits in \mathbf{C}_{\Leftarrow} give rise to such “ambilimits” in \mathbf{C} .

- If \mathbf{C} has zero morphisms, then a biproduct in \mathbf{C} is the same thing as a dagger product in \mathbf{C}_{\Leftarrow} .
- Idempotents in \mathbf{C} split if and only if (dagger) projections in \mathbf{C}_{\Leftarrow} have dagger splittings.
- A more interesting example comes from domain theory, where there is an important limit-colimit coincidence. Regard the partially ordered set (\mathbb{N}, \leq) of natural numbers as a category ω , and let $D: \omega \rightarrow \mathbf{DCPO}$ be a chain of embeddings, i.e. each map in ω is mapped to an embedding by D . Then the embeddings define unique projections, resulting in a chain of projections $D^*: \omega^{\text{op}} \rightarrow \mathbf{DCPO}$. A fundamental fact in domain theory [6, 3.3.2] is that the colimit of D coincides with the limit of D^* . One can go through the construction and show that this “ambilimit” can equivalently be described as the dagger limit of (D, D^*, ω) , where $(D^*, D): \omega^{\text{op}} \rightarrow \mathbf{DCPO}_{\Leftarrow}$.

This viewpoint is developed in more detail in chapter 5.

Example 4.2.8. In an inverse category the normalization and independence axioms of Definition 4.2.1 are automatically satisfied, and hence dagger limits are simply limits.

Example 4.2.9. Fix \mathbf{J} and a weakly initial Ω . If \mathbf{C} has (\mathbf{J}, Ω) -shaped limits, so does $[\mathbf{D}, \mathbf{C}]$. Let $D: \mathbf{J} \rightarrow [\mathbf{D}, \mathbf{C}]$ be a diagram. For each $X \in \mathbf{D}$, there is a dagger limit $L(X) = \text{dlim}^{\Omega} D(-)(X)$ with cone $l_A^X: L(X) \rightarrow D(A)(X)$. For $f: X \rightarrow Y$ in \mathbf{D} , there is a cone $D(A)(f) \circ l_A^X: L(X) \rightarrow D(A)(Y)$, and hence a unique map $L(f): L(X) \rightarrow L(Y)$ satisfying $l_A^Y \circ L(f) = D(A)(f) \circ l_A^X$. The resulting functor $L: \mathbf{D} \rightarrow \mathbf{C}$ is a limit of D , with cone $l_A^X: L(X) \rightarrow D(A)(X)$. This limit is in fact a dagger limit (D, Ω) , because the normalization and independence axioms hold for each component l_A^X , and the dagger in $[\mathbf{D}, \mathbf{C}]$ is computed componentwise.

We end this section by recording how dagger functors interact with dagger limits. Any dagger functor preserves dagger limits as soon as it preserves limits. The same holds for reflection and creation of (dagger) limits when the functor is faithful.

¹This is so that one can give every cone a trivial cocone structure and vice versa.

Lemma 4.2.10. *Let $F: \mathbf{C} \rightarrow \mathbf{D}$ be a dagger functor. If F preserves limits of type \mathbf{J} , then it preserves (\mathbf{J}, Ω) -shaped dagger limits. If F reflects (creates) limits of type \mathbf{J} , then it reflects (creates) (\mathbf{J}, Ω) -shaped dagger limits.*

Proof. All dagger functors preserve partial isometries and commutativity of projections, and faithful dagger functors also reflect these. \square

4.3 Uniqueness up to unitary isomorphism

Theorem 4.3.1. *Let $D: \mathbf{J} \rightarrow \mathbf{C}$ be a diagram and $\Omega \subseteq \mathbf{J}$ be weakly initial. Let L be a dagger limit of (D, Ω) and let M be a limit of D . The canonical isomorphism of cones $L \rightarrow M$ is unitary iff M is a dagger limit of (D, Ω) . In particular, the dagger limit of (D, Ω) is defined up to unitary isomorphism.*

Example 4.3.2. Let \mathbf{J} be a discrete category of arbitrary cardinality. As \mathbf{J} has only one weakly initial class (the one consisting of all objects), the dagger limit of any diagram $D: \mathbf{J} \rightarrow \mathbf{C}$ is unique up to unitary isomorphism. These are exactly the dagger products. However, note that Definition 4.2.1 does not require enrichment in commutative monoids nor the equation

$$\text{id}_{A \oplus B} = l_A^\dagger l_A + l_B^\dagger l_B \quad (4.3.1)$$

(and thus works for infinite \mathbf{J} as well). Moreover, it doesn't require \mathbf{C} to have a zero object or zero morphisms in order to be defined up to unitary iso. On the other hand, if \mathbf{C} has zero morphisms, it is not hard to show directly that a dagger product in the sense of Definition 4.2.1 satisfies the traditional equations involving zero. Moreover, if \mathbf{C} is enriched in commutative monoids, equation (4.3.1) also follows. These facts are proven for more general \mathbf{J} in Propositions 4.4.13 and 4.4.14.

In fact, from Definition 4.2.1 one can glean a definition of (ordinary) biproduct of A and B that is unique up to isomorphism in an ordinary category \mathbf{C} , but doesn't require the existence of zero morphisms, and so generalizes the usual definition. For example, in \mathbf{Set} the biproduct $\emptyset \oplus \emptyset$ exists and is the empty set. Slightly more interestingly, if \mathbf{C} has all binary biproducts (in the traditional sense) and \mathbf{D} is any non-empty category, then $\mathbf{C} \sqcup \mathbf{D}$ has binary biproducts of pairs of objects from \mathbf{C} (in the generalized sense), but doesn't have zero morphisms. Of course, if a category has all binary biproducts in this generalized sense, one can show that it also has zero morphisms, so this definition is more general only in categories with some but not all biproducts. For more details, see Section 5.2.

Example 4.3.3. Let \mathbf{J} be the indiscrete category on n objects. When considering functors $\mathbf{J} \rightarrow \mathbf{C}$ that don't preserve the dagger on \mathbf{J} , the parameter Ω matters. For example, take $n = 2$ in the diagram $D: \mathbf{J} \rightarrow \mathbf{Hilb}$ defined by $D(1) = D(2) = \mathbb{C}$ where $D(1 \rightarrow 2)$ multiplies by 2 but $D(2 \rightarrow 1)$ divides by 2. Now Ω cannot be all of $\{1, 2\}$, because no limiting cone can consist of partial isometries. If $\Omega = \{1\}$, there is a dagger limit $L = \mathbb{C}$ with $l_1 = 1$ and $l_2 = 2$. If $\Omega = \{2\}$, there is a dagger limit $L = \mathbb{C}$ with $l_1 = \frac{1}{2}$ and $l_2 = 1$. These two dagger limits are clearly not unitarily isomorphic.

The same can happen with *chains*, where the preorder of integers is regarded as a category \mathbf{J} . Consider the diagram $D: \mathbf{J} \rightarrow \mathbf{Hilb}$ defined by $D(n) = \mathbb{C}$, and $D(n \rightarrow n+1)$ is multiplication by -1^n . Now Ω can either consist of even numbers or the odd numbers. Hence D has two dagger limits and they are not unitarily isomorphic.

One might hope to get rid of this dependence on Ω by strengthening the definition to select exactly one of a diagram's several dagger limits. However, this is impossible in general. Write $\mathbf{J}(X)$ for the indiscrete category on a nonempty set $X \subset \mathbb{R} \setminus \{0\}$ of objects. Define $D(X): \mathbf{J}(X) \rightarrow \mathbf{FHilb}$ by mapping the unique arrow $x \rightarrow y$ to the morphism $\mathbb{C} \rightarrow \mathbb{C}$ that multiplies by $\frac{x}{y}$. A choice of a dagger limit for each $D(X)$ amounts to a choice function on $\mathbb{R} \setminus \{0\}$. Thus there is no way to strengthen Definition 4.2.1 to make dagger limits unique in a way that doesn't depend on a choice of a weakly initial class.

Example 4.3.4. In the domain theory part of Example 4.2.7, in fact $\Omega = \mathbf{J}$. Hence the bilimit is unique up to unique unitary isomorphism in $\mathbf{DCPO}_{\sqsubseteq}$. In \mathbf{DCPO} , this means that any isomorphism of the limit half of such bilimits, is also an isomorphism of the colimit half.

Example 4.3.5. Consider dagger equalizers of $f, g: A \rightarrow B$ in the sense of Definition 4.2.1. Any weakly initial class must contain A , and thus the dagger equalizer is unique up to unitary iso if it exists. Moreover, it is readily seen to coincide with the traditional definition. For if $e: E \rightarrow A$ is the dagger equalizer in the above sense, then e is monic and a partial isometry, and thus dagger monic, so that e is a dagger equalizer in the traditional sense.

Theorem 4.3.6. Let $l_A: L \rightarrow D(A)$ and $m_A: M \rightarrow D(A)$ be limits of the same diagram $D: \mathbf{J} \rightarrow \mathbf{C}$ where \mathbf{C} is a dagger category, and let $f: L \rightarrow M$ be the unique isomorphism of limits. Then the following conditions are equivalent:

- (i) f is unitary;

(ii) f is also a morphism of colimits;

(iii) the following diagram commutes for any A and B in \mathbf{J} :

$$\begin{array}{ccc} D(A) & \xrightarrow{l_A^\dagger} & L \\ m_A^\dagger \downarrow & & \downarrow l_B \\ M & \xrightarrow{m_B} & D(B) \end{array}$$

Proof. (i) \Rightarrow (ii) By definition f^{-1} is the unique map $M \rightarrow L$ that is compatible with the limit structure, whereas f^\dagger is the unique map $M \rightarrow L$ that is compatible with the colimit structure. As f is unitary, these coincide, whence f^{-1} is simultaneously a map of limits and colimits. Therefore so too is f .

(ii) \Rightarrow (iii) By (ii), both of the triangles in the following diagram commute.

$$\begin{array}{ccc} D(A) & \xrightarrow{l_A^\dagger} & L \\ m_A^\dagger \downarrow & \searrow f & \downarrow l_B \\ M & \xrightarrow{m_B} & D(B) \end{array}$$

(iii) \Rightarrow (i) To prove that $f: L \rightarrow M$ is unitary, it suffices to establish $f \circ f^\dagger = \text{id}_M$. By the two universal properties of M , we may further reduce to pre- and postcomposing with structure maps to and from the diagram D . The following diagram commutes.

$$\begin{array}{ccccc} M & \xrightarrow{f^\dagger} & L & \xrightarrow{f} & M \\ m_A^\dagger \uparrow & & \nearrow l_A^\dagger & \searrow l_B & \downarrow m_B \\ D(A) & \xrightarrow{m_A^\dagger} & M & \xrightarrow{m_B} & D(B) \end{array}$$

Hence f is unitary. □

The previous theorem highlights why one would want limits in dagger categories to be defined up to unitary isomorphism: if $(L, \{l_A\}_{A \in \mathbf{J}})$ is a limit of D , then $(L, \{l_A^\dagger\}_{A \in \mathbf{J}})$ is a colimit of $\dagger \circ D$, and being defined up to unitary iso ensures that the limit and the colimit structures are compatible with each other.

We now set out to prove Theorem 4.3.1. The following lemma will be crucial.

Lemma 4.3.7. Let $l_A: L \rightarrow D(A)$ and $m_A: M \rightarrow E(A)$ be dagger limits of (D, Ω) and (E, Ω) , respectively, where $D, E: \mathbf{J} \rightrightarrows \mathbf{C}$. Let $\sigma: D \rightarrow E$ be an adjointable natural transformation. The following diagram commutes for every A and B in \mathbf{J} .

$$\begin{array}{ccccc}
 D(A) & \xrightarrow{l_A^\dagger} & L & \xrightarrow{l_B} & D(B) \\
 \sigma_A \downarrow & & & & \downarrow \sigma_B \\
 E(A) & \xrightarrow{m_A^\dagger} & M & \xrightarrow{m_B} & E(B)
 \end{array} \tag{4.3.2}$$

Proof. First we show that it is enough to check that (4.3.2) commutes whenever $A, B \in \Omega$. So assume that it does and let $X, Y \in \mathbf{J}$ be arbitrary. By weak initiality of Ω we can find maps $f: A \rightarrow X$ and $g: B \rightarrow Y$ for some $A, B \in \Omega$. Now we prove the claim for X and Y assuming the claim for A and B by showing that the diagram

$$\begin{array}{ccccccc}
 D(X) & \xrightarrow{\sigma_X} & E(X) & & & & \\
 \downarrow l_X^\dagger & \searrow Df^\dagger & \downarrow Ef^\dagger & \xrightarrow{m_X^\dagger} & & & \\
 & & D(A) & \xrightarrow{\sigma_A} & E(A) & \xrightarrow{m_A^\dagger} & M \\
 & \nearrow l_A^\dagger & \downarrow \sigma_B & \searrow m_B & & & \downarrow m_Y \\
 & & D(B) & \xrightarrow{\sigma_B} & E(B) & & \\
 & \nearrow l_B & \downarrow Dg & \searrow Eg & & & \\
 L & \xrightarrow{l_Y} & D(Y) & \xrightarrow{\sigma_Y} & E(Y) & &
 \end{array}$$

commutes. The parallelogram on the bottom (top) commutes because σ is natural (and adjointable). The triangles on the left of these parallelograms commute because L is a cone and the triangles on the right commute because M is. Finally, the remaining shape is just diagram (4.3.2) with $A, B \in \Omega$.

For the rest of the proof, write $l_{A,B} := l_B l_A^\dagger: D(A) \rightarrow L \rightarrow D(B)$ and similarly $m_{A,B} := m_B m_A^\dagger$. Note that $l_{B,A}^\dagger = l_{A,B}$ and $m_{B,A}^\dagger = m_{A,B}$. Moreover, $l_{A,B}$ and $m_{A,B}$ are partial isometries whenever $A, B \in \Omega$ by Remark 2.1.5. Our goal is to show that

$$m_{A,B} \sigma_A = \sigma_B l_{A,B} \tag{4.3.3}$$

for all A and B in \mathbf{J} .

Let $f: L \rightarrow M$ be the unique map making this square commute for all A in \mathbf{J} :

$$\begin{array}{ccc} L & \xrightarrow{l_A} & D(A) \\ f \downarrow & & \downarrow \sigma_A \\ M & \xrightarrow{m_A} & E(A) \end{array}$$

If $A \in \Omega$, then m_A is a partial isometry, so the following diagram commutes:

$$\begin{array}{ccccccc} D(A) & \xrightarrow{l_A^\dagger} & L & \xrightarrow{l_A} & D(A) & \xrightarrow{\sigma_A} & E(A) \\ & & \downarrow l_A & \searrow f & & \nearrow m_A & \uparrow m_A \\ & & D(A) & & M & & E(A) \\ & & \xrightarrow{\sigma_A} & & \downarrow m_A & & \\ & & E(A) & & & \xrightarrow{m_A^\dagger} & M \end{array}$$

This shows that $\sigma_A l_{A,A} = m_{A,A} \sigma_A l_{A,A}$. Repeating the argument with σ replaced by σ^\dagger gives $\sigma_A^\dagger m_{A,A} = l_{A,A} \sigma_A^\dagger m_{A,A}$. Combining the first equation with the dagger of the second shows that for every $A \in \Omega$:

$$m_{A,A} \sigma_A = \sigma_A l_{A,A}. \quad (4.3.4)$$

Next, we check that the following diagram commutes:

$$\begin{array}{ccccccc} D(A) & \xrightarrow{\sigma_A} & E(A) & \xrightarrow{m_A^\dagger} & M & \xrightarrow{m_B} & E(B) \\ & & \downarrow m_A^\dagger & \nearrow m_A^\dagger & \uparrow m_A^\dagger & \searrow m_B & \downarrow m_B \\ & & D(A) & & E(A) & & E(B) \\ & & \downarrow l_A & \nearrow \sigma_A & \downarrow m_A & & \downarrow m_B \\ & & L & \xrightarrow{f} & M & & E(A) \\ & & \downarrow l_B & & \downarrow m_B & & \downarrow m_A \\ D(B) & \xrightarrow{\sigma_B} & E(B) & \xrightarrow{m_B^\dagger} & M \end{array}$$

(i) (ii) (iii) (iv) (v) (vi)

Region (i) commutes since m_A is a partial isometry and region (ii) since m_B is. Region (iii) is equation (4.3.4). Regions (iv) and (v) commute by definition of f . Finally, the commutativity of projections $m_A^\dagger m_A$ and $m_B^\dagger m_B$ shows that region (vi) commutes.

Commutativity of the outermost rectangle says that

$$m_{A,B}\sigma_A = m_{A,B}m_{B,A}\sigma_B l_{A,B} \quad (4.3.5)$$

for every $A, B \in \Omega$.

Exchanging M and L and replacing σ with σ^\dagger in the above diagram now gives $l_{A,B}\sigma_A^\dagger = l_{A,B}l_{B,A}\sigma_B^\dagger m_{A,B}$, and applying the dagger on both sides shows

$$\sigma_A l_{B,A} = m_{B,A}\sigma_B l_{A,B} l_{B,A} \quad (4.3.6)$$

for every $A, B \in \Omega$. Exchanging A and B gives

$$\sigma_B l_{A,B} = m_{A,B}\sigma_A l_{B,A} l_{A,B} \quad (4.3.7)$$

for every $A, B \in \Omega$. Finally, combine these equations:

$$\begin{aligned} m_{A,B}\sigma_A &= m_{A,B}m_{B,A}\sigma_B l_{A,B} && \text{by (4.3.5)} \\ &= m_{A,B}m_{B,A}\sigma_B l_{A,B} l_{B,A} l_{A,B} && \text{since } l_{A,B} \text{ is a partial isometry} \\ &= m_{A,B}\sigma_A l_{B,A} l_{A,B} && \text{by (4.3.6)} \\ &= \sigma_B l_{A,B} && \text{by (4.3.7)} \end{aligned}$$

This completes the proof. \square

Corollary 4.3.8. *In the situation of Lemma 4.3.7, the unique morphism $f: L \rightarrow M$ satisfying*

$$\begin{array}{ccc} L & \overset{f}{\dashrightarrow} & M \\ l_A \downarrow & & \downarrow m_A \\ D(A) & \xrightarrow{\sigma_A} & E(A) \end{array}$$

also makes the following diagram commute.

$$\begin{array}{ccc} L & \xrightarrow{f} & M \\ l_A^\dagger \uparrow & & \uparrow m_A^\dagger \\ D(A) & \xrightarrow{\sigma_A} & E(A) \end{array} \quad (4.3.8)$$

In other words, the map of limits $L \rightarrow M$ induced by σ coincides with the map of colimits $L \rightarrow M$ induced by $\sigma: \dagger \circ D \rightarrow \dagger \circ E$.

Proof. To show that f makes diagram (4.3.8) commute, it suffices to postcompose with an arbitrary m_B and show that the following diagram commutes.

$$\begin{array}{ccccc}
 D(A) & \xrightarrow{\sigma_A} & E(A) & \xrightarrow{m_A^\dagger} & M \\
 l_A^\dagger \downarrow & & & & \downarrow m_B \\
 L & \xrightarrow{l_B} & D(B) & \xrightarrow{\sigma_B} & E(B) \\
 & \searrow f & & \nearrow m_B & \\
 & & M & &
 \end{array}$$

The bottom part commutes by definition of f , and the top rectangle by Lemma 4.3.7. \square

We can now prove Theorem 4.3.1.

Proof of Theorem 4.3.1. If the map of cones $L \rightarrow M$ is unitary, it is straightforward to verify that M is a dagger limit of (D, Ω) . For the converse, by Theorem 4.3.6 it suffices to prove that the diagram

$$\begin{array}{ccc}
 D(A) & \xrightarrow{l_A^\dagger} & L \\
 m_A^\dagger \downarrow & & \downarrow l_B \\
 M & \xrightarrow{m_B} & D(B)
 \end{array}$$

commutes for any A and B in \mathbf{J} . This follows from Lemma 4.3.7 by choosing $\sigma = \text{id}_D$, which is clearly adjointable. \square

Note that the preceding proof only used Lemma 4.3.7 in the special case of $\sigma = \text{id}$. This special case is not sufficient for the sequel and indeed the general versions of Lemma 4.3.7 and of Corollary 4.3.8 are used later on, e.g. in Sections 4.5 and 4.8.

4.4 Completeness

Perhaps the most obvious definitions of dagger completeness would be “every small diagram has a dagger limit for some/all possible weakly initial Ω ”. However, here dagger category theory deviates from ordinary category theory. Such a definition would be too strong to allow interesting models, as the following theorems show.

Theorem 4.4.1. *If a dagger category has dagger equalizers, dagger pullbacks and finite dagger products, then it must be indiscrete.*

Theorem 4.4.2. *If a dagger category has dagger equalizers and infinite dagger products, then it must be indiscrete.*

In proving these degeneration theorems, it is useful to isolate some lemmas that derive key properties implied by having dagger equalizers, dagger pullbacks and finite dagger products.

Lemma 4.4.3. *The dagger pullback of a morphism $f: A \rightarrow B$ along id_B exists if and only if f is a partial isometry. In particular, if a dagger category has all dagger pullbacks, then every morphism is a partial isometry.*

Proof. If f is a partial isometry, then A is a dagger pullback of f along id_B , with cone id_A and f . For the converse, let P be a dagger pullback of f along id_B , with cone $p_A: P \rightarrow A$ and $p_B: P \rightarrow B$. Since A is also a pullback, there exists a unique isomorphism $g: A \rightarrow P$ making the diagram

$$\begin{array}{ccccc}
 A & & & & \\
 \downarrow \text{id} & \searrow g & & \xrightarrow{f} & \\
 & P & \xrightarrow{p_B} & & B \\
 & \downarrow p_A & & & \downarrow \text{id} \\
 & A & \xrightarrow{f} & & B
 \end{array}$$

commute. Since g is an isomorphism, this implies that $p_A = g^{-1}$. Because p_A is a partial isometry, both p_A and g are therefore unitary. Now $f = p_B g$ means that f factors as the composite of a unitary morphism and a partial isometry and hence is a partial isometry itself. \square

It follows from the previous lemma that in a dagger category with dagger pullbacks, every subobject is a dagger subobject since every monomorphism is dagger monic.

Recall that a monoid $(M, +, 0)$ is *cancellative* when $x + z = y + z$ implies $y = z$.

Lemma 4.4.4. *If a dagger category has finite dagger products, then it is uniquely enriched in commutative monoids and admits a matrix calculus. If it furthermore has dagger equalizers, then addition is cancellative.*

Proof. Dagger products are in particular biproducts, and it is well known that biproducts make a category uniquely semiadditive; see e.g. [115, 18.4], [148, 2.4], or [69, 1.1]. For the claim that dagger equalizers make addition cancellative, we refer to [148, 2.6]. \square

We can now prove the theorems stated in the beginning of this section.

Proof of Theorem 4.4.1. Let $f: A \rightarrow B$; we will prove that $f = 0_{A,B}$. By Lemma 4.4.3, the tuple $\langle \text{id}, f \rangle: A \rightarrow A \oplus B$ is a partial isometry. Expanding this fact using the matrix calculus results in the equations $\text{id} + f^\dagger f = \text{id}$ and $f = f + f$. Applying cancellativity of addition to the latter equation now gives $f = 0$. \square

Proof of Theorem 4.4.2. Let $f: A \rightarrow B$; we will prove that $f = 0_{A,B}$. Observe that in Lemma 4.4.4, infinite dagger products induce the ability to add infinitely many parallel morphisms in the same way as binary dagger products enable binary addition: if $f_i: A \rightarrow B$ is an I -indexed family of morphisms and \mathbf{C} has $|I|$ -ary dagger products, the sum $\sum f_i$ can be defined as the composite $\Delta_B^\dagger(\bigoplus_i f_i)\Delta_A: A \rightarrow \bigoplus_i A \rightarrow \bigoplus_i B \rightarrow B$, where $\Delta_A: A \rightarrow \bigoplus_i A$ is the diagonal map. This implies that the category is enriched in Σ -monoids [62]. Hence the following variant of the Eilenberg swindle makes sense.

$$\begin{aligned} 0 + (f + f + \dots) &= f + f + \dots \\ &= f + (f + f + \dots) \end{aligned}$$

It now follows from cancellativity (of binary addition) that $f = 0$. \square

Note that both theorems rest on an interplay between various dagger limits. Indeed, none of the dagger limits in question are problematic on their own: **Hilb** has dagger equalizers and finite dagger products, **Rel** has arbitrary dagger products, and **Pinj** has dagger pullbacks.

Given Theorems 4.4.1 and 4.4.2, dagger completeness is not an interesting concept. However, below we define a (finite) completeness condition for dagger categories, and show that it is equivalent to the existence of dagger products, dagger equalizers, and dagger intersections.

Definition 4.4.5. A class Ω of objects of a category \mathbf{J} is called a *basis* when every object B allows a unique $A \in \Omega$ making $\mathbf{J}(A, B)$ non-empty. The category \mathbf{J} is called *based* when there exists a basis, and *finitely based* when there exists a finite basis. We say a dagger category \mathbf{C} has *(finitely) based dagger limits*, or that it is *(finitely)*

based dagger complete if for every category \mathbf{J} with a (finite) basis Ω , and any diagram $D: \mathbf{J} \rightarrow \mathbf{C}$ the dagger limit of (D, Ω) exists.

Example 4.4.6. The shape $\bullet \rightrightarrows \bullet$, giving rise to equalizers, is finitely based. Any (finite) discrete category, the shape giving rise to (finite) products, is (finitely) based. Any indiscrete category is also finitely based. A category is (finitely) based if and only if it is the (finite) disjoint union of categories, each having a weakly initial object. In particular, any dagger category is based and is finitely based iff it has finitely many connected components.

Since both dagger equalizers and arbitrary dagger products exist in a based dagger complete category, all such categories are indiscrete by Theorem 4.4.2 and hence the notion is uninteresting. However, being finitely (based) complete need not trivialize the category as seen below in Corollary 4.4.9.

Example 4.4.7. Finitely based diagrams need not be finite. For example, given any family $f_i: A \rightarrow B$ of parallel arrows, their joint dagger equalizer is the dagger limit of a finitely based diagram. Similarly, a dagger intersection is the dagger limit of a diagram of dagger monomorphisms $m_i: A_i \rightarrow B$, i.e. a wide (dagger) pullback. Whenever there are at least two monomorphisms, this diagram is not a based category. However, we may also obtain the dagger intersection as the dagger limit of the diagram consisting of $m_i m_i^\dagger: B \rightarrow B$, i.e. as the limit of the induced projections on B . Then we may always take Ω to be a singleton. Hence being finitely based dagger complete is a stronger requirement than being finitely complete.

Theorem 4.4.8. *A dagger category is finitely based dagger complete if and only if it has dagger equalizers, dagger intersections and finite dagger products.*

Proof. One direction is obvious, because the shapes of dagger equalizers, dagger intersections, and finite dagger products are all finitely based. For the converse, assume that a dagger category \mathbf{C} has dagger equalizers, dagger intersections and finite dagger products. Let \mathbf{J} have a finite basis Ω . For a given diagram $D: \mathbf{J} \rightarrow \mathbf{C}$, we will construct a dagger limit of (D, Ω) . Fix $A \in \Omega$. For every parallel pair $f, g: A \rightarrow B$ of morphisms in \mathbf{J} , pick a dagger equalizer $e_{f,g}: E_{f,g} \rightarrow D(A)$ of $D(f)$ and $D(g)$. Pick a dagger intersection $m_A: L_A \rightarrow D(A)$ of all $e_{f,g}$. Pick a dagger product L of L_A for all $A \in \Omega$, and write $p_A: L \rightarrow L_A$ for the product cone. Given a morphism $f: A \rightarrow B$ in \mathbf{J} with $A \in \Omega$, define l_A as the composite $D(f)m_A p_A: L \rightarrow L_A \rightarrow D(A) \rightarrow D(B)$. Since Ω is a basis, the choice of A is forced on us. By construction l_A is independent of the choice

of $f: A \rightarrow B$. It is easy to see that $l_A: L \rightarrow D(A)$ is a limiting cone, so it remains to check the normalization and independence axioms for Ω . Given $A \in \Omega$,

$$\begin{aligned}
l_A l_A^\dagger l_A &= m_A p_A p_A^\dagger m_A^\dagger m_A p_A \\
&= m_A p_A p_A^\dagger p_A && \text{since } m_A \text{ is an isometry} \\
&= m_A p_A && \text{since } p_A \text{ is a partial isometry} \\
&= l_A.
\end{aligned}$$

Moreover, since $p_A: L \rightarrow D(A)$ forms a product, $l_A^\dagger l_A l_B^\dagger l_B = 0_L = l_B^\dagger l_B l_A^\dagger l_A$ if $A, B \in \Omega$. \square

Corollary 4.4.9. *The categories **FHilb** and **Hilb** are finitely based dagger complete.*

Corollary 4.4.10. *Both categories have dagger equalizers and dagger products. To see that dagger intersections exist, note that a dagger monic corresponds to an inclusion of a closed subspace, and hence the intersection can be computed as the set-theoretic intersection of the corresponding closed subspaces, which is still closed. Hence the result follows from Theorem 4.4.8.*

The following theorem characterizes dagger categories having dagger-shaped limits similarly.

Theorem 4.4.11. *Let κ be a cardinal number. A dagger category has dagger-shaped limits of shapes with at most κ many connected components if and only if it has dagger split infima of projections, dagger stabilizers, and dagger products of at most κ many objects.*

Proof. One implication is trivial. For the other, we first reduce to the case of a single connected component. Given $D: \mathbf{J} \rightarrow \mathbf{C}$, with connected components \mathbf{J}_i , assume that we can use dagger limits of projections and dagger stabilizers to build a dagger limit L_i of each restriction D_i . Then we can define the dagger limit of D by taking the dagger product of all the L_i . For L is a limit of D , as each L_i is a limit of D_i . Moreover, since $L_i \rightarrow D_i(A)$ is a partial isometry, $L_i \rightarrow D_i(A) \rightarrow L_i$ is the identity. Therefore each $L \rightarrow L_i \rightarrow D(A)$ is a partial isometry. If A and B are in the same connected component of \mathbf{J} , then $l_A^\dagger l_A = l_B^\dagger l_B$. If A and B are in different connected components, then $l_B l_A = 0_{D_A, D_B}$. Hence the independence equations are satisfied, and L indeed is the dagger limit of D .

Now focus on the case when \mathbf{J} is connected. Pick any object A of \mathbf{J} . For every $f, g: A \rightarrow B$, take the dagger stabilizer $E_{f,g} \rightarrow D(A)$ of $D(f)$ and $D(g)$, and let $p_{f,g}$

be the induced projection $D(A) \rightarrow E_{f,g} \rightarrow D(A)$ on $D(A)$. Let \mathcal{P} be the set of all such projections, and let $l: L \rightarrow D(A)$ be the limit. For each B in \mathbf{J} , choose a morphism $f: A \rightarrow B$, and define $l_B: L \rightarrow D(B)$ as $D(f) \circ l$. By construction, l_B is independent of the choice of f . We claim that this makes L into a dagger limit of D . First, to see it is a cone, take an arbitrary $g: B \rightarrow C$. Then $D(g)l_B = D(g)D(f)l_A = D(gf)l_A = l_C$, as desired. Because $L \rightarrow D(A)$ is a partial isometry, so is each $L \rightarrow D(B)$, as in Example 4.2.3. Moreover, $l_A^\dagger l_A = l_B^\dagger l_B$ when $A, B \in \mathbf{J}$. Finally, to show that L is indeed a limit, take an arbitrary cone $m_B: M \rightarrow D(B)$. As it is a cone, it is uniquely determined by m_A . But m_A must factor through each stabilizer, so m_A is also a cone for \mathcal{P} . The universal property of L ensures that m_A factors through l_A . \square

Corollary 4.4.12. *The dagger category \mathbf{Rel} has all dagger-shaped limits, and the dagger categories \mathbf{FinRel} and $\mathbf{Span}(\mathbf{FinSet})$ have all dagger-shaped limits with finitely many connected components.*

Proof. Observe that \mathbf{Rel} has all (small) dagger products, and that similarly \mathbf{FinRel} and $\mathbf{Span}(\mathbf{FinSet})$ have finite dagger products. Combine Proposition 4.2.5 and Theorem 4.4.11. \square

We end this section by comparing our notion of finite dagger completeness to that of [148]. Thus we must restrict to categories enriched in commutative monoids. This simplifies things, as in the following lemma.

Proposition 4.4.13. *Let Ω be a basis for \mathbf{J} and $D: \mathbf{J} \rightarrow \mathbf{C}$ a diagram in a category with zero morphisms. A limit $l_A: L \rightarrow D(A)$ of D is a dagger limit of (D, Ω) if and only if:*

- l_A is a partial isometry whenever $A \in \Omega$; and
- $l_B l_A^\dagger = 0_{A,B}$ whenever $A, B \in \Omega$ and $A \neq B$.

Proof. Any limit satisfying the above conditions also satisfies normalization. For independence, the second condition gives $l_A^\dagger l_A l_B^\dagger l_B = 0_{L,L} = l_B^\dagger l_B l_A^\dagger l_A$.

Conversely, given a dagger limit $l_A: L \rightarrow D(A)$ of (D, Ω) , we wish to show that $l_B l_A^\dagger = 0_{A,B}$ whenever $A, B \in \Omega$ are distinct. Fix distinct $A, B \in \Omega$ and define a natural transformation $\sigma: D \Rightarrow D$ by

$$\sigma_C = \begin{cases} \text{id}_{D(C)} & \text{if } \mathbf{J}(A, C) \neq \emptyset \\ 0_{D(C), D(C)} & \text{otherwise.} \end{cases}$$

By construction σ is adjointable. Hence Lemma 4.3.7 guarantees $l_B l_A^\dagger = l_B l_A^\dagger \text{id} = l_B l_A^\dagger \sigma_A = \sigma_B l_B l_A^\dagger = 0_{l_B l_A^\dagger} = 0$. \square

Proposition 4.4.14. *Let Ω be a finite basis for \mathbf{J} and $D: \mathbf{J} \rightarrow \mathbf{C}$ a diagram in a dagger category \mathbf{C} enriched in commutative monoids. A limit $l_A: L \rightarrow D(A)$ is a dagger limit of (D, Ω) if and only if $\text{id}_L = \sum_{A \in \Omega} l_A^\dagger l_A$.*

Proof. Assume first that L is a dagger limit of (D, Ω) . Since \mathbf{C} is enriched in commutative monoids, the composition $\mathbf{C}(A, B) \otimes \mathbf{C}(B, C) \rightarrow \mathbf{C}(A, C)$ is bilinear by definition of tensor product of commutative monoids, and so the unit morphisms of addition are zero morphisms. Therefore $l_B l_A^\dagger = 0$ by Proposition 4.4.13 when $A, B \in \Omega$ are distinct. Hence

$$l_B \sum_{A \in \Omega} l_A^\dagger l_A = l_B l_B^\dagger l_B + l_B \sum_{B \neq A \in \Omega} l_A l_A = l_B + 0 = l_B$$

for every $B \in \Omega$, which implies $\text{id}_L = \sum_{A \in \Omega} l_A^\dagger l_A$.

For the converse, assume that L is a limit of D satisfying $\text{id}_L = \sum_{A \in \Omega} l_A^\dagger l_A$. Fix $A \in \Omega$ and define $\sigma: D \Rightarrow D$ as in the proof of Proposition 4.4.13. Now σ induces a unique morphism $f: L \rightarrow L$ making the following square commute for any C in \mathbf{J} :

$$\begin{array}{ccc} L & \xrightarrow{f} & L \\ l_C \downarrow & & \downarrow l_C \\ D(C) & \xrightarrow{\sigma_C} & D(C) \end{array}$$

Now $f = \text{id}_L f = (\sum_{B \in \Omega} l_B^\dagger l_B) f = \sum_{B \in \Omega} l_B^\dagger l_B f = \sum_{B \in \Omega} l_B^\dagger \sigma_B l_B = l_A^\dagger l_A + 0 = l_A^\dagger l_A$. It follows that

$$l_B l_A^\dagger l_A = 0 \tag{4.4.1}$$

for distinct $A, B \in \Omega$, so that $l_A = l_A \text{id}_L = l_A \sum_{A \in \Omega} l_A^\dagger l_A = l_A l_A^\dagger l_A + 0 = l_A l_A^\dagger l_A$ and each l_A is a partial isometry. Finally (4.4.1) implies $l_B l_A^\dagger = 0_{A, B}$ for distinct A, B , making L into a dagger limit of (D, Ω) by Proposition 4.4.13. \square

Thus whenever \mathbf{J} has a finite basis and \mathbf{C} is appropriately enriched, Definition 4.4.5 coincides with the notion of completeness in [148]. But our notion is not more general: when the diagram does not admit a finite basis, the two notions are different. For instance, **FHilb** does not have all dagger pullbacks, because not all morphisms are partial isomorphisms. But for every diagram $A \rightarrow C \leftarrow B$ there exists a pullback $A \leftarrow L \rightarrow B$ with $\text{id}_L = l_A^\dagger l_A + l_B^\dagger l_B$. Conversely, the following example below exhibits a dagger pullback in **Rel** that does not satisfy such a summation.

Example 4.4.15. Consider the objects $A = \{1, 2\}$ and $B = \{1\}$ in **Rel**, and morphisms $R = A \times A: A \rightarrow A$ and $S = B \times A: B \rightarrow A$. Then the object A with the cone morphisms

$l_A = \text{id}_A: A \rightarrow A$ and $l_B = A \times B: A \rightarrow B$ form a pullback of R and S . Now both l_A and l_B are partial isometries, and $l_A^\dagger l_A l_B^\dagger l_B = l_B^\dagger l_B l_A^\dagger l_A$, so this is a dagger pullback. But $l_A^\dagger l_A + l_B^\dagger l_B = A \times A \neq \text{id}_A$.

4.5 Global dagger limits

If limits happen to exist for all diagrams of a fixed shape, it is well known that they can also be formulated as an adjoint to the constant functor. This section explores this phenomenon of global limits in the dagger setting.

More precisely, fix a shape \mathbf{J} and a weakly initial Ω . Assume that a dagger category \mathbf{C} has (\mathbf{J}, Ω) -shaped dagger limits. Then there is an induced right adjoint L to the diagonal functor $\Delta: \mathbf{C} \rightarrow \mathbf{Cat}(\mathbf{J}, \mathbf{C})$. It restricts to a dagger functor $\hat{L}: [\mathbf{J}, \mathbf{C}]_\dagger \rightarrow \mathbf{C}$: $L(\sigma^\dagger) = L(\sigma)^\dagger$ for any $D, E: \mathbf{J} \rightarrow \mathbf{C}$ and adjointable $\sigma: D \Rightarrow E$. This follows from Corollary 4.3.8, since $L(\sigma^\dagger)$ is the map of limits induced by $\sigma^\dagger: E \rightarrow D$, and $L(\sigma)^\dagger$ is the map of colimits induced by $\sigma^\dagger: \dagger \circ E \rightarrow \dagger \circ D$. Normalization and independence endow the adjunction $\Delta \dashv L$ with further properties.

Normalization Consider the counit $\varepsilon: \Delta L \rightarrow \text{id}_{\mathbf{Cat}(\mathbf{J}, \mathbf{C})}$. By construction, for any diagram $D: \mathbf{J} \rightarrow \mathbf{C}$, each $(\varepsilon_D)_A$ is a partial isometry whenever $A \in \Omega$. “The counit is a partial isometry when restricted to Ω ”.

Independence Write d_A for the component of $\varepsilon_D: \Delta L(D) \rightarrow D$ at A , and e_A for the component of ε_E at A . For a fixed $A \in \Omega$, define a natural transformation $\rho_A: \hat{L} \rightarrow \hat{L}$ by setting $(\rho_A)_D = d_A^\dagger d_A: L(D) \rightarrow D(A) \rightarrow L(D)$. To see that this forms a natural transformation $\hat{L} \Rightarrow \hat{L}$, let $\sigma: D \Rightarrow E$ be adjointable and consider the following diagram.

$$\begin{array}{ccccc}
 L(D) & \xrightarrow{d_B} & D(A) & \xrightarrow{d_A^\dagger} & L(D) \\
 L(\sigma) \downarrow & & \sigma_A \downarrow & & \downarrow L(\sigma) \\
 L(E) & \xrightarrow{e_A} & E(A) & \xrightarrow{e_A^\dagger} & L(E)
 \end{array}$$

The left square commutes by definition of $L(\sigma)$, and the right one by definition of $L(\sigma^\dagger)^\dagger = L(\sigma)$, so that the rectangle expressing naturality of ρ_A commutes. Clearly, $\rho_A \rho_B = \rho_B \rho_A$ whenever $A, B \in \Omega$.

These properties in fact characterize dagger categories \mathbf{C} having all (\mathbf{J}, Ω) -shaped limits.

Theorem 4.5.1. *A dagger category \mathbf{C} has all (\mathbf{J}, Ω) -shaped limits if and only if the diagonal functor $\Delta: \mathbf{C} \rightarrow \mathbf{Cat}(\mathbf{J}, \mathbf{C})$ has a right adjoint L such that:*

- *the counit is a partial isometry when restricted to Ω ;*
- *L restricts to a dagger functor $\hat{L}: [\mathbf{J}, \mathbf{C}]_{\dagger} \rightarrow \mathbf{C}$;*
- *the family $\rho(A)_D = d_A^{\dagger} d_A$ is natural $\hat{L} \rightarrow \hat{L}$ for any $A \in \Omega$;*
- *if $A, B \in \Omega$, then $\rho_A \rho_B = \rho_B \rho_A$.*

Proof. We already proved the implication from left to right above. The other implication is straightforward: it is well-known that if L is the right adjoint to the diagonal then LD is a limit of D for each D . Hence we only need to check normalization and independence on Ω . The counit being a partial isometry when restricted to Ω means that for each $A \in \Omega$ the structure map $LD \rightarrow DA$ of the limit is a partial isometry, giving us normalization. The condition $\rho_A \rho_B = \rho_B \rho_A$ for $A, B \in \Omega$ then amounts to independence. \square

We leave open the question whether the fourth condition of Theorem 4.5.1 is necessary in general.

Further restricting the shape \mathbf{J} yields a cleaner special cases of the previous theorem.

Theorem 4.5.2. *Let Ω be a basis for \mathbf{J} . A dagger category \mathbf{C} has all (\mathbf{J}, Ω) -shaped limits if and only if the diagonal functor $\Delta: \mathbf{C} \rightarrow \mathbf{Cat}(\mathbf{J}, \mathbf{C})$ has a right adjoint L such that:*

- *the counit is a partial isometry when restricted to Ω ;*
- *L restricts to a dagger functor $\hat{L}: [\mathbf{J}, \mathbf{C}]_{\dagger} \rightarrow \mathbf{C}$.*

Proof. The implication from left to right follows from the previous theorem. For the other direction, the conditions imply that a diagram D has a limit given by $d_A: L(D) \rightarrow D(A)$, and furthermore d_A is a partial isometry if $A \in \Omega$. It remains to verify that $d_A^{\dagger} d_A$ commutes with $d_B^{\dagger} d_B$ when $A, B \in \Omega$. We may assume that Ω has at least two distinct objects. We will show that \mathbf{C} has zero morphisms and $d_B d_A^{\dagger} = 0_{D(A), D(B)}$ whenever $A, B \in \Omega$ are distinct.

Fix distinct $A, B \in \Omega$ and $X_0 \in \mathbf{C}$. For objects X, Y of \mathbf{C} , define $D_{X,Y}: \mathbf{J} \rightarrow \mathbf{C}$ by

$$D_{X,Y}(C) = \begin{cases} X & \text{if } \mathbf{J}(A, C) \neq \emptyset, \\ Y & \text{if } \mathbf{J}(B, C) \neq \emptyset, \\ X_0 & \text{otherwise,} \end{cases}$$

and mapping morphisms to id_X , id_Y or id_{X_0} as appropriate. Define $0_{X,Y}$ as the composite $X = D_{X,Y}(A) \rightarrow L(D_{X,Y}) \rightarrow D_{X,Y}(B) = Y$. Let $f: Y \rightarrow Z$ be arbitrary. This induces a natural transformation $\sigma: D_{X,Y} \Rightarrow D_{X,Z}$ such that σ_C is f if $\mathbf{J}(B, C) \neq \emptyset$ and $\sigma_C = \text{id}$ otherwise. It is clearly adjointable, so the following diagram commutes.

$$\begin{array}{ccccc} D_{X,Y}(A) = X & \longrightarrow & LD_{X,Y} & \longrightarrow & D_{X,Y}(B) = Y \\ \sigma_A = \text{id} \downarrow & & \downarrow L\sigma = (L\sigma^\dagger)^\dagger & & \downarrow \sigma_B = f \\ D_{X,Z}(A) = X & \longrightarrow & LD_{X,X} & \longrightarrow & D_{X,Z}(B) = Z \end{array}$$

Hence $f \circ 0_{X,Y} = 0_{X,Z}$. Taking daggers shows that $0_{Y,X}f^\dagger = 0_{Z,Y}$ for all X, Y and $f: Y \rightarrow Z$. Thus \mathbf{C} indeed has zero morphisms.

Finally, given an arbitrary $D: \mathbf{J} \rightarrow \mathbf{D}$, we will show that $d_{A,B} = 0_{D(A),D(B)}$ for distinct A, B . Define $\sigma: D \Rightarrow D$ by

$$\sigma_C = \begin{cases} 0_{D(C),D(C)} & \text{if } \mathbf{J}(A, C) \neq \emptyset, \\ \text{id} & \text{otherwise.} \end{cases}$$

As σ is adjointable, the following diagram commutes.

$$\begin{array}{ccccc} D(A) & \xrightarrow{d_A^\dagger} & LD & \xrightarrow{d_B} & D(B) \\ \sigma_A = 0 \downarrow & & \downarrow L\sigma = (L(\sigma)^\dagger)^\dagger & & \downarrow \sigma_B = \text{id} \\ D(A) = X & \xrightarrow{d_A^\dagger} & LD & \xrightarrow{d_B} & D(B) \end{array}$$

Thus $d_{A,B} = 0_{D(A),D(B)}$. □

Theorem 4.5.3. *Let \mathbf{J} be a dagger category. A dagger category \mathbf{C} has a dagger limit for every dagger functor $\mathbf{J} \rightarrow \mathbf{C}$ if and only if the diagonal functor $\Delta: \mathbf{C} \rightarrow \mathbf{Dagcat}(\mathbf{J}, \mathbf{C})$ has a dagger adjoint such that the counit is a partial isometry.*

Proof. The implication from left to right is straightforward once one remembers from Example 4.2.3 that for any dagger limit $(L, \{l_A\}_{A \in \mathbf{J}})$ of a dagger-shaped diagram, every l_A is a partial isometry.

For the other direction, the dagger adjoint to the diagonal clearly gives a limit for each dagger functor $\mathbf{J} \rightarrow \mathbf{C}$. It remains to verify that they are all dagger limits. The counit being a partial isometry implies the normalization condition, so it suffices to check independence. If \mathbf{J} is connected, this is trivial. If \mathbf{J} is not connected, then, as in the previous proof, \mathbf{C} has zero morphisms and for any diagram D we have $d_{A,B} = 0_{D(A),D(B)}$ whenever A and B are in different components of \mathbf{J} . \square

4.6 Dagger adjoint functors

When dealing with dagger limits and dagger functors preserving them, the obvious question arises when there exists a dagger adjoint. That is, is there a dagger version of the adjoint functor theorem?

Simply replacing limits with dagger limits in any standard proof of the adjoint functor theorem [115] doesn't quite get there. If \mathbf{C} has all dagger products and dagger equalizers, it must be indiscrete by Theorem 4.4.2. Hence any continuous functor $\mathbf{C} \rightarrow \mathbf{D}$ satisfying the solution set condition vacuously has an adjoint. A more interesting dagger adjoint functor theorem must therefore work with a finitely based complete category \mathbf{C} and solution sets of a finite character.

There is a further obstacle. Ordinarily, the adjoint functor theorem shows that its assumptions imply a universal arrow $\eta_A: A \rightarrow GF(A)$ for each object A , so that the desired adjoint is given by $A \mapsto F(A)$. This will not do for dagger categories, as the resulting functor F need not preserve the dagger. This is essentially because universal arrows need not be defined up to unitary isomorphism. For an example, consider the identity functor $G: \mathbf{FHilb} \rightarrow \mathbf{FHilb}$, and define $\eta_A: A \rightarrow A$ to be multiplication by $1 + \dim A$. Then each η_A is an universal arrow for G , defining an adjoint $F: \mathbf{FHilb} \rightarrow \mathbf{FHilb}$ that sends $f: A \rightarrow B$ to $f(1 + \dim B)/(1 + \dim A)$, which is not a dagger functor. Of course, this example evaporates by choosing the 'correct' universal arrows. But there are more involved examples of dagger functors admitting adjoints but no dagger adjoints.

The moral is that a dagger adjoint to $G: \mathbf{C} \rightarrow \mathbf{D}$ requires more than G -universal arrows $A \rightarrow GF(A)$ for each object. The universal arrows must fit together, in the sense that they form an adjointable natural transformation. Unfortunately, we do not know of conditions on solution sets guaranteeing this. As a first step towards a dagger adjoint functor theorem proper we provide the following theorem.

Theorem 4.6.1. *Suppose a dagger category \mathbf{C} has, and a dagger functor $G: \mathbf{C} \rightarrow \mathbf{D}$ preserves, dagger intersections and dagger equalizers. Then G has a dagger adjoint if and only if there is a dagger functor $H: \mathbf{D} \rightarrow \mathbf{C}$ and a natural transformation $\tau: \text{id}_{\mathbf{D}} \rightarrow GH$ such that each component of τ is weakly G -universal.*

Proof. One implication is trivial. For the other, define $F(A)$ to be a dagger intersection of all dagger monomorphisms $m: M \rightarrow H(A)$ for which τ_A factorizes through $G(m)$. As G preserves dagger intersections, τ_A factorizes via $GF(A) \rightarrow GH(A)$, say $\tau_A = G(\sigma_A)\eta_A$. It suffices to prove that (i) η_A is G -universal, so that $A \mapsto FA$ extends uniquely to a functor, and then that (ii) F is a dagger functor.

First of all, η_A is weakly G -universal since τ_A is so: given $f: A \rightarrow G(X)$, pick h such that $f = G(h)\tau_A = G(h\sigma_A)\eta_A$. Moreover, if h and h' satisfied $G(h)\eta_A = G(h')\eta_A$, consider the equalizer $e: E \rightarrow F(A)$ of h and h' . By assumption, η_A factors through $G(e)$ and hence τ_A factors through $G(\sigma_A e)$, so that $\sigma_A e$ is already in the dagger intersection defining $F(A)$. In other words, e is unitary, and hence $h = h'$. Thus η_A is G -universal, and we can extend $A \mapsto F(A)$ to a functor by defining $F(f)$ as the unique map making the following square commute.

$$\begin{array}{ccc} A & \xrightarrow{\eta_A} & GFA \\ f \downarrow & & \downarrow GF(f) \\ B & \xrightarrow{\eta_B} & GFB \end{array}$$

Next we show that F is a dagger functor. By Lemma 2.1.15 it suffices to show that σ is a natural transformation $F \rightarrow H$. By naturality of τ , the top part of

$$\begin{array}{ccccccc} & & & & G(\sigma_A) & & \\ & & & & \xrightarrow{\quad} & & \\ & & & & GFA & \xrightarrow{\quad} & GHA \\ \eta_A \uparrow & & & & & & \downarrow GH(f) \\ A & \xrightarrow{f} & B & \xrightarrow{\eta_B} & GFB & \xrightarrow{G(\sigma_B)} & GHB \\ & \searrow \eta_A & & \nearrow GFf & & & \\ & & & & GFA & & \end{array}$$

commutes, whereas the bottom part commutes by naturality of η . As η_A is G -universal,

we conclude that the square

$$\begin{array}{ccc}
 FA & \xrightarrow{\sigma_A} & HA \\
 F(f) \downarrow & & \downarrow H(f) \\
 FB & \xrightarrow{\sigma_B} & HB
 \end{array}$$

commutes, making $\sigma: F \Rightarrow H$ natural. □

4.7 Polar decomposition

Polar decomposition, as standardly understood, provides a way to factor any bounded linear map between Hilbert spaces into a partial isometry and a positive morphism [63, Chapter 16]. As dagger limits are defined in terms of partial isometries, one might hope that polar decomposition connects dagger limits and ordinary limits. This section explores this connection. We start by defining polar decomposition abstractly, and prove that this modified property holds in the category of Hilbert spaces. Recall that a bimorphism in a category is a morphism that is both monic and epic.

Definition 4.7.1. Let $f: A \rightarrow B$ be a morphism in a dagger category. A *polar decomposition* of f consists of two factorizations of f as $f = pi = jp$,

$$\begin{array}{ccc}
 A & \xrightarrow{i} & A \\
 p \downarrow & \searrow f & \downarrow p \\
 B & \xrightarrow{j} & B
 \end{array}$$

where p is a partial isometry and i and j are self-adjoint bimorphisms. A category *admits polar decomposition* when every morphism has a polar decomposition.

Note that if f is monic then so is p : indeed, if $pg = ph$ then $fg = jpg = jph = fh$, whence $g = h$. Hence if f is monic then p is dagger monic since it is both monic and a partial isometry.

Unlike usual expositions of polar decomposition, we require i and j to be bimorphisms. On the other hand, we don't require them to be positive since mere self-adjointness suffices for our purposes – most notably Theorems 4.7.7 and 4.7.10. However, for other purposes this definition might need to be modified and hence it should be only seen as a starting point for an abstract notion of polar decomposition.

We begin by proving that **Hilb** admits polar decomposition in the above sense. Given that our notion of polar decomposition is slightly different from the usual one (merely, a single factorization as $f = pi$ with p a partial isometry and i positive), there is some work to do. First, recall that if f is a bounded linear map, i.e. a morphism of **Hilb**, then $|f|$ is the (unique) positive square root of $f^\dagger f$.

Lemma 4.7.2. *For any morphism f in **Hilb** we have $|f^\dagger|f = f|f|$.*

Proof. First we show that it suffices to prove the claim for non-expansive f , i.e. we may assume that $\|f\| \leq 1$. Assuming the claim holds for non-expansive maps, take f with $\|f\| > 1$. Define $g := f/\|f\|$. Now $|g| = |f|/\|f\|$ since positive square roots are unique and both sides square to $f^\dagger f/\|f\|^2$. Hence $|g^\dagger|g = g|g|$ amounts to saying that

$$|f^\dagger|f/\|f\|^2 = f|f|/\|f\|^2$$

so that multiplying by $\|f\|^2$ gives the result for f . Hence we may assume that f is non-expansive.

By (the proof of) [15, Theorem 23.2], the square root of a positive non-expansive operator p is the strong limit of the sequence $(p_n)_{n \in \mathbb{N}}$, where $p_0 = 0$ and $p_{n+1} := p_n - (p - p_n^2)/2$. Applying this result to $f^\dagger f$ we see that $|f|$ is the strong limit of the sequence $(q_n)_{n \in \mathbb{N}}$ where $q_0 = 0$ and $q_{n+1} := q_n - (f^\dagger f - q_n^2)/2$. Similarly, $|f^\dagger|$ is the strong limit of the sequence $(r_n)_{n \in \mathbb{N}}$ defined by $r_0 = 0$ and $r_{n+1} := r_n - (f f^\dagger - r_n^2)/2$. Since precomposing (and postcomposing) with a non-expansive map is continuous (in the strong operator topology), to prove that $|f^\dagger|f = f|f|$ it suffices to prove that $r_n f = f q_n$, which we do by induction. For $n = 0$, both sides evaluate to 0. Assuming the claim holds for n we see that it holds for $n + 1$, since

$$\begin{aligned} r_{n+1}f &= (r_n - (f f^\dagger - r_n^2)/2)f \\ &= f q_n - (f f^\dagger f - f q_n^2)/2 \text{ by the induction hypothesis} \\ &= f(q_n - (f^\dagger f - q_n^2)/2) = f q_{n+1} \end{aligned}$$

completing the proof. □

Theorem 4.7.3. *The category **Hilb** admits polar decomposition.*

Proof. We modify the standard construction of a polar decomposition [63, Problem 134] to satisfy Definition 4.7.1: the standard construction gives a factorization of $f: H \rightarrow K$ into $f = p|f|$, where p is a partial isometry satisfying $\ker p = \ker f$. This in fact fixes p uniquely: H decomposes into a direct sum $H \cong \ker f \oplus \ker f^\perp$ and since

(i) $\ker f = \ker |f| = \ker p$ and (ii) $\ker |f|^\perp = \overline{\operatorname{im} |f|}$ (see e.g. [31, Lemma 2.1]), H in fact decomposes as $H \cong \overline{\operatorname{im} |f|} \oplus \ker p$. The action of p on its kernel is clear and on its orthocomplement it has to be given by continuous extension of $|f|x \mapsto fx$ for $p|f| = f$ to hold. Now, let r be the projection onto $\ker p = \ker |f|$, and set $i := |f| + r$. On $\ker |f|$ i acts as the identity and on its orthocomplement it acts as $|f|$. Hence i is positive. By construction, $\ker i = 0$ so that i is monic. Being self-adjoint it is also epic. Moreover, since $\ker p = \ker |f|$ the factorization $f = pi$ follows from $f = p|f|$.

Similarly f^\dagger factors as $f^\dagger = q|f^\dagger|$ where q is defined similarly, and this factorization can be modified to obtain $f^\dagger = qj$, where j is a self-adjoint (in fact positive) bimorphism. Taking daggers, we have $f = jq$.

Hence it remains to show that $q^\dagger = p$. Since the dagger in **Hilb** is given by adjoints of bounded linear maps, this boils down to showing that

$$\langle px, y \rangle = \langle x, qy \rangle \quad (4.7.1)$$

holds for all $x \in H$ and $y \in K$. We can now use our orthogonal decompositions $H \cong \overline{\operatorname{im} |f|} \oplus \ker p$ and $K \cong \overline{\operatorname{im} |f^\dagger|} \oplus \ker q$ and consider separately the cases where x and y are in each of the summands. If $x \in \ker p$ then the left hand side of (4.7.1) equals zero. But then x is orthogonal to $\operatorname{im} f = \operatorname{im} q$ so the right hand side is zero as well. Similarly, both sides equal zero when $y \in \ker q$. Thus we're left to consider the case when $x \in \overline{\operatorname{im} |f|}$ and $y \in \overline{\operatorname{im} |f^\dagger|}$, and by continuity we may further assume $x \in \operatorname{im} |f|$ and $y \in \operatorname{im} |f^\dagger|$. Pick $z \in H$ and $w \in K$ with $|f|z = x$ and $|f^\dagger|w = y$. Now $px = p|f|z = fz$ and $qy = q|f^\dagger|w = f^\dagger w$, so (4.7.1) boils down to whether

$$\langle fz, |f^\dagger|w \rangle = \langle |f|z, f^\dagger w \rangle$$

Now the left hand side equals $\langle |f^\dagger|fz, w \rangle$ and the right hand side equals $\langle f|f|z, w \rangle$ so they are equal by Lemma 4.7.2, completing the proof. \square

In **Hilb**, we can not guarantee that i and j are isomorphisms in general. A good example is when $f: \ell^2(\mathbb{N}) \rightarrow \ell^2(\mathbb{N})$ is defined on the n -th basis element e_n by $f(e_n) = e_n/(n+1)$. Then the factorization above gives id as the partial isometry and f as the positive bimorphism. However, f does not have an inverse in **Hilb** – indeed, the “inverse” defined by $e_n \mapsto (n+1)e_n$ is not bounded.

Other dagger categories admitting polar decomposition include inverse categories, such as **Pinj**, and any groupoid, in which every morphism itself is already a partial isometry.

One might think that polar decomposition is an orthogonal factorization system, but there are several differences. First, the composition of partial isometries need not be a partial isometry, and the composition of self-adjoint bimorphisms need not be self-adjoint. Second, an isomorphism need not be a partial isometry nor self-adjoint. Third, p , i , and j are not required to be unique to f . Fourth, the factorization $f = pi$ respects the dagger: even though one may also factor $f^\dagger = qj$ and hence $f = j^\dagger q$, we are additionally requiring that $p = q$.

Recall that a dagger category is unitary whenever two objects being isomorphic implies that they are also unitarily isomorphic.

Proposition 4.7.4. *Dagger categories that have polar decomposition are unitary.*

Proof. Factor an isomorphism $f: A \rightarrow B$ as $f = pi = jp$ with p a partial isometry and i, j self-adjoint bimorphisms. Now $pi = f$ implies that p has a right inverse and $jp = f$ that it has a left inverse. Hence p must be an isomorphism. Also being a partial isometry, $p: A \rightarrow B$ is therefore unitary. \square

The theme of the rest of this section will be that dagger limits may be viewed as the partial isometry part of a polar decomposition of ordinary limits.

Example 4.7.5. The theme of the rest of this section will be that dagger limits may be viewed as the partial isometry part of a polar decomposition of ordinary limits. Here are some examples to warm up to this theme.

- Let $e: E \rightarrow A$ be an equalizer of morphisms $f, g: A \rightarrow B$ in a dagger category. If $e = pi = jp$ is a polar decomposition, then $p: A \rightarrow B$ is a dagger equalizer of f and g . Indeed, since i is a bimorphism we see that $fp = gp$, so that p factors through e as $p = ek$ for some $k: A \rightarrow A$. Precomposing with i we see that $e = eki$, whence $ki = \text{id}$. Since i is a bimorphism and has a left inverse, it is an isomorphism and $k = i^{-1}$. Hence $p = ei^{-1}$ is an equalizer and hence monic, and by definition p is a partial isometry and hence dagger monic.
- A cone over a family of dagger monomorphisms $f_i: A_i \rightarrow B$ consists formally of a map $l_i: A \rightarrow A_i$ for each i and of a map $f: A \rightarrow B$. However, the whole cone is determined by the map f : this is because $f_i l_i = f$ implies $l_i = f_i^\dagger f_i l_i = f_i^\dagger f$ by dagger monicness of f_i . A map $f: A \rightarrow B$ defines a cone in this manner iff $f_i f_i^\dagger f = f$ for each i . Now, if f defines a limiting cone and $f = pi = jp$ is a polar decomposition, then $p: A \rightarrow B$ is a dagger intersection of $\{f_i\}$: since i is a

bimorphism we see that p defines a cone for f_i and hence factors through f . As in the equalizer case, this implies that i is an isomorphism whence $p = fi^{-1}$ is a pullback of monics and hence monic. Since p is a partial isometry by definition it is also dagger monic.

- Let a projection $e = e^\dagger = e^2: A \rightarrow A$ be split by $f: B \rightarrow A$ and $g: A \rightarrow B$, so $e = fg$ and $gf = \text{id}_B$. If $f = pi = jp$ is a polar decomposition, then $p: B \rightarrow A$ is a dagger splitting of e : indeed f is the equalizer of id and e and hence so is p by the above, so that p is monic and hence dagger monic. It remains to check that $e = pp^\dagger$. Now $ep = p$ so that $pp^\dagger = epp^\dagger$ as well. Applying the dagger to both sides of this results in $pp^\dagger = pp^\dagger e$. On the other hand, $e = e^2 = fge = pige = pp^\dagger pige = pp^\dagger fge = pp^\dagger e^2 = pp^\dagger e$. Hence $e = pp^\dagger e = pp^\dagger$.

These examples are no accident, and the theorems proven below will bear out this theme. Before studying polar decompositions of more general diagrams we need the following lemma.

Lemma 4.7.6. *Let $f: A \rightarrow B$ be a morphism in a dagger category with polar decomposition $f = pi = jp$, and let $g: A \rightarrow A$ be self-adjoint. If g commutes with $f^\dagger f$, then it also commutes with $p^\dagger p$.*

Proof. Observe that $f^\dagger f = p^\dagger pii$.

$$\begin{array}{ccccc}
 A & \xrightarrow{f} & B & \xrightarrow{f^\dagger} & A \\
 i \downarrow & \nearrow p & j \downarrow & \nearrow p^\dagger & \\
 A & \xrightarrow{f} & B & & \\
 i \downarrow & \nearrow p & & & \\
 A & & & &
 \end{array}$$

It follows that

$$p^\dagger p g p^\dagger p i i = p^\dagger p g f^\dagger f = p^\dagger p f^\dagger f g = f^\dagger f g = g f^\dagger f = g p^\dagger p i i.$$

Because i is a bimorphism, $p^\dagger p g p^\dagger p = g p^\dagger p$. Since g is self-adjoint, the left hand side of this equation is self-adjoint. Therefore also the right-hand side is self-adjoint. Thus $p^\dagger p g = g p^\dagger p$. \square

The next theorem roughly shows that ‘‘polar decomposition turns based limits into dagger limits’’.

Theorem 4.7.7. *Let Ω be a basis of \mathbf{J} . Assume that $D: \mathbf{J} \rightarrow \mathbf{C}$ has a limit $l_A: L \rightarrow D(A)$ satisfying $l_A^\dagger l_A l_B^\dagger l_B = l_B^\dagger l_B l_A^\dagger l_A$ for all $A, B \in \Omega$. If \mathbf{C} admits polar decomposition, (D, Ω) has a dagger limit.*

Proof. Pick a polar decomposition $l_A = p_A i_A = j_A p_A$ for each $A \in \Omega$. For $B \in \mathbf{J} \setminus \Omega$, set $p_B = D(f)p_A$, where A is the unique object in Ω with $\mathbf{J}(A, B) \neq \emptyset$. If $f, g: A \rightarrow B$, then

$$D(f)p_A i_A = D(f)l_A = D(g)l_A = D(g)p_A i_A$$

and hence $D(f)p_A = D(g)p_A$. So p_B is independent of the choice of $f: A \rightarrow B$, and $p_A: L \rightarrow D(A)$ forms a cone. By construction, each p_A is a partial isometry whenever $A \in \Omega$. Moreover, by assumption and Lemma 4.7.6, $p_A^\dagger p_A$ commutes with $l_B^\dagger l_B$ when $A, B \in \Omega$. Then, by another application of the lemma $p_A^\dagger p_A$ commutes with $p_B^\dagger p_B$.

It remains to show that $p_A: L \rightarrow D(A)$ forms a limiting cone. We will establish this by proving that the unique map $f: L \rightarrow L$ of cones from p_A to l_A is an isomorphism. Thus we need to find a map $g: L \rightarrow L$ from l_A to p_A that is the inverse of f . That g is a map of cones means that the triangle

$$\begin{array}{ccc} L & \xrightarrow{l_A} & D(A) \\ g \downarrow & \nearrow p_A & \\ L & & \end{array}$$

commutes for each A in \mathbf{J} , or equivalently for each $A \in \Omega$. Postcomposing with the bimorphisms j_A we see that this is equivalent to finding $g: L \rightarrow L$ such that

$$\begin{array}{ccc} L & \xrightarrow{l_A} & D(A) \\ g \downarrow & & \downarrow j_A \\ L & \xrightarrow{l_A} & D(A) \end{array}$$

for each $A \in \Omega$. As $l_A: L \rightarrow D(A)$ is a limit cone, the existence of such a g follows as soon as $j_A l_A: L \rightarrow D(A)$ with $A \in \Omega$ generates a cone. But

$$j_A l_A = j_A p_A i_A = l_A i_A$$

and $l_A i_A: L \rightarrow D(A)$ with $A \in \Omega$ obviously generates a cone. Thus we have found a cone map $g: L \rightarrow L$ from l_A to p_A . It suffices to show that it is the inverse of f . On the one hand $fg = \text{id}_L$ by the universal property of the cone l_A . On the other hand,

$p_A g f = p_A$ for each $A \in \Omega$, and by postcomposing with j_A we see that $g f$ is also a cone map from l_A to l_A , and thus equal to the identity. \square

In particular, if \mathbf{C} admits polar decomposition and A_1 and A_2 have a product $(A_1 \times A_2, p_1, p_2)$ satisfying $p_1^\dagger p_1 p_2^\dagger p_2 = p_2^\dagger p_2 p_1^\dagger p_1$ then the dagger product of A_1 and A_2 exists as well. Using polar decomposition and splittings of projections, one can also construct the dagger product of A and B from their biproduct, without any further conditions required from the biproduct.

Recall that, in a category with a zero object (or more generally, zero morphisms), a biproduct of A_1 and A_2 consists of $(A_1 \oplus A_2, p_1, p_2, i_1, i_2)$ where $(A_1 \oplus A_2, p_1, p_1)$ is a product of A_1 and A_2 , $(A_1 \oplus A_2, i_1, i_2)$ is their coproduct, and moreover

$$\begin{aligned} p_1 i_1 &= \text{id}_{A_1} & p_2 i_2 &= \text{id}_{A_2} \\ p_2 i_1 &= 0_{A_1, A_2} & p_1 i_2 &= 0_{A_2, A_1} \end{aligned}$$

or more succinctly,

$$p_n i_k = \delta_{k,n} := \begin{cases} \text{id}_{A_n} & \text{if } n = k \\ 0_{A_k, A_n} & \text{otherwise.} \end{cases}$$

In Section 5.2 we will see how to generalize this definition, allowing us to work without assuming zero morphisms. The theorem below remains true for such generalized biproducts, as explained in Remark 5.2.5.

Theorem 4.7.8. *Let \mathbf{C} be a dagger category that admits polar decomposition and has dagger splittings of projections. If two objects A_1 and A_2 of \mathbf{C} have a biproduct, they have a dagger product as well.*

Proof. Given two objects A_1 and A_2 , take their biproduct $(A_1 \oplus A_2, p_1, p_2, i_1, i_2)$. Assume first that we've produced a cone $q_i: P \rightarrow A_i$ such that $q_n q_k^\dagger = \delta_{n,k}$. As $(A_1 \oplus A_2, i_1, i_2)$ is a coproduct and $(A_1 \oplus A_2, p_1, p_2)$ is a product we can find unique maps $f: A_1 \oplus A_2 \rightarrow P$ and $g: P \rightarrow A_1 \oplus A_2$ making the diagram

$$\begin{array}{ccccc} & & \delta_{n,k} & & \\ & & \xrightarrow{\quad} & & \\ A_k & \xrightarrow{q_k^\dagger} & & \xrightarrow{q_n} & A_n \\ & \searrow & & \nearrow & \\ & & P & & \\ i_k \downarrow & & \xrightarrow{f} & & \uparrow p_n \\ A_1 \oplus A_2 & & & & A_1 \oplus A_2 \\ & & \xrightarrow{g} & & \end{array}$$

commute. By the universal properties of $A_1 \oplus A_2$ this implies that $g f = \text{id}$. Now, f is a map of cocones $(A_1 \oplus A_2, i_1, i_2) \rightarrow (P, q_1^\dagger, q_2^\dagger)$. We will prove that it is also a map of

cones $(A_1 \oplus A_2, p_1, p_2) \rightarrow (P, q_1, q_2)$. Consider the diagram

$$\begin{array}{ccccc}
 A_1 \oplus A_2 & \xrightarrow{f} & P & & \\
 \uparrow i_k & & \nearrow q_k^\dagger & & \downarrow q_n \\
 A_k & \xrightarrow{i_k} & A_1 \oplus A_2 & \xrightarrow{p_n} & A_n
 \end{array}$$

The bottom part commutes since both paths equal $\delta_{n,k}$ and the top part commutes by definition of f . Hence the rectangle commutes, and since i_1 and i_2 are jointly epic this shows that f is a map of cones. Now, g^\dagger is a map of cocones $(A_1, A_2, p_1^\dagger, p_2^\dagger) \rightarrow (P, q_1^\dagger, q_2^\dagger)$ so the same argument shows that g^\dagger is also a map of cones. Taking the dagger again, this means that g is not only a map of cones $(P, q_1, q_2) \rightarrow (A_1 \oplus A_2, p_1, p_2)$ but also a map of cocones $(P, q_1^\dagger, q_2^\dagger) \rightarrow (A_1 \oplus A_2, i_1, i_2)$. These observations are true for any cone P satisfying $q_n q_k^\dagger = \delta_{n,k}$. As any such cone satisfies normalization and independence, it is sufficient to find a cone P for which we can prove that $fg = \text{id}_P$ since then (P, q_1, q_2) will also be a product. Finding such a P is what we'll do in the remainder.

Factorize p_1 as $p_1 = pi = jp$ and i_2 as $i_2 = kr = rl$. Set $d_1 := p$ and $d_2 := r^\dagger$. We claim that

$$d_n d_k^\dagger = \delta_{n,k}$$

Indeed, since p_1 and i_2^\dagger are epimorphisms the maps d_1 and d_2 are dagger epic. Moreover, $0 = p_2 i_1 = j d_1 d_2^\dagger l = j 0 l$ whence $d_1 d_2^\dagger = 0$ and thus $d_2 d_1^\dagger = 0$. Hence $(A_1 \oplus A_2, d_1, d_2)$ is a cone satisfying (i), so as remarked at the end of the first half of the proof, we get maps

$$f: (A_1 \oplus A_2, p_1, p_2, i_1, i_2) \rightleftarrows (A_1 \oplus A_2, d_1, d_2, d_1^\dagger, d_2^\dagger): g$$

that are maps of cones and cocones. Taking daggers, we have maps

$$g^\dagger: (A_1 \oplus A_2, i_1^\dagger, i_2^\dagger, p_1^\dagger, p_2^\dagger) \rightleftarrows (A_1, A_2, d_1, d_2, d_1^\dagger, d_2^\dagger): f^\dagger$$

that are also compatible with the (co)cone structures. Moreover, since $(A_1 \oplus A_2, p_1, p_2, i_1, i_2)$ and $(A_1 \oplus A_2, i_1^\dagger, i_2^\dagger, p_1^\dagger, p_2^\dagger)$ are biproducts, there is exactly one (co)cone map in both directions, and these are isomorphisms. Since

$$f^\dagger f: (A_1 \oplus A_2, p_1, p_2, i_1, i_2) \rightleftarrows (A_1 \oplus A_2, i_1^\dagger, i_2^\dagger, p_1^\dagger, p_2^\dagger): gg^\dagger$$

are (co)cone maps, these maps have to be inverses to each other. Hence $h := fgg^\dagger f^\dagger$ is a self-adjoint cone map $(A_1 \oplus A_2, d_1, d_2) \rightarrow (A_1 \oplus A_2, d_1, d_2)$ satisfying

$$h^2 = (fgg^\dagger f^\dagger)(fgg^\dagger f^\dagger) = f(gg^\dagger f^\dagger f)gg^\dagger f^\dagger = fgg^\dagger f^\dagger = h$$

Thus h is a projection; let $e: P \rightarrow (A_1 \oplus A_2)$ split it and set $q_i := d_i e$. Now

$$q_n q_k^\dagger = d_n e e^\dagger d_k^\dagger = d_n h d_k^\dagger = d_n d_k = \delta_{n,k}$$

so we have canonical maps

$$\tilde{f}: (A_1 \oplus A_2, p_1, p_2, i_1, i_2) \xleftrightarrow{\quad} (P, q_1, q_2, q_1^\dagger, q_2^\dagger): \tilde{g}$$

and $\tilde{g} \circ \tilde{f} = \text{id}_{A_1 \oplus A_2}$ holds automatically, whereas whether $\tilde{f} \circ \tilde{g} = \text{id}_P$ is at stake. By construction e is a map of cones and hence e^\dagger is a map of cocones. Hence the bottom path in region (i) of

is a map of cones and thus equal to \tilde{g} , establishing commutativity of region (i). Similarly, $e^\dagger f$ is a composite of two maps of cocones and hence equal to \tilde{f} , so that region (ii) commutes. Region (iii) commutes by definition of h and the remaining triangles on the left commute by definition of e . Hence the whole diagram commutes, showing that $\tilde{f} \circ \tilde{g} = \text{id}_P$, as desired. \square

Corollary 4.7.9. *A dagger category admitting polar decomposition has finitely based dagger limits if and only if its underlying category has equalizers, intersections and either (i) finite biproducts or (ii) finite products $(A_1 \dots A, p_1, \dots, p_n)$ satisfying $p_i^\dagger p_i p_j^\dagger p_j = p_j^\dagger p_j p_i^\dagger p_i$.*

Proof. Case (i) follows from Example 4.7.5 and Theorems 4.4.8 and 4.7.8, whereas case (ii) follows directly from Theorems 4.4.8 and 4.7.7. \square

What made the previous results work for based diagrams is that we did not need to worry about the path taken to an object. However, if $A, B \in \Omega$ are distinct and admit maps $f: A \rightarrow C \leftarrow B: g$, the equation $D(f)l_A = l_C = D(g)l_B$ need not imply that $D(f)p_A = D(g)p_B$. Hence forgetting about the bimorphisms is not possible for

arbitrary diagrams, and to make it work one has to change the diagram so that joint polar decomposition becomes available. The following theorem makes precise this idea that “polar decomposition turns limits into dagger limits of isomorphic diagrams”. Recall that a category is balanced if all bimorphisms in it are isomorphisms.

Theorem 4.7.10. *Consider a diagram $D: \mathbf{J} \rightarrow \mathbf{C}$ in a balanced dagger category \mathbf{C} that admits polar decomposition. Suppose that the diagram has a limit $l_A: L \rightarrow D(A)$ such that $l_A^\dagger l_A l_B^\dagger l_B = l_B^\dagger l_B l_A^\dagger l_A$ for all A and B in some weakly initial Ω . There is a diagram E naturally isomorphic to D such that (E, Ω) has a dagger limit.*

Proof. Pick a polar decomposition $l_A = p_A i_A = j_A p_A$ for each $A \in \mathbf{J}$. As \mathbf{C} is balanced, each i_A and j_A is an isomorphism, so we can define a new diagram $E: \mathbf{J} \rightarrow \mathbf{C}$ by $E(A) = D(A)$ on objects, and by $E(f) = j_B^{-1} D(f) j_A$ on morphisms $f: A \rightarrow B$. By construction $j_A: E(A) \rightarrow D(A)$ forms a natural isomorphism $E \Rightarrow D$. Moreover, since $l_A = j_A p_A$, in fact L is a limit of E with a limiting cone $p_A: L \rightarrow E(A)$ of partial isometries. It remains to check independence. This is done as in the proof of Theorem 4.7.7. By assumption and Lemma 4.7.6, $p_A^\dagger p_A$ commutes with $l_B^\dagger l_B$ when $A, B \in \Omega$. Then, by another application of the lemma $p_A^\dagger p_A$ commutes with $p_B^\dagger p_B$. \square

4.8 Commutativity of limits and colimits

In this section we investigate to what extent dagger limits commute with dagger (co)limits.

Let us start by looking at whether dagger limits commute with dagger limits. Assume that \mathbf{C} has all (\mathbf{J}, Ω) -shaped dagger limits and all (\mathbf{K}, Ψ) -shaped dagger limits. Then, a bifunctor $D: \mathbf{J} \times \mathbf{K} \rightarrow \mathbf{C}$ induces functors $\mathbf{J} \rightarrow \mathbf{C}$ and $\mathbf{K} \rightarrow \mathbf{C}$ defined by $j \mapsto \text{dlim}_k^\Psi D(j, k)$ and $k \mapsto \text{dlim}_j^\Omega D(j, k)$. Since limits commute with limits, there exists a canonical isomorphism $\text{dlim}_j^\Omega \text{dlim}_k^\Psi D(j, k) \simeq \text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k)$ between the two limits of D . In keeping with dagger category theory, we would like this canonical isomorphism to be unitary. Moreover, we would like both sides to be dagger limits of $(D, \Omega \times \Psi)$. We now prove that this holds whenever \mathbf{J} and \mathbf{K} are based and \mathbf{C} has zero morphisms. Later we will relax this to arbitrary \mathbf{J} and \mathbf{K} under a technical condition on the bifunctor D , that we conjecture is in fact not necessary.

Theorem 4.8.1. *Let Ω and Ψ be bases for \mathbf{J} and \mathbf{K} . Assume that \mathbf{C} has zero morphisms, all (\mathbf{J}, Ω) -shaped dagger limits, all (\mathbf{K}, Ψ) -shaped dagger limits, and let $D: \mathbf{J} \times \mathbf{K} \rightarrow \mathbf{C}$ be a bifunctor. Then $\text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k)$ and $\text{dlim}_j^\Omega \text{dlim}_k^\Psi D(j, k)$ are both dagger limits of $(D, \Omega \times \Psi)$ and hence unitarily isomorphic.*

Proof. By symmetry, it suffices to prove the claim for $\mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k)$. Since ordinary limits commute with limits, $\mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k)$ is a limit of D . Hence we need only check normalization and independence for $(j, k) \in \Omega \times \Psi$. In the presence of zero morphisms, we can use the simpler description from Proposition 4.4.13.

For normalization, we start by showing that, for fixed $(j, k) \in \Omega \times \Psi$, the morphism p_k defined as the composition of canonical morphisms

$$\mathrm{dlim}_j^\Omega D(j, k) \rightarrow \mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) \rightarrow \mathrm{dlim}_j^\Omega D(j, k) \rightarrow D(j, k) \rightarrow \mathrm{dlim}_j^\Omega D(j, k)$$

factors through the canonical morphism $\mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) \rightarrow \mathrm{dlim}_j^\Omega D(j, k)$. This is done by extending it to a cone on $\mathrm{dlim}_j^\Omega D(j, k)$ for the functor $\mathrm{dlim}_j^\Omega D(j, -): \mathbf{K} \rightarrow \mathbf{C}$. For h in \mathbf{K} , define $p_h: \mathrm{dlim}_j^\Omega D(j, k) \rightarrow \mathrm{dlim}_j^\Omega D(j, h)$ by

$$p_h = \begin{cases} \mathrm{dlim}_j^\Omega D(j, f) p_k & \text{if } \mathbf{K}(k, h) \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

To see that this defines a cone, it suffices to check that p_h is independent of the choice of $f: k \rightarrow h$. By the universal property of $\mathrm{dlim}_j^\Omega D(j, h)$, we may postcompose with a projection to an arbitrary $D(i, h)$ with $i \in \Omega$ and show that the resulting morphism does not depend on the choice of f . This splits into two cases depending on whether $i \neq j$ or not. If $i \neq j$, then the end result is always zero, since the following diagram commutes for every f :

$$\begin{array}{ccccc} & & \mathrm{dlim}_j^\Omega D(j, k) & \xrightarrow{\mathrm{dlim}_j^\Omega D(j, f)} & \mathrm{dlim}_j^\Omega D(j, h) \\ & \nearrow & \downarrow & & \downarrow \\ D(j, k) & \xrightarrow{0} & D(i, k) & \xrightarrow{D(i, f)} & D(i, h) \\ & \searrow & & & \downarrow \text{id} \\ & & & & D(i, h) \\ & \searrow & & & \\ & & 0 & \xrightarrow{\quad\quad\quad} & D(i, h) \end{array}$$

In case $i = j$, we will prove that the following diagram commutes:

$$\begin{array}{ccccccc} \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & \mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & D(j, k) \\ & & & & \downarrow & & \downarrow \\ & & & & D(j, k) & \longleftarrow & \mathrm{dlim}_j^\Omega D(j, k) \\ & & & & \downarrow D(j, f) & & \downarrow \mathrm{dlim}_j^\Omega D(j, f) \\ & & & & D(j, h) & \longleftarrow & \mathrm{dlim}_j^\Omega D(j, h) \\ & & \mathrm{dlim}_j^\Omega D(j, h) & \longrightarrow & & & \end{array}$$

The top right square commutes because $j \in \Omega$, and the rest of the diagram commutes by definition of $\text{dlim}_j^\Omega D(j, f)$. The path along the top is p_h followed by a projection to $D(j, h)$, whereas the other path is independent of the choice of $f: k \rightarrow h$ since $\text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k)$ is a cone for $\text{dlim}_j^\Omega D(j, -)$. Thus

$$p_h: \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, h)$$

forms a cone. Hence it factors through $\text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k)$. In particular, p_k factors through $\text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, k)$. By Remark 2.1.5 this implies that the following diagram of canonical morphisms commutes:

$$\begin{array}{ccccccccc} \text{dlim}_j^\Omega D(j, k) & \longrightarrow & \text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) & \longrightarrow & \text{dlim}_j^\Omega D(j, k) & \longrightarrow & D(j, k) & \longrightarrow & \text{dlim}_j^\Omega D(j, k) \\ & & \downarrow & & & & & & \uparrow \\ \text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) & \longrightarrow & \text{dlim}_j^\Omega D(j, k) & \longrightarrow & D(j, k) & \longrightarrow & \text{dlim}_j^\Omega D(j, k) & \longrightarrow & \text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \end{array}$$

As the path along the bottom is self-adjoint, so is the top path. So the projections

$$\text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, k)$$

and

$$\text{dlim}_j^\Omega D(j, k) \rightarrow D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, k)$$

commute. Finally, Remark 2.1.5 guarantees that the composition of canonical morphisms $\text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, k) \rightarrow D(j, k)$ is a partial isometry.

For independence, it suffices to show for $(j, k), (i, h) \in \Omega \times \Psi$ that the composition $D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, h) \rightarrow D(i, h)$ of canonical morphisms is zero when $(j, k) \neq (i, h)$. If $k \neq h$ this follows from the fact that $\text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, h)$ is zero. Hence we may assume that $h = k$ and consider the case $i \neq j$. As above, the projections

$$\text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, k)$$

and

$$\text{dlim}_j^\Omega D(j, k) \rightarrow D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, k)$$

commute. Hence the following diagram of canonical morphisms commutes:

$$\begin{array}{ccccccc}
 D(j, k) & \longrightarrow & \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & \mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & \mathrm{dlim}_j^\Omega D(j, k) \\
 \downarrow & & \uparrow & & \nearrow & & \downarrow \\
 \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & D(j, k) & & D(j, k) & \xrightarrow{0} & D(i, k) \\
 \downarrow & & & & \nearrow & & \\
 \mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & \mathrm{dlim}_j^\Omega D(j, k) & & & &
 \end{array}$$

This concludes the proof. \square

Next, consider whether dagger limits commute with dagger colimits. If \mathbf{C} has all (\mathbf{J}, Ω) -shaped dagger limits and all (\mathbf{K}, Ψ) -shaped dagger colimits (i.e. $(\mathbf{K}^{\mathrm{op}}, \Psi)$ -shaped dagger limits), it is natural to ask when the canonical morphism

$$\mathrm{dcolim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) \rightarrow \mathrm{dlim}_j^\Omega \mathrm{dcolim}_k^\Psi D(j, k)$$

is unitary. This canonical morphism τ and morphisms α_k are defined by making the following diagram commute for each k :

$$\begin{array}{ccccc}
 \mathrm{dcolim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) & \longleftarrow & \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & D(j, k) \\
 & \searrow \tau & \downarrow \alpha_k & & \downarrow \\
 & & \mathrm{dlim}_j^\Omega \mathrm{dcolim}_k^\Psi D(j, k) & \longrightarrow & \mathrm{dcolim}_k^\Psi D(j, k)
 \end{array}$$

A priori one might hope τ to be unitary very generally. After all, (\mathbf{K}, Ψ) -shaped dagger colimits are just $(\mathbf{K}^{\mathrm{op}}, \Psi)$ -shaped dagger limits, so one might expect that commutativity of limits with colimits boils down to commutativity of limits with limits. To be slightly more precise, given $D: \mathbf{J} \times \mathbf{K} \rightarrow \mathbf{C}$, one would like to define $\hat{D}: \mathbf{J} \times \mathbf{K}^{\mathrm{op}} \rightarrow \mathbf{C}$ by “applying the dagger to the second variable” and then calculating as follows:

$$\begin{aligned}
 \mathrm{dcolim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) &= \mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega \hat{D}(j, k) \\
 &\simeq_+ \mathrm{dlim}_j^\Omega \mathrm{dlim}_k^\Psi \hat{D}(j, k) = \mathrm{dlim}_j^\Omega \mathrm{dcolim}_k^\Psi D(j, k)
 \end{aligned}$$

This, however, is a trap: \hat{D} is not guaranteed to be a bifunctor. Indeed, the formula $\hat{D}(f, g) = D(f, \mathrm{id})D(g, \mathrm{id})^\dagger$ defines a bifunctor $\mathbf{J} \times \mathbf{K}^{\mathrm{op}} \rightarrow \mathbf{C}$ if and only if every

morphism $(f, g): (j, k) \rightarrow (i, h)$ in $\mathbf{J} \times \mathbf{K}$ makes the following diagram commute:

$$\begin{array}{ccc} D(j, h) & \xrightarrow{D(f, \text{id})} & D(i, k) \\ D(\text{id}, g)^\dagger \downarrow & & \downarrow D(\text{id}, g)^\dagger \\ D(j, k) & \xrightarrow{D(f, \text{id})} & D(i, k) \end{array}$$

Definition 4.8.2. A bifunctor $D: \mathbf{J} \times \mathbf{K} \rightarrow \mathbf{C}$ into a dagger category \mathbf{C} is *adjointable* when $D(-, g): D(-, k) \Rightarrow D(-, h)$ is an adjointable natural transformation for each $g: k \rightarrow h$ in \mathbf{K} . Equivalently, D is adjointable when $D(f, -)$ is an adjointable natural transformation for each morphism f in \mathbf{J} .

Let us temporarily go back to considering whether dagger limits commute with dagger limits. The extra condition of adjointability of D lets us prove that this is true for arbitrary shapes \mathbf{J} and \mathbf{K} . We conjecture that the extra condition is in fact not needed.

Theorem 4.8.3. *Assume that \mathbf{C} has all (\mathbf{J}, Ω) -shaped dagger limits, all (\mathbf{K}, Ψ) -shaped dagger limits, and let $D: \mathbf{J} \times \mathbf{K} \rightarrow \mathbf{C}$ be an adjointable bifunctor. Then $\text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k)$ and $\text{dlim}_j^\Omega \text{dlim}_k^\Psi D(j, k)$ are both dagger limits of $(D, \Omega \times \Psi)$ and hence unitarily isomorphic.*

Proof. By symmetry, it suffices to prove the claim for $\text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k)$. Since ordinary limits commute with limits, $\text{dlim}_k^\Psi \text{dlim}_j^\Omega D(j, k)$ is the limit of D . Hence we only need to check normalization and independence for $(j, k) \in \Omega \times \Psi$. Consider the following diagram of canonical morphisms for some morphism $f: k \rightarrow h$ in \mathbf{K} :

$$\begin{array}{ccc} \text{dlim}_j^\Omega D(j, k) & \xrightarrow{\text{dlim}_j^\Omega D(j, f)} & \text{dlim}_j^\Omega D(j, h) \\ \downarrow & & \downarrow \\ D(j, k) & \xrightarrow{D(\text{id}, f)} & D(j, h) \\ \downarrow & & \downarrow \\ \text{dlim}_j^\Omega D(j, k) & \xrightarrow{\text{dlim}_j^\Omega D(j, f)} & \text{dlim}_j^\Omega D(j, h) \end{array}$$

The top square commutes by definition of $\text{dlim}_j^\Omega D(j, f)$. Since $D(-, f)$ is adjointable, the bottom square commutes too by Corollary 4.3.8. Hence the whole diagram commutes, and the family $\sigma_k: \text{dlim}_j^\Omega D(j, k) \rightarrow D(j, k) \rightarrow \text{dlim}_j^\Omega D(j, k)$ is natural in k . Since $\sigma: \text{dlim}_j^\Omega D(j, -) \rightarrow \text{dlim}_j^\Omega D(j, -)$ is natural and pointwise self-adjoint, it is

an adjointable natural transformation. Lemma 4.3.7 now makes the following diagram of canonical morphisms commute for each $j \in \Omega$ and $h, k \in \Psi$:

$$\begin{array}{ccccccc}
 \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & D(j, k) & \longrightarrow & \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & \mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) \\
 \downarrow & & & & & & \downarrow \\
 \mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) & \longrightarrow & \mathrm{dlim}_j^\Omega D(j, h) & \longrightarrow & D(j, h) & \longrightarrow & \mathrm{dlim}_j^\Omega D(j, h)
 \end{array}
 \tag{**}$$

Choosing $k = h$ shows that the projections $\mathrm{dlim}_j^\Omega D(j, k) \rightarrow D(j, k) \rightarrow \mathrm{dlim}_j^\Omega D(j, k)$ and $\mathrm{dlim}_j^\Omega D(j, k) \rightarrow \mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) \rightarrow \mathrm{dlim}_j^\Omega D(j, k)$ commute. Remark 2.1.5 then guarantees that the composite $\mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k) \rightarrow \mathrm{dlim}_j^\Omega D(j, k) \rightarrow D(j, k)$ of canonical morphisms is a partial isometry for each $(j, k) \in \Omega \times \Psi$, establishing normalization.

For independence, pick (j, k) and (i, h) in $\Omega \times \Psi$. We will show that the diagram in Figure 4.1 commutes. The fact that $k, h \in \Psi$ and $\mathrm{dlim}_k^\Psi \mathrm{dlim}_j^\Omega D(j, k)$ are normalized and independent at Ψ ensures the commutativity of regions (i), (iii), (viii), (xi), (xi),(xii) and (xiv). Similarly, $i, j \in \Omega$ implies the commutativity of region (iv). The remaining regions, namely (ii), (v), (vi), (vii), (x) and (xiii), are all instances of diagram (**). \square

We return to considering whether dagger limits commute with dagger colimits. If D is not adjointable, the colimit of limits need not be (unitarily) isomorphic to the limit of colimits.

Example 4.8.4. Dagger kernels need not commute with dagger cokernels in \mathbf{FHilb} if the bifunctor D is not adjointable. For a counterexample, let \mathbf{J} and \mathbf{K} both be the shape $f, g: A \rightrightarrows B$ giving rise to equalizers. If $D: \mathbf{J} \times \mathbf{K} \rightarrow \mathbf{FHilb}$ maps each $D(f, -)$ and $D(-, f)$ to zero, then fixing the rest of D corresponds to a choice of a commuting square in \mathbf{FHilb} . Let D be thus defined by the square:

$$\begin{array}{ccc}
 0 & \xrightarrow{0} & \mathbb{C} \\
 0 \downarrow & & \downarrow \mathrm{id} \\
 \mathbb{C} & \xrightarrow{\mathrm{id}} & \mathbb{C}
 \end{array}$$

Taking daggers of the horizontal arrows gives a square that does not commute, so D is not adjointable. Now, on the one hand, first taking cokernels horizontally and then

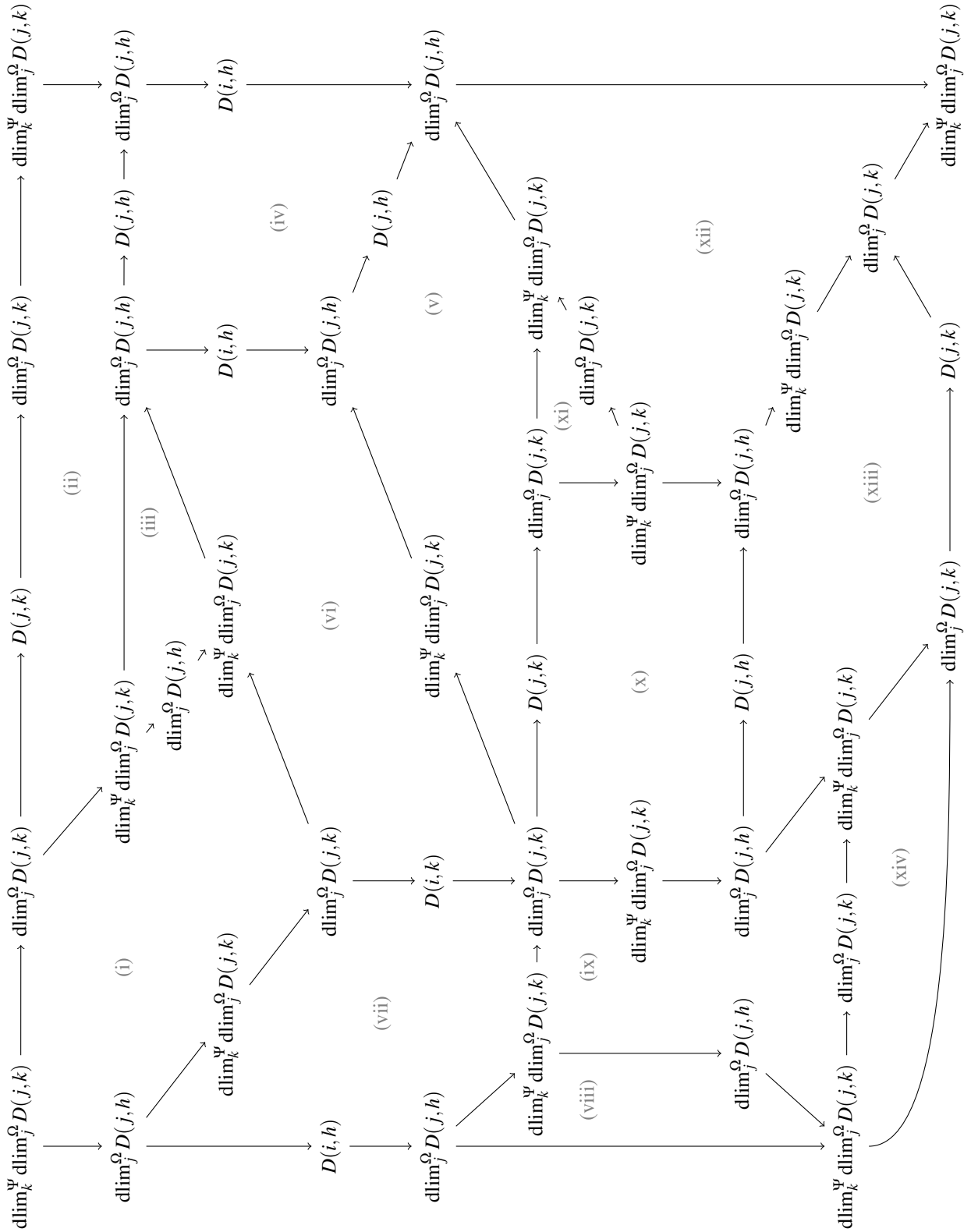


Figure 4.1: Proof of independence in Theorem 4.8.3.

taking kernels vertically gives \mathbb{C} :

$$\begin{array}{ccccc}
 & & & & \ker 0 = \mathbb{C} \\
 & & & & \downarrow \text{id} \\
 0 & \xrightarrow{0} & \mathbb{C} & \xrightarrow{\text{id}} & \text{coker } 0 = \mathbb{C} \\
 0 \downarrow & & \downarrow \text{id} & & \downarrow 0 \\
 \mathbb{C} & \xrightarrow{\text{id}} & \mathbb{C} & \xrightarrow{0} & \text{coker } \text{id}_{\mathbb{C}} = 0
 \end{array}$$

On the other hand, first taking kernels vertically and then taking cokernels horizontally gives 0:

$$\begin{array}{ccccc}
 \ker 0 = 0 & \xrightarrow{0} & \ker \text{id}_{\mathbb{C}} = 0 & \xrightarrow{0} & \text{coker } 0 = 0 \\
 0 \downarrow & & \downarrow 0 & & \\
 0 & \xrightarrow{0} & \mathbb{C} & & \\
 0 \downarrow & & \downarrow \text{id} & & \\
 \mathbb{C} & \xrightarrow{\text{id}} & \mathbb{C} & &
 \end{array}$$

Thus $\text{dlim}_j \text{dcolim}_k D(j, k) = \mathbb{C}$ is not isomorphic to $\text{dcolim}_k \text{dlim}_j D(j, k) = 0$, let alone unitarily so.

However, D not being adjointable is the only obstruction to dagger limits commuting with dagger colimits.

Theorem 4.8.5. *If \mathbf{C} has all (\mathbf{J}, Ω) -shaped dagger limits, all (\mathbf{K}, Ψ) -shaped dagger colimits, and $D: \mathbf{J} \times \mathbf{K} \rightarrow \mathbf{C}$ is an adjointable bifunctor, then the canonical morphism $\text{dcolim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega \text{dcolim}_k^\Psi D(j, k)$ is unitary.*

Proof. Because D is adjointable, $\hat{D}(f, g) = D(f, \text{id})D(g, \text{id})^\dagger$ defines an adjointable bifunctor $\hat{D}: \mathbf{J} \times \mathbf{K}^{\text{op}} \rightarrow \mathbf{C}$. It follows from Theorem 4.8.3 that

$$\begin{aligned}
 \text{dcolim}_k^\Psi \text{dlim}_j^\Omega D(j, k) &= \text{dlim}_k^\Psi \text{dlim}_j^\Omega \hat{D}(j, k) \\
 &\simeq_{\dagger} \text{dlim}_j^\Omega \text{dlim}_k^\Psi \hat{D}(j, k) = \text{dlim}_j^\Omega \text{dcolim}_k^\Psi D(j, k)
 \end{aligned}$$

Thus there exists a unitary morphism u making the following diagram commute:

$$\begin{array}{ccc}
 \text{dcolim}_k^\Psi \text{dlim}_j^\Omega D(j, k) & \xrightarrow{\quad\quad\quad} & \text{dlim}_j^\Omega D(j, k) \\
 \downarrow u & & \downarrow \\
 \text{dlim}_j^\Omega \text{dcolim}_k^\Psi D(j, k) & \xrightarrow{\quad\quad\quad} & \text{dcolim}_k^\Psi D(j, k) \xrightarrow{\quad\quad\quad} \text{dcolim}_k^\Psi D(j, k)
 \end{array}$$

We will show that u satisfies the commuting diagram that defines the canonical morphism $\tau: \text{dcolim}_k^\Psi \text{dlim}_j^\Omega D(j, k) \rightarrow \text{dlim}_j^\Omega \text{dcolim}_k^\Psi D(j, k)$, whence $u = \tau$. Postcomposing with the canonical morphism $\text{dlim}_j^\Omega \text{dcolim}_k^\Psi D(j, k) \rightarrow \text{dcolim}_k^\Psi D(j, k) \rightarrow D(j, k)$, this follows from commutativity of the following diagram:

$$\begin{array}{ccccc}
 \text{dlim}_j^\Omega \text{dcolim}_k^\Psi D(j, k) & \xrightarrow{\quad\quad\quad} & \text{dcolim}_k^\Psi D(j, k) & & \\
 \uparrow u & & \downarrow & & \\
 \text{dcolim}_k^\Psi \text{dlim}_j^\Omega D(j, k) & \xrightarrow{\quad\quad\quad} & \text{dlim}_j^\Omega D(j, k) & \xrightarrow{\quad\quad\quad} & D(j, k) \\
 \uparrow & & & & \uparrow \\
 \text{dlim}_j^\Omega D(j, k) & \xrightarrow{\quad\quad\quad} & D(j, k) & \xrightarrow{\quad\quad\quad} & \text{dcolim}_k^\Psi D(j, k) \\
 & \searrow \alpha_k & & \nearrow & \\
 & & \text{dlim}_j^\Omega \text{dcolim}_k^\Psi D(j, k) & &
 \end{array}$$

After all, the top part and bottom parts commute by definition of u and α_k . The commutativity of the middle rectangle is guaranteed by Lemma 4.3.7 once we show that the limiting cone morphisms $\text{dlim}_j^\Omega D(j, k) \rightarrow D(j, k)$ are the components of an adjointable natural transformation $\text{dlim}_j^\Omega D(j, -) \Rightarrow D(j, -)$. The naturality square commutes by definition of $\text{dlim}_j^\Omega D(j, f)$:

$$\begin{array}{ccc}
 \text{dlim}_j^\Omega D(j, k) & \xrightarrow{\text{dlim}_j^\Omega D(j, f)} & \text{dlim}_j^\Omega D(j, h) \\
 \downarrow & & \downarrow \\
 D(j, k) & \xrightarrow{D(\text{id}, f)} & D(j, h)
 \end{array}$$

Adjointability follows from Corollary 4.3.8 since $D(-, f)$ is adjointable:

$$\begin{array}{ccc}
 \text{dlim}_j^\Omega D(j, k) & \xrightarrow{\text{dlim}_j^\Omega D(j, f)} & \text{dlim}_j^\Omega D(j, h) \\
 \uparrow & & \uparrow \\
 D(j, k) & \xrightarrow{D(\text{id}, f)} & D(j, h)
 \end{array}$$

This concludes the proof. \square

Chapter 5

Ordinary limit-colimit coincidences via dagger limits

5.1 Ambilimits

Sometimes in ordinary (=non-dagger) category theory certain limits coincide with certain colimits, biproducts in abelian categories being a paradigmatic example. The usual way of understanding this is via absolute (co)limits, i.e. those that are preserved by every (enriched) functor. For instance, biproducts can be defined purely equationally in terms of $+$ and 0 , and hence are preserved by any functor preserving $+$ and 0 , or in other words, by any functor respecting the enrichment in commutative monoids. In fact, in the enriched setting one can show that limits coinciding with colimits is equivalent to absoluteness [139, 51].

Here we would like to offer an alternative perspective, not requiring enrichment. For instance, it turns out to be perfectly feasible to axiomatize biproducts without requiring *any* enrichment, and arguably this answers the question “What does it mean to be a product and a coproduct in a compatible way?”. From this point of view, it is then a theorem that the more general notion of biproduct coincides with the **CMon**-enriched notion whenever the category in question is suitably enriched.

Instead of talking at length of “limit-colimit coincidences” or risking confusion with bicategory theory by using the word “bilimit”, we will talk about ambilimits. The idea is to capture an object being simultaneously a limit and a colimit in a compatible manner.

Definition 5.1.1. Let $D_1 : \mathbf{J} \rightarrow \mathbf{C}$ and $D_2 : \mathbf{J}^{\text{op}} \rightarrow \mathbf{C}$ be a pair of diagrams satisfying $D_1(A) = D_2(A)$ for all $A \in \mathbf{J}$. Then an *ambilimit* of (D_1, D_2) is a tuple $(L, (p_A)_{A \in \mathbf{J}}, (i_A)_{A \in \mathbf{J}})$

such that $(L, (p_A)_{A \in \mathbf{J}})$ is a limit of D_1 , $(L, (i_A)_{A \in \mathbf{J}})$ is a colimit of D_2 and moreover the following equations hold:

normalization p_A and i_A are regular inverses to each other for every $A \in \mathbf{J}$, i.e. $i_A p_A i_A = i_A$ and $p_A i_A p_A = p_A$;

independence the idempotents on L induced by these commute, i.e. $i_A p_A i_B p_B = i_B p_B i_A p_A$ for all $A, B \in \mathbf{J}$.

In concrete examples of ambilimits, one often just specifies one D_i , the other one being derivable from it, see e.g. the limit-colimit coincidence in domain theory below. Before justifying the definition by the examples it captures, let us first satisfy a theoretical concern of canonicity: whenever one has two ambilimits of the same diagrams, a priori there are two isomorphisms between the ambilimits – one induced by the limit structure and one by the colimit structure. It turns out that these two isomorphisms always coincide.

Theorem 5.1.2. *If $(L, (p_A)_{A \in \mathbf{J}}, (i_A)_{A \in \mathbf{J}})$ and $(M, (q_A)_{A \in \mathbf{J}}, (j_A)_{A \in \mathbf{J}})$ are two ambilimits of (D_1, D_2) , there is a unique isomorphism $M \rightarrow L$ that is both a morphism of cones and cocones.*

Proof. The axioms of ambilimits imply that L and M are dagger limits of (D_1, D_2, \mathbf{J}) , where $(D_1, D_2): \mathbf{J} \rightarrow \mathbf{C}_{\rightleftharpoons}$ and $\mathbf{C}_{\rightleftharpoons}$ is the cofree dagger category on \mathbf{C} from Proposition 2.1.10. Hence there is a unitary isomorphism $M \rightarrow L$ in $\mathbf{C}_{\rightleftharpoons}$ by Theorem 4.3.1. Unpacking the meaning of this back to \mathbf{C} we conclude the existence and uniqueness of the desired isomorphism $M \rightarrow L$. \square

Of course, the preceding proof could be unwound to provide a “dagger-free” proof of the theorem above. It is unclear if doing so would give additional insight or if a different, direct proof exists. Similarly, one could have worked with the more general notion of “ambilimit of (D_1, D_2, Ω) ” where $\Omega \subset \mathbf{J}$ is weakly initial. However, apart from dagger equalizers and other dagger limits, we do not know of interesting examples that would warrant the added generality, so we assume $\Omega = \mathbf{J}$ throughout for simplicity and hence suppress it from the notation. We list some examples below, and the rest of the chapter develops two additional examples in more detail.

Example 5.1.3. • A zero object in \mathbf{C} can be characterized as an ambilimit of the unique functor $\emptyset \rightarrow \mathbf{C}_{\rightleftharpoons}$.

- Splittings of idempotents are ambilimits: let $p = p^2$ be an idempotent in \mathbf{C} and let M be the one-object category generated by a single non-identity arrow e satisfying $e = e^2$. Then $e \mapsto p$ induces a functor $M \rightarrow \mathbf{C}_{\rightleftharpoons}$ and a splitting of p is exactly the same thing as an ambilimit of this functor.
- If \mathbf{G} is a group, considered as a one-object category, then a functor $F: \mathbf{G} \rightarrow \mathbf{C}$ induces a functor $F': \mathbf{G} \rightarrow \mathbf{C}^{\text{op}}$ by $F'(g) = F(g^{-1})$, i.e. $F' = F \circ (-)^{-1}$, where $(-)^{-1}$ is the canonical dagger on \mathbf{G} . Now an ambilimit (F, F') corresponds to a split subobject of $F(*)$ that is simultaneously G -invariant and G -coinvariant. These exist for finite-dimensional k -representations of finite group \mathbf{G} when k is characteristic zero, and they exist for any \mathbf{G} in **Rel**, **Span(FinSet)** and **PInj** by Proposition 4.2.5.
- Both of the previous examples are special cases of the following: let \mathbf{M} be an involutive monoid, i.e. a one object dagger category. Then any functor $\mathbf{M} \rightarrow \mathbf{C}$ factors uniquely via a dagger functor $\mathbf{M} \rightarrow \mathbf{C}_{\rightleftharpoons}$, and one may consider ambilimits of this functor. Again, by Proposition 4.2.5 **Rel**, **Span(FinSet)** and **PInj** have such ambilimits.

5.2 Biproducts without zero morphisms

Given two objects A and B living in some category \mathbf{C} , their biproduct – according to a standard definition [111] – consists of an object $A \oplus B$ together with maps

$$A \begin{array}{c} \xleftarrow{p_A} \\ \xrightarrow{i_A} \end{array} A \oplus B \begin{array}{c} \xleftarrow{i_B} \\ \xrightarrow{p_B} \end{array} B$$

such that

$$p_A i_A = \text{id}_A \qquad p_B i_B = \text{id}_B \qquad (5.2.1)$$

$$p_B i_A = 0_{A,B} \qquad p_A i_B = 0_{B,A} \qquad (5.2.2)$$

and

$$\text{id}_{A \oplus B} = i_A p_A + i_B p_B. \qquad (5.2.3)$$

For us to be able to make sense of the equations, we must assume that \mathbf{C} is enriched in commutative monoids. One can get a slightly more general definition that only

requires zero morphisms but no addition – that is, enrichment in pointed sets – by replacing the last equation with the condition that $(A \oplus B, p_A, p_B)$ is a product of A and B and that $(A \oplus B, i_A, i_B)$ is their coproduct. This definition is not vastly more general, for one can show that if \mathbf{C} has all binary biproducts in this latter sense, then \mathbf{C} is uniquely enriched in commutative monoids and the biproducts in \mathbf{C} are biproducts in the earlier sense as well.

But what if we do not assume that \mathbf{C} is enriched over pointed sets? One alternative would be to postulate, just for the objects A and B in question, maps $0_{A,B}: A \rightarrow B$ and $0_{B,A}: B \rightarrow A$ that act like zero maps in that it does not matter what you pre- and postcompose them with, but this is not very satisfactory – it is as if one tried to generalize the first definition by assuming that just the particular homset $\text{hom}(A \oplus B, A \oplus B)$ happens to be a commutative monoid.

However, by Theorem 5.1.2, one can get a well-behaved notion of a biproduct in any category \mathbf{C} , with no assumptions about enrichment, by replacing the equations referring to zero with the single equation

$$i_A p_A i_B p_B = i_B p_B i_A p_A, \quad (5.2.4)$$

which states that the two canonical idempotents on $A \oplus B$ commute with one another. In this section we show that the notion agrees with the other definitions whenever \mathbf{C} is appropriately enriched. We also show how to characterize them in terms of ambidextrous adjunctions and discuss some examples recognized by the more general definition.

We start by spelling out the new, enrichment-free definition of a biproduct.

Definition 5.2.1. A biproduct of A and B in \mathbf{C} is a tuple $(A \oplus B, p_A, p_B, i_A, i_B)$ such that $(A \oplus B, p_A, p_B)$ is a product of A and B , $(A \oplus B, i_A, i_B)$ is their coproduct, and the following equations hold:

$$\begin{aligned} p_A i_A &= \text{id}_A \\ p_B i_B &= \text{id}_B \\ i_A p_A i_B p_B &= i_B p_B i_A p_A \end{aligned}$$

Even though the definition above does not refer to zero morphisms, the map $p_B i_A: A \rightarrow A \oplus B \rightarrow B$ behaves a lot like one.

Definition 5.2.2. A morphism $a: A \rightarrow B$ is constant if $af = ag$ for all $f, g: C \rightarrow A$. Coconstant morphisms are defined dually and a morphism is called a zero morphism

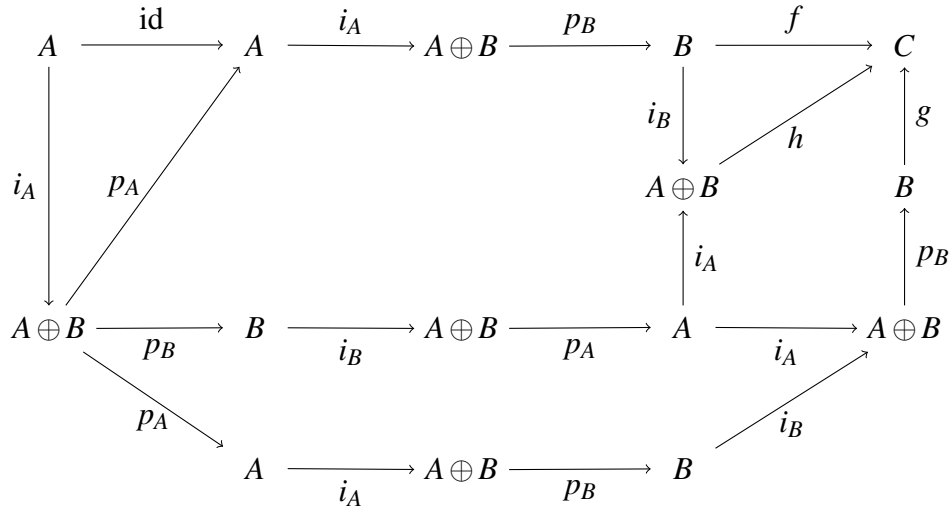
if it is both constant and coconstant. A category has zero morphisms if for every pair of objects A and B there is a zero morphism $A \rightarrow B$.

Remark 5.2.3. If there is a zero morphism $A \rightarrow B$ and a morphism $B \rightarrow A$, then the zero morphism $A \rightarrow B$ is unique. Furthermore, a category has zero morphisms iff for every A and B there is a morphism $0_{A,B}$ such that for every $g: B \rightarrow D$ and $f: C \rightarrow A$ we have $g0_{A,B}f = 0_{C,D}$, and this is equivalent to the category being enriched in pointed sets. The collection of all zero morphisms in a category forms a partial zero structure in the sense of [55].

Proof. Assume first that 0 and $0'$ are zero morphisms $A \rightarrow B$ and let $f: B \rightarrow A$ be an arbitrary morphism. Since 0 and $0'$ are zero morphisms, we see that $0 = 0'f0 = 0'$, as desired. The other claims follow readily. \square

Lemma 5.2.4. Let $(A \oplus B, p_A, p_B, i_A, i_B)$ be the biproduct of A and B . Then $p_B i_A$ is a zero morphism.

Proof. As the definition of biproducts is self-dual, it suffices to prove that $p_B i_A$ is coconstant. First we observe that the diagram



commutes, where h is the cotuple $[gp_B i_A, f]$. Now the top path in the outermost shape is $f p_B i_A$ and by the biproduct equations the bottom path is equal to $g p_B i_A$. \square

Remark 5.2.5. We can now see that Theorem 4.7.8 is still true even if the word biproduct in it is interpreted in the sense of Definition 5.2.1. This is because the only zero morphisms needed for the proof are between A and B , and hence the proof is valid verbatim.

Corollary 5.2.6. *If \mathbf{C} has all binary biproducts, then it has zero morphisms.*

Proof. Combine Lemma 5.2.4 and Remark 5.2.3. \square

Given Lemma 5.2.4, it is easy to check that whenever \mathbf{C} has zero morphisms: equation (5.2.4) is equivalent to equation (5.2.2). Given a biproduct in the sense of Definition 5.2.2, it is a biproduct in the sense of Definition 5.2.1, since $i_A p_A i_B p_B = 0 = i_B p_B i_A p_A$. Conversely, let $(A \oplus B, p_A, p_B, i_A, i_B)$ be a biproduct in the sense of Definition 5.2.1 in a category with zero morphisms. Now by Lemma 5.2.4 and Remark 5.2.3 we have $p_B i_A = 0_{A,B}$, as desired, and similarly $p_A i_B = 0_{B,A}$. If \mathbf{C} is enriched in commutative monoids, then our definition is equivalent to (5.2.3) just because the other definition in terms of (5.2.2) and universal properties is.

Besides being equivalent to the usual definitions when \mathbf{C} is appropriately enriched, it behaves well even when \mathbf{C} is not assumed to be enriched.

Using Lemma 5.2.4, one can then proceed to check that biproducts in our sense work just like one would expect. For example, Definition 5.2.1 and other results of this section generalize from the binary case to the biproduct of an arbitrary-sized collection of objects, and one can easily show that if $(A \oplus B)$ and $(A \oplus B) \oplus C$ exist, then $(A \oplus B) \oplus C$ satisfies the axioms for the ternary biproduct of A, B, C . Similarly, using Lemma 5.2.4 one can show that for $f: A \rightarrow C$ and $g: B \rightarrow D$ we have $f + g = f \times g$ whenever the biproducts $A \oplus B$ and $C \oplus D$ exist.

Recall that an ambiadjoint to a functor $F: \mathbf{C} \rightarrow \mathbf{D}$ is a functor $D: \mathbf{D} \rightarrow \mathbf{C}$ that is simultaneously both left and right adjoint to F .

Theorem 5.2.7. *\mathbf{C} has biproducts iff the diagonal $\Delta: \mathbf{C} \rightarrow \mathbf{C} \times \mathbf{C}$ has an ambiadjoint $(-)\oplus(-)$ such that the unit $(i_A, i_B): (A, B) \rightarrow (A \oplus B, A \oplus B)$ of the adjunction $(-)\oplus(-) \dashv \Delta$, is a section of the counit $(p_A, p_B): (A \oplus B, A \oplus B) \rightarrow (A, B)$ of the adjunction $\Delta \dashv (-)\oplus(-)$, i.e. $(p_A \circ i_A, p_B \circ i_B) = (\text{id}_A, \text{id}_B)$ for $A, B \in \mathbf{C}$.*

Proof. The implication from left to right is routine. For the other direction, a right adjoint to the diagonal is well-known to fix binary products, and dually, a left adjoint fixes binary coproducts. Thus it remains to check that the required equations governing p_A, p_B, i_A and i_B are satisfied. By naturality, the diagram

$$\begin{array}{ccccc}
 A & \xrightarrow{i_A} & A \oplus B & \xrightarrow{p_B} & B \\
 \text{id} \downarrow & & \downarrow \text{id} \oplus f & & \downarrow f \\
 A & \xrightarrow{i_A} & A \oplus C & \xrightarrow{p_C} & C
 \end{array}$$

commutes for any f . Thus $i_A p_B$ is coconstant and by duality it is constant, so it is zero. Hence \mathbf{C} has zero morphisms and $i_A p_B = 0$. Thus $i_A p_A i_B p_B = 0 = i_B p_B i_A p_A$. As $(p_A \circ i_A, p_B \circ i_B) = (\text{id}_A, \text{id}_B)$ by assumption, this concludes the proof. \square

Given Theorem 5.2.6, genuinely new examples recognized under this must be in categories that have neither all binary biproducts nor zero morphisms.

- In \mathbf{Set} the biproduct $\emptyset \oplus \emptyset$ exists and is the empty set.
- Let \mathbf{C} be any category with biproducts, and let \mathbf{D} be any non-empty category. Then in the coproduct category $\mathbf{C} \sqcup \mathbf{D}$, the biproduct $A \oplus B$ exists whenever $A, B \in \mathbf{C}$. More concretely, in $\mathbf{Ab} \sqcup \mathbf{Set}$ the binary biproduct of any two abelian groups exists and is computed just as in \mathbf{Ab} , even though $\mathbf{Ab} \sqcup \mathbf{Set}$ lacks zero morphisms.
- In any preorder $A \oplus B$ exists if and only if $A \cong B$.
- Commutative inverse semigroups are sets equipped with a binary operation that is commutative and associative and in which for every element x there is a unique y such that $xyx = x$ and $yxy = x$ [107]. The obvious choice of morphism is a function that preserves the binary operation. Not every such semigroup has a neutral element, so we call a homomorphism $f: S \rightarrow T$ unital if it preserves neutral elements. Let \mathbf{C} be the category of commutative inverse semigroups and unital homomorphisms. Then $S \oplus T$ exists if and only if S and T both have a neutral element, in which case $S \oplus T$ is constructed just as in \mathbf{Ab} .
- A function $f: (X, d_X) \rightarrow (Y, d_Y)$ between metric spaces is non-expansive if $d_X(x, y) \geq d_Y(f(x), f(y))$ for all $x, y \in X$. It is contractive if there is some $c \in [0, 1)$ such that $cd_X(x, y) \geq d_Y(f(x), f(y))$ for all $x, y \in X$. Let \mathbf{Met} be the category of metric spaces and non-expansive maps and let \mathbf{Con} be the category of contractive endomorphisms. To be more precise, let \mathbb{N} denote the monoid of natural numbers. Then we define \mathbf{Con} as the full subcategory of $[\mathbb{N}, \mathbf{Met}]$ with objects given by contractive endomorphisms. In \mathbf{Con} , the terminal object is $!: \{*\} \rightarrow \{*\}$, and for any s in \mathbf{Con} , the biproduct $s \oplus !$ exists if and only if s has a (necessarily unique) fixed point. To see this, assume first that the biproduct $s \oplus !$ exists. Then in particular there is a morphism $! \rightarrow s$, whence s has a fixed point. Conversely, if s has a fixed point, s itself can be given the structure of a biproduct $s \oplus !$ in a relatively obvious manner.

5.3 Limit-colimit coincidence from domain theory

In this section we will show how to understand the limit-colimit coincidence from domain theory (originally remarked in [128] and further generalized in [135]) as an ambilimit. Before doing so, we briefly recall the setup of [135].

Definition 5.3.1. An \mathbf{O} -category is a category in which (i) every homset is a partial order that has least upper bounds of ω -chains and (ii) composition is ω -continuous.

\mathbf{O} -categories were originally introduced in [149] and are enriched categories in the sense of [101]: the enrichment is over the category \mathbf{O} where objects are posets with l.u.b.s of ω -chains and morphisms are ω -continuous functions. Since we do not need the full generality of enriched category theory, we work with the concrete definition above.

Definition 5.3.2. A pair $(f: A \rightarrow B, g: B \rightarrow A)$ of morphisms in an \mathbf{O} -category is an *embedding-projection pair* if

$$gf = \text{id}_A \text{ and } fg \leq \text{id}_B,$$

in which case we call f an embedding and g a projection.

It turns out that either half of such a pair determines the other: hence for an \mathbf{O} -category \mathbf{C} one can define \mathbf{C}^E , the category of embeddings in \mathbf{C} , as the wide subcategory of \mathbf{C} with embeddings as its morphisms. The category of projections \mathbf{C}^P is defined similarly. In general neither of these is an \mathbf{O} -category. Since one half of a projection-embedding pair determines the other, there are in fact contravariant identity-on-objects isomorphisms $(-)^P: \mathbf{C}^E \leftrightarrow \mathbf{C}^P: (-)^E$ taking one half of a projection-embedding pair to the other one.

Definition 5.3.3. Let \mathbf{C} be an \mathbf{O} -category and $D: \omega \rightarrow \mathbf{C}^E$ be a diagram. A cocone $(L, (e_n)_{n \in \omega})$ for $\omega \xrightarrow{D} \mathbf{C}^E \hookrightarrow \mathbf{C}$ is an *\mathbf{O} -colimit* if each e_n is an embedding, and $(e_n \circ e_n^P)_{n \in \omega}$ is an increasing sequence having id_L as its least upper bound.

Theorem 5.3.4. Let \mathbf{C} be an \mathbf{O} -category and $D: \omega \rightarrow \mathbf{C}^E$ be a diagram. The following conditions are equivalent for any cocone $(L, (e_n)_{n \in \omega})$ for $\omega \xrightarrow{D} \mathbf{C}^E \hookrightarrow \mathbf{C}$:

- (i) L is a colimit in \mathbf{C}
- (ii) L is an \mathbf{O} -colimit in \mathbf{C}

(iii) $(L, (e_n)^P)$ is a limit in \mathbf{C}

(iv) $(L, (e_n)^P)$ is an \mathbf{O} -limit in \mathbf{C}

Proof. See the proof of [135, Theorem 2]. Note that the relevant part of the theorem states a slightly weaker result, saying that D has a colimit iff it has an \mathbf{O} -colimit, but the proof shows in Proposition D that in fact that any colimit is necessarily an \mathbf{O} -colimit. \square

Instead of \mathbf{O} -(co)limits one could just work with the unenriched notion provided by Definition 5.1.1. First we note that the independence axiom is superfluous for ambilimits of chains. For that purpose, we fix some notation. For a chain $(D_1, D_2): \omega^{\text{op}} \rightarrow \mathbf{C}_{\rightleftharpoons}$, we write $D(n)$ for $D_1(n) = D_2(n)$ and q_n for $D_1(n+1 \rightarrow n)$ and e_n for $D_2(n \rightarrow n+1)$. Note that the chain is uniquely defined by the objects $D(n)$ and the morphisms q_n and e_n . More generally, we will write $q_{m,n}$ for $D_1(m \rightarrow n)$ and $e_{n,m}$ for $D_2(n \rightarrow m)$.

Proposition 5.3.5. *Let $(D_1, D_2): \omega^{\text{op}} \rightarrow \mathbf{C}_{\rightleftharpoons}$ be a chain and $(L, (p_n, i_n))$ be a cone for it. If L satisfies normalization, then it satisfies independence.*

Proof. We wish to prove that

$$i_n p_n i_m p_m = i_m p_m i_n p_n$$

follows from the assumptions. We will prove a stronger claim, namely that whenever $n < m$, both sides of this equation are equal to $i_n p_n$. For the left hand side this is done as follows:

$$\begin{aligned} i_n p_n i_m p_m &= i_n q_{m,n} p_m i_m p_m \text{ because } L \text{ is a cone} \\ &= i_n q_{m,n} p_m \text{ by normalization} \\ &= i_n p_n \text{ because } L \text{ is a cone.} \end{aligned}$$

The right hand side is similar:

$$\begin{aligned} i_m p_m i_n p_n &= i_m p_m i_m e_{n,m} p_n \text{ because } L \text{ is a cone} \\ &= i_m e_{n,m} p_n \text{ by normalization} \\ &= i_n p_n \text{ because } L \text{ is a cone.} \end{aligned}$$

\square

Corollary 5.3.6. *Given the assumptions of Theorem 5.3.4, the equivalent conditions (i)-(iv) are equivalent to $(L, (e_n^P, e_n))$ being an ambilimit of $(D^P, D): \omega^{\text{op}} \rightarrow \mathbf{C}_{\rightleftharpoons}$.*

Such ambilimits of chains might exist in categories that are not \mathbf{O} -categories. For instance, they are known to exist in categories enriched over complete metric spaces ([12, 9, 22]). The next section studies another case not falling under either of these.

5.4 Fixed points

The main use of Theorem 5.3.4 is as a building block for models of algebraic datatypes. Algebraic datatypes are usually modelled as initial algebras for certain endofunctors [108]. Using Adámek’s Theorem [7], one can show that every functor preserving \mathbf{O} -colimits has an initial algebra. Moreover, \mathbf{O} -colimits are absolute in the sense that any locally continuous functor preserves them. Hence initial algebras are numerous in \mathbf{O} -categories. An added benefit is that the initial algebras thus built turn out to be canonical fixed points, i.e. also terminal coalgebras. This enables one to both program with these datatypes (via the initial algebra structure) and to reason about them (via the terminal coalgebra structure).

In our case we are working without enrichment, which confers a possible benefit: now one can hope to find suitable models for programming semantics in categories that are not suitably enriched. However, this comes at a price: it is less clear that many functors preserve such ambilimits and hence it is unclear if one can build canonical fixed points for a wide class of functors. In this section we show that, under suitable assumptions, rig-polynomial functors have canonical fixed points. We interleave the discussion with a dagger analogue of the same story. The goal of the dagger version is to develop a theory suitable for languages taking semantics in dagger categories. For instance, the language defined in [125] takes its semantics in $\mathbf{Hilb}_{\leq 1}$, the category of Hilbert spaces and non-expansive linear maps. Moreover, the semantics implicitly use various initial algebras.

We start by covering the usual theory very briefly.

Definition 5.4.1. Let F be an endofunctor on a category \mathbf{C} . An F -algebra is a map $a: F(A) \rightarrow A$, and a morphism of F -algebras $a \rightarrow b$ is a map $f: A \rightarrow B$ in \mathbf{C} making

the square

$$\begin{array}{ccc} F(A) & \xrightarrow{Ff} & F(B) \\ a \downarrow & & \downarrow b \\ A & \xrightarrow{f} & B \end{array}$$

commute. This results in a category of F -algebras, resulting in the notion of an *initial algebra* for an endofunctor. F -coalgebras and terminal coalgebras are defined dually.

Lambek's lemma, a basic result about initial algebras, asserts that whenever $a: F(A) \rightarrow A$ is an initial F -algebra, the morphism a is in fact an isomorphism in \mathbf{C} . This makes the following definition reasonable.

Definition 5.4.2. A canonical fixed point of an endofunctor F is an initial F -algebra a such that a^{-1} is a terminal coalgebra. A *dagger fixed point* for a dagger endofunctor $F: \mathbf{C} \rightarrow \mathbf{C}$ is a canonical fixed point $a: FA \rightarrow A$ such that a is unitary.

Note that the self-duality given by the dagger implies that a^\dagger is a terminal coalgebra. Since Lambek's lemma says that any initial algebra a is an isomorphism, one could equivalently define a dagger fixed point as an initial algebra $a: FA \rightarrow A$ for which a is unitary. Hence we are merely requiring something that is invertible to be dagger invertible.

We refer the reader to [48, 18] for more on canonical fixed points and to the review article [8] for more on fixed points of functors in general. As mentioned before, Adámek's Theorem [7] is used to construct many initial algebras. As is usual in theoretical computer science, we focus on the special case of chains indexed by natural numbers, the generalization to larger ordinals being straightforward.

Theorem 5.4.3. *Let \mathbf{C} be a category that has an initial object 0 and let F be an endofunctor on it. If the chain*

$$0 \rightarrow F(0) \rightarrow FF(0) \rightarrow \dots$$

has a colimit L and F preserves it, then $a: F(L) \rightarrow L$ is an initial F -algebra, where $a: F(L) \rightarrow L$ is the unique morphism making the triangle

$$\begin{array}{ccc} F^n(0) & \xrightarrow{F(i_{n-1})} & F(L) \\ & \searrow i_n & \downarrow a \\ & & L \end{array}$$

commute.

Proof. See e.g. [7] or [135]. \square

Next we prove variants of Adámek's Theorem for ambilimits and for dagger categories. By a chain of isometries we mean a sequence $A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \dots$ where each f_i is a dagger isometry. A dagger colimit of a chain of isometries is a dagger colimit of such a diagram with support all objects in the diagram. One can easily adapt Adámek's Theorem[7] to the dagger setting. Below is a special case for ω -chains, generalizing to ordinals $\alpha > \omega$ is straightforward.

Theorem 5.4.4. *Assume that \mathbf{C} has dagger colimits of chains of isometries and a zero object. If $F: \mathbf{C} \rightarrow \mathbf{C}$ is a dagger functor preserving such colimits, then it has a dagger fixed point.*

Proof. Let (L, l_n) be the dagger colimit of the chain $0 \rightarrow F(0) \rightarrow FF(0) \rightarrow \dots$, and let $a: F(L) \rightarrow L$ be the unique map making the triangle

$$\begin{array}{ccc} F^n(1) & \xrightarrow{F(i_{n-1})} & F(L) \\ & \searrow i_n & \downarrow a \\ & & L \end{array}$$

commute. Then a is an initial algebra by Theorem 5.4.3 and hence a^\dagger is a terminal coalgebra. Moreover, since a is an isomorphism of dagger colimits, it is unitary by Theorem 4.3.1, and hence a is a dagger fixed point. \square

Theorem 5.4.5. *Let \mathbf{C} be a category that has a zero object and F an endofunctor on it. If the chain*

$$0 \begin{array}{c} \longleftarrow \\ \xrightarrow{0} \\ \longleftarrow \end{array} F(0) \begin{array}{c} \xleftarrow{F(0)} \\ \xrightarrow{F(0)} \\ \xleftarrow{F(0)} \end{array} FF(0) \longleftarrow \dots$$

has an ambilimit preserved by F , then F has a canonical fixed point.

Proof. Let $(L, (p_n, i_n))$ be an ambilimit of the chain above. By Theorem 5.4.3 $a: F(L) \rightarrow L$ is an initial algebra and by the dual of the same theorem $b: L \rightarrow F(L)$ is a terminal coalgebra, where a and b are the unique maps making the triangles

$$\begin{array}{ccc} F^n(1) & \xrightarrow{F(i_{n-1})} & F(L) \\ & \searrow i_n & \downarrow a \\ & & L \end{array} \qquad \begin{array}{ccc} F^n(1) & \xleftarrow{F(p_{n-1})} & F(L) \\ & \swarrow p_n & \uparrow b \\ & & L \end{array}$$

commute. It suffices to prove that $b = a^{-1}$. This is easy: a is in fact an isomorphism of colimits and b is an isomorphism of limits, and by Theorem 5.1.2 these coincide. \square

The goal of this section is to leverage Theorems 5.4.4 and 5.4.5 to build canonical fixed points for a wide class of endofunctors. Concrete examples of ambilimits of chains seem to suggest that they tend to exist when D_1 and D_2 are compatible in some sense. For instance, in the case of \mathbf{O} -categories, either half of the chain determines the other half uniquely. We first investigate the case of chains of split monics i.e. chains

$$D(0) \begin{array}{c} \xleftarrow{q_0} \\ \xrightarrow{e_0} \end{array} D(1) \begin{array}{c} \xleftarrow{q_1} \\ \xrightarrow{e_1} \end{array} D(2) \longleftarrow \dots$$

that satisfy $q_n e_n = \text{id}$ for every n . Recall the notation defined before Proposition 5.3.5: For a chain $(D_1, D_2): \omega^{\text{op}} \rightarrow \mathbf{C}_{\rightleftharpoons}$, we write $D(n)$ for $D_1(n) = D_2(n)$ and q_n for $D_1(n+1 \rightarrow n)$ and e_n for $D_2(n \rightarrow n+1)$, and more generally $q_{m,n}$ denotes $D_1(m \rightarrow n)$ and $e_{n,m}$ means $D_2(n \rightarrow m)$.

Theorem 5.4.6. *Given a chain*

$$D(0) \begin{array}{c} \xleftarrow{q_0} \\ \xrightarrow{e_0} \end{array} D(1) \begin{array}{c} \xleftarrow{q_1} \\ \xrightarrow{e_1} \end{array} D(2) \longleftarrow \dots$$

satisfying $q_n e_n = \text{id}_{D(n)}$, the following are equivalent:

(i) *It has an ambilimit $(L, (p_n, i_n))$*

(ii) *The diagram*

$$D(0) \xrightarrow{e_0} D(1) \xrightarrow{e_1} D(2) \longleftarrow \dots$$

has a colimit $(L, (i_n))$ and the diagram

$$D(0) \xleftarrow{q_0} D(1) \xleftarrow{q_1} D(2) \longleftarrow \dots$$

a limit $(M, (q_n))$ such that the canonical morphism $f: L \rightarrow M$ is an isomorphism.

This canonical map is the unique map $f: L \rightarrow M$ making the diagram

$$\begin{array}{ccc} L & \xrightarrow{f} & M \\ i_n \uparrow & & \downarrow q_m \\ D(n) & \xrightarrow{f_{n,m}} & D(m) \end{array}$$

commute, where $f_{n,m}$ is the cone structure on $D(n)$ given by

$$f_{n,m} := \begin{cases} e_{n,m} & \text{if } n \leq m \\ q_{n,m} & \text{otherwise.} \end{cases}$$

Remark 5.4.7. The assumption $q_n e_n = \text{id}_{D(n)}$ is needed to ensure that $f_{n,m}$ defines a cone structure on $D(n)$, which in turn is needed to guarantee the existence of f . Given the theorem, both biproducts (in the presence of zero morphisms) and ambilimits of chains of split monics can be phrased in terms of a canonical map from the colimit to the limit being an isomorphism. Hence one might be tempted to guess that this holds for all ambilimits. However, it is unclear if this holds for arbitrary diagrams: indeed, given an arbitrary diagram $(D_1, D_2): \mathbf{J} \rightarrow \mathbf{C}_{\rightleftharpoons}$ it is unclear why there should be a canonical morphism from the colimit of D_2 to the limit of D_1 .

Proof. To prove that (i) implies (ii) it suffices to show that id_L makes the diagram

$$\begin{array}{ccc} & \text{id} & \\ & \longrightarrow & \\ L & & L \\ i_n \uparrow & & \downarrow p_m \\ D(n) & \xrightarrow{f_{n,m}} & D(m) \end{array}$$

commute. We begin by showing that $p_n i_n = \text{id}_{D(n)}$. This follows from normalization once we show that p_n is epic. In fact it is split epic: the maps $(f_{n,m})_{m \in \mathbb{N}}$ define a cone structure on $D(n)$ which hence must factor via some $g: D(n) \rightarrow L$ (it will follow that $g = i_n$). Hence $\text{id}_{D(n)} = f_{n,n} = p_n g$ so that p_n splits, as desired. Now it is easy to show that the diagram above commutes for all n and m . We show this holds in case $n \geq m$, when $f_{n,m} = q_{n,m}$, the case $n < m$ being similar. Since $p_n i_n = \text{id}_{D(n)}$, the triangle on the left in the diagram

$$\begin{array}{ccccc} D(n) & & & & \\ & \xrightarrow{q_{n,m}} & & & \\ & \searrow \text{id} & & & \\ & & D(n) & \xrightarrow{q_{n,m}} & D(m) \\ & \nearrow p_n & & & \\ L & & & & \\ & & & \nearrow p_m & \end{array}$$

commutes. The triangle on the top commutes by definition and the bottom triangle commutes because $(L, (p_k)_{k \in \mathbb{N}})$ is a cone. Hence the whole diagram commutes, as desired.

To prove that (ii) implies (i), note that defining $p_n := (q_n \circ f)$ makes (L, p_n) into a limit. Hence by Proposition 5.3.5 it suffices to check that $(L, (p_n, i_n))$ satisfies normalization. But this is immediate from the fact that $p_n i_n = q_n f i_n = f_{n,n} = \text{id}_{D(n)}$. \square

Corollary 5.4.8. *If $(L, (p_n, i_n))$ is the ambilimit of a chain*

$$[D(0) \begin{array}{c} \xleftarrow{q_0} \\ \xrightarrow{e_0} \end{array} D(1) \begin{array}{c} \xleftarrow{q_1} \\ \xrightarrow{e_1} \end{array} D(2) \longleftarrow \dots]$$

satisfying $q_n e_n = \text{id}_{D(n)}$, then the composite $p_m e_n : D(n) \rightarrow L \rightarrow D(m)$ satisfies

$$p_m e_n = \begin{cases} e_{n,m} & \text{if } n \leq m \\ q_{n,m} & \text{otherwise.} \end{cases}$$

In the setting of \mathbf{O} -categories, for a functor to preserve \mathbf{O} -colimits it is enough for it to be locally continuous, i.e. an enriched functor. In the unenriched setting of this chapter, we don't have as convenient a sufficient condition to ensure preservation of ambilimits of chains. However, it turns out that for chains of split monics, it is enough to preserve the universal property of the limit (or of the colimit), provided that the resulting diagram admits an ambilimit. In particular, being (co)continuous will often ensure preservation of ambilimits of suitable chains, even if in principle an ambilimit is both a limit and a colimit.

Lemma 5.4.9. *Let*

$$D(0) \begin{array}{c} \xleftarrow{q_0} \\ \xrightarrow{e_0} \end{array} D(1) \begin{array}{c} \xleftarrow{q_1} \\ \xrightarrow{e_1} \end{array} D(2) \longleftarrow \dots$$

be a chain in \mathbf{C} satisfying $q_n e_n = \text{id}_{D(n)}$ with an ambilimit $(L, (p_n), (i_n))$, and let $F : \mathbf{C} \rightarrow \mathbf{D}$ be a functor. If

- *the chain*

$$FD(0) \begin{array}{c} \xleftarrow{Fq_0} \\ \xrightarrow{Fe_0} \end{array} FD(1) \begin{array}{c} \xleftarrow{Fq_1} \\ \xrightarrow{Fe_1} \end{array} FD(2) \longleftarrow \dots$$

has an ambilimit and

- *(FL, Fi_n) is a colimit of the chain*

$$FD(0) \xrightarrow{Fe_0} FD(1) \xrightarrow{Fe_1} FD(2) \dots,$$

then $(FL, (Fp_n), (Fi_n))$ is in fact an ambilimit.

Proof. Let $(L', (p'_n), (i'_n))$ be the ambilimit of

$$FD(0) \begin{array}{c} \xleftarrow{Fq_0} \\ \xrightarrow{Fe_0} \end{array} FD(1) \begin{array}{c} \xleftarrow{Fq_1} \\ \xrightarrow{Fe_1} \end{array} FD(2) \longleftarrow \dots$$

Since (FL, Fi_n) is a colimit of the chain

$$FD(0) \xrightarrow{Fe_0} FD(1) \xrightarrow{Fe_1} FD(2) \dots \rightarrow$$

there is a unique isomorphism of colimits $f: (FL, Fi_n) \rightarrow (L, i_n)$, and this makes $(FL, p'_n \circ f, Fi_n)$ into an ambilimit. Moreover, the top triangle of

$$\begin{array}{ccc} FL & \xrightarrow{p'_m \circ f} & FD(m) \\ Fi_n \uparrow & \nearrow Ff_{n,m} & \uparrow Fp_m \\ FD(n) & \xrightarrow{Fi_n} & FL \end{array}$$

commutes by Corollary 5.4.8 (since $(FL, p'_n \circ f, Fi_n)$ is an ambilimit in \mathbf{D}), as does the bottom triangle (since L is an ambilimit in \mathbf{C}). Hence the universal property of the colimit gives us $p'_n \circ f = Fp_n$, so that $(FL, (Fp_n), (Fi_n))$ is an ambilimit as desired. \square

Definition 5.4.10. A rig category¹ is a category \mathbf{C} with a symmetric monoidal structure $(\mathbf{C}, \oplus, 0)$ and a monoidal structure (\mathbf{C}, \otimes, I) . The coherence isomorphisms of the \oplus -part will have components denoted by

$$\begin{array}{ll} \alpha_{A,B,C}^{\oplus}: (A \oplus B) \oplus C \rightarrow A \oplus (B \oplus C) & \lambda_A^{\oplus}: 0 \oplus A \rightarrow A \\ \rho_A^{\oplus}: 0 \oplus A \rightarrow A & \sigma_{A,B}^{\oplus}: A \oplus B \rightarrow B \oplus A \end{array}$$

Similarly, the coherence isomorphisms for (\mathbf{C}, \otimes, I) will be denoted by $\alpha^{\otimes}, \lambda^{\otimes}, \rho^{\otimes}$ and σ^{\otimes} . The two monoidal structures should of course interact. Concretely, this means that we have

- Left and right distributors δ^l and δ^r with components

$$\begin{array}{l} \delta_{A,B,C}^l: A \otimes (B \oplus C) \cong (A \otimes B) \oplus (A \otimes C) \\ \delta_{A,B,C}^r: (A \oplus B) \otimes C \cong (A \otimes C) \oplus (B \otimes C) \end{array}$$

- And left and right annihilators τ^l and τ^r with components

$$\tau_A^l: A \otimes 0 \cong 0 \quad \tau_A^r: 0 \otimes A \cong 0$$

¹Also known as a bimonoidal category

These coherence isomorphisms can be seen as categorifications of axioms for a rig (semi-ring). This data is supposed to satisfy various coherence laws, see [100] and [105] for details.

In the sequel all rig categories are assumed to be symmetric without further warning. This means that also the monoidal category (\mathbf{C}, \otimes, I) is symmetric in manner coherent with the rest of the structure. A *rig dagger category* is a rig category with a dagger such that all the coherence isomorphisms are unitary and moreover both monoidal structures are dagger functors $\mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$.

Example 5.4.11. • **Pfn**, the category of sets and partial functions, is a rig category: \oplus is given by the disjoint union and \otimes by the cartesian product (in **Set**).

- **Pinj** is a rig dagger category with the same interpretations for \oplus and \otimes .
- **Hilb** is a rig dagger category, when \oplus is given by the dagger biproduct and \otimes by the tensor product. The subcategory **Hilb** $_{\leq 1}$ of Hilbert spaces and non-expansive linear maps inherits a rig dagger structure from this, even though \oplus no longer satisfies a universal property.
- **Ban** $_{\leq 1}$, the category of Banach spaces and non-expansive linear maps, is a rig category, when \oplus is interpreted as the ℓ^1 -sum, i.e. the coproduct of **Ban** $_{\leq 1}$ and \otimes is the projective tensor product. Formally, this is the completion of the algebraic tensor product $A \otimes B$ under the norm

$$\|x\| = \inf \left\{ \sum_{k=1}^n \|a_k\| \|b_k\| \mid x = \sum_{k=1}^n a_k \otimes b_k \right\}$$

We won't define in detail a notion of a morphism of rig categories. Roughly speaking they are functors that preserve the rig structure up to coherent natural isomorphisms (or transformations, depending on whether one wants the stronger or weaker notion). Intuitively, it is clear that the inclusions **Pinj** \rightarrow **Pfn** and **Hilb** $_{\leq 1}$ \rightarrow **Hilb** satisfy this.

More interestingly, there are functors $\ell^2: \mathbf{Pinj} \rightarrow \mathbf{Hilb}_{\leq 1}$ and $\ell^1: \mathbf{Pfn} \rightarrow \mathbf{Ban}_{\leq 1}$ and these turn out to be compatible with the rig structure. The functor ℓ^2 is defined on objects by

$$A \mapsto \ell^2 A := \{ \phi: A \rightarrow \mathbb{C} \mid \sum |\phi(A)|^2 < \infty \}$$

and its action on a partial injection $f: A \rightarrow B$ is given by

$$\ell^2(f)(\phi)(b) := \sum_{a \in f^{-1}b} \phi(a) = \begin{cases} \phi(f^\dagger a) & \text{if } f^\dagger a \text{ is defined,} \\ 0 & \text{otherwise.} \end{cases}$$

The functor ℓ^2 is known to preserve the dagger and the two monoidal structures \oplus and \otimes , see [72].

The functor ℓ^1 is defined similarly: an object A is sent to the Banach space

$$\ell^1 A := \{\phi: A \rightarrow \mathbb{C} \mid \sum |\phi(A)| < \infty\}$$

and a partial function $f: A \rightarrow B$ is sent to the contraction given by

$$\ell^1(f)(\phi)(b) := \sum_{a \in f^{-1}b} \phi(a).$$

One usually thinks of ℓ^1 as a functor $\mathbf{Set} \rightarrow \mathbf{Ban}_{\leq 1}$, in which case it is the left adjoint to the unit ball functor $U: \mathbf{Ban}_{\leq 1} \rightarrow \mathbf{Set}$ defined by $U(B) := \{v \in B \mid \|v\| \leq 1\}$.

Proposition 5.4.12. *The functor $\ell^1: \mathbf{Pfn} \rightarrow \mathbf{Ban}_{\leq 1}$ preserves the rig structure, i.e. we have natural isomorphisms*

$$\ell^1(A \oplus B) \cong \ell^1(A) \oplus \ell^1(B) \quad \text{and} \quad \ell^1(A \otimes B) \cong \ell^1(A) \otimes \ell^1(B).$$

Proof. Write $i: \mathbf{Set} \rightarrow \mathbf{Pfn}$ for the usual inclusion. Then $\ell^1 \circ i$ has a right adjoint U and hence preserves colimits. Thus $\ell^1(A \oplus B) = \ell^1(i(A + B)) \cong \ell^1(A) \oplus \ell^1(B)$ as desired. For \otimes , we calculate in terms of Schauder bases: $\ell^1(A)$ has a canonical Schauder basis indexed by A : the basis vector corresponding to $a \in A$ is given by the Dirac delta δ_a defined by

$$\delta_a(b) := \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise.} \end{cases}$$

Hence $\ell^1(A \otimes B)$ has a canonical basis indexed by $A \times B$. Consider the function sending the basis element $\delta_{(a,b)}$ to $\delta_a \otimes \delta_b$. This induces an isometric isomorphism from the algebraic span of $\{\delta_{(a,b)} \mid (a,b) \in A \times B\}$ to the algebraic span of $\{\delta_a \otimes \delta_b \mid (a,b) \in A \times B\}$. Since both of these are dense subspaces, this induces an isometric isomorphism $\ell^1(A \otimes B) \rightarrow \ell^1(A) \otimes \ell^1(B)$, i.e. an isomorphism in $\mathbf{Ban}_{\leq 1}$. \square

The functors ℓ^1 and ℓ^2 in fact preserve much more structure. To make this precise, we will formalize a wide class of chains that always have ambilimits in our categories

of interest. A naive attempt at formalizing this might be to require the existence of ambilimits of chains of split monics. However, this is asking for too much: given a chain of split monics, there might be many chains of epis splitting them, and only some pairs of chains might admit an ambilimit.

A concrete example of this is the following. Let $D_2: \omega \rightarrow \mathbf{Pfn}$ be given by

$$(n \leq m) \mapsto \{1, \dots, n\} \hookrightarrow \{1, \dots, m\}.$$

The “correct” way of splitting the inclusion $\{1, \dots, n\} \hookrightarrow \{1, \dots, m\}$ is via the map

$$k \mapsto \begin{cases} k & \text{if } k \leq n \\ \text{undefined,} & \text{otherwise} \end{cases}$$

since this results in a chain D_1 such that (D_1, D_2) has an ambilimit given by \mathbb{N} . If instead one splits each inclusion by the map $k \mapsto \min\{k, n\}$, the limit is $\mathbb{N} \cup \{\infty\}$ and the canonical map is the inclusion $\mathbb{N} \rightarrow \mathbb{N} \cup \{\infty\}$ which is not an isomorphism.

To avoid pathologies like this, we need restrict to a smaller class of diagrams that we require to have ambilimits. Note that in our categories of interest, the unit for \oplus is in fact a zero object. This gives rise to canonical projections and inclusions

$$A \begin{array}{c} \xleftarrow{p_A} \\ \xrightarrow{i_A} \end{array} A \oplus B \begin{array}{c} \xleftarrow{i_B} \\ \xrightarrow{p_B} \end{array} B$$

subject to the biproduct equations (5.2.1) and (5.2.2), even though $A \oplus B$ need not satisfy a universal property (indeed, in $\mathbf{Hilb}_{\leq 1}$ it does not). The projection $p_B: A \oplus B \rightarrow B$ can be defined as the composite $\lambda_B^{\oplus} \circ (0 \oplus \text{id}): A \oplus B \rightarrow 0 \oplus B \cong B$, and the other morphisms are defined similarly.

To keep in line with domain theoretic terminology, we call the pair (i_A, p_A) an embedding-projection pair. Similarly, a chain $(D_1, D_2): \omega^{\text{op}} \rightarrow \mathbf{C}_{\rightleftharpoons}$ is called an *embedding-projection chain* if it is of the form

$$A_0 \begin{array}{c} \xleftarrow{p_0} \\ \xrightarrow{i_0} \end{array} A_0 \oplus A_1 \begin{array}{c} \xleftarrow{p_{0,1}} \\ \xrightarrow{i_{0,1}} \end{array} A_0 \oplus A_1 \oplus A_2 \longleftarrow \dots$$

Similarly, chains of the form

$$A_0 \xrightarrow{i_0} A_0 \oplus A_1 \xrightarrow{i_{0,1}} A_0 \oplus A_1 \oplus A_2 \rightarrow \dots$$

are called chains of embeddings and chains of the form

$$A_0 \xleftarrow{p_0} A_0 \oplus A_1 \xleftarrow{p_{0,1}} A_0 \oplus A_1 \oplus A_2 \leftarrow \dots$$

are called chains of projections.

Definition 5.4.13. A (dagger) rig category is ω -continuous if

- The monoidal unit 0 is in fact a zero object.
- It has ambilimits of embedding-projection chains. (It has dagger colimits of chains of isometries.)
- The functors $X \mapsto A \oplus X$ and $X \mapsto A \otimes X$ preserve them.

It is not immediately obvious from the definition that an ω -continuous rig dagger category is ω -continuous as a rig category. However, it is easy to show that this holds. Because (\mathbf{C}, \oplus) is dagger monoidal, the canonical inclusions and projections are dagger monics and dagger epimorphisms respectively. Hence every chain of embeddings has a dagger colimit, and the normalization and independence equations guarantee that it is in fact an ambilimit for the corresponding embedding-projection chain.

The main reason for defining ω -continuous rig categories is that they provide a natural setting for building canonical fixed points for polynomial functors. We give the general theory before verifying that the categories we have studied so far in this section are ω -continuous.

Definition 5.4.14. Let \mathbf{C} be a rig category. A *rig-polynomial functor* on \mathbf{C} is a functor of the form

$$X \mapsto \left(\bigoplus_{i=0}^n A_i \otimes X^{\otimes i} \right)$$

If \mathbf{C} is a a rig dagger category that has (anti)symmetrized tensor powers, then a generalized polynomial endofunctor is defined as a rig-polynomial endofunctor, except that some of the powers $X^{\otimes i}$ can be replaced by (anti-)symmetrized tensor powers from example 4.2.4.

Note that when $n = 0$ we have $X^{\otimes n} \cong 1$ by convention.

Lemma 5.4.15. *The functors $B \oplus -$, $B \otimes -$ and $X \mapsto X^{\otimes n}$ preserve embedding-projection chains up to isomorphism. In a rig dagger category, they preserve chains of embeddings up to unitary isomorphism.*

Proof. It is clear that applying the endofunctor $B \oplus -$ to an embedding-projection chain results in an embedding-projection chain. This is also true $B \otimes -$ since

$$B \otimes A_0 \begin{array}{c} \xleftarrow{\text{id} \otimes p_0} \\ \xrightarrow{\text{id} \otimes i_0} \end{array} B \otimes (A_0 \oplus A_1) \begin{array}{c} \xleftarrow{\text{id} \otimes p_{0,1}} \\ \xrightarrow{\text{id} \otimes i_{0,1}} \end{array} B \otimes (A_0 \oplus A_1 \oplus A_2) \longleftarrow \dots$$

is isomorphic to

$$B_0 \begin{array}{c} \xleftarrow{p_0} \\ \xrightarrow{i_0} \end{array} B_0 \oplus B_1 \begin{array}{c} \xleftarrow{p_{0,1}} \\ \xrightarrow{i_{0,1}} \end{array} B_0 \oplus B_1 \oplus B_2 \xleftarrow{\quad} \dots$$

where $B_i = B \otimes A_i$. Finally, an easy calculation shows that applying $X \mapsto X^{\otimes n}$ to the chain

$$A_0 \begin{array}{c} \xleftarrow{p_0} \\ \xrightarrow{i_0} \end{array} A_0 \oplus A_1 \begin{array}{c} \xleftarrow{p_{0,1}} \\ \xrightarrow{i_{0,1}} \end{array} A_0 \oplus A_1 \oplus A_2 \xleftarrow{\quad} \dots$$

results in a chain isomorphic to

$$B_0 \xleftarrow{\quad} B_1 \xleftarrow{\quad} B_2 \xleftarrow{\quad} \dots$$

where

$$B_i = \bigoplus_{f: \{1, \dots, n\} \rightarrow \{0, \dots, i\}} A_{f(1)} \otimes \dots \otimes A_{f(n)}$$

and the embedding-projection pair $B_i \rightarrow B_{i-1}$ comes from the inclusion $\{0, \dots, i\} \hookrightarrow \{0, \dots, i+1\}$. In a rig dagger category, all of these isomorphisms are unitary. \square

Note that the functor $X \mapsto X^{\oplus n}$ is naturally isomorphic to $X \mapsto (I^{\oplus n} \otimes X)$ and hence automatically preserves ambilimits of embedding-projection chains in an ω -continuous rig category. More surprisingly, the functor $X \mapsto X^{\otimes n}$ also preserves such ambilimits.

Lemma 5.4.16. *Let \mathbf{C} be an ω -continuous rig category. Then the functor $X \mapsto X^{\otimes n}$ preserves ambilimits of embedding-projection chains for any $n \in \mathbb{N}$. If \mathbf{C} is an ω -continuous rig dagger category, then the functor $X \mapsto X^{\otimes n}$ preserves dagger colimits of chains of isometries for any $n \in \mathbb{N}$.*

Proof. Since $X \mapsto X^{\otimes n}$ preserves embedding-projection chains by Lemma 5.4.15 it suffices to show that it preserves colimits of chains of embeddings by Lemma 5.4.9.

So let $D: \omega \rightarrow \mathbf{C}$ be such a chain. Now we have canonical isomorphisms

$$\begin{aligned} (\operatorname{colim}_i D(i))^{\otimes n} &= (\operatorname{colim}_{i_1} D(i_1)) \otimes (\operatorname{colim}_i D(i))^{\otimes n-1} \\ &\cong \operatorname{colim}_{i_1} (D(i_1) \otimes (\operatorname{colim}_i D(i))^{\otimes n-1}) \text{ by } \omega\text{-continuity of } \mathbf{C} \\ &\cong \dots \cong \operatorname{colim}_{i_1, \dots, i_n} D(i_1) \otimes \dots \otimes D(i_n) \end{aligned}$$

so it suffices to prove that

$$\operatorname{colim}_{i_1, \dots, i_n} D(i_1) \otimes \dots \otimes D(i_n) \cong \operatorname{colim}_i D(i)^{\otimes n}.$$

This is not hard to show directly, but follows readily from more general principles. It is easy to check that the diagonal functor $\omega \rightarrow \omega^n$ is final in the sense of [111, IX.3] (since its image is cofinal in the order-theoretic sense) and hence the colimits agree by [111, Theorem IX.3.1].

For the dagger case, note that the calculation above shows that a colimit of isometries is taken to a colimit of isometries by $X \mapsto X^{\otimes n}$. Since $X \mapsto X^{\otimes n}$ is a dagger functor it automatically preserves cocones satisfying normalization and independence, and hence it also preserves *dagger* colimits of isometries. \square

In a sense, the proof above boils down to a special case of the general fact that directed colimits can be built from colimits of chains, see [10, Corollary 1.7].

Theorem 5.4.17. *If \mathbf{C} is an ω -continuous rig category, then every rig-polynomial endofunctor has a canonical fixed point.*

If \mathbf{C} is an ω -continuous rig dagger category, then every generalized rig-polynomial dagger endofunctor has a dagger fixed point.

Proof. It is easy to check that ω -continuous rig categories with zero objects are closed under countable products and that any tupling of functors preserving ambilimits of embedding-projection chains also preserves such ambilimits. Moreover, projection functors and constant functors preserve such ambilimits. As any rig-polynomial functor can be built as a composite of these and the basic building blocks $X \mapsto A \otimes X$, $X \mapsto X^{\otimes n}$ and $X \mapsto A \oplus X$, it suffices to show that the basic building blocks preserve ambilimits of embedding-projection chains: $X \mapsto A \oplus X$ and $X \mapsto A \otimes X$ do so by assumption and $X \mapsto X^{\otimes n}$ does so by Lemma 5.4.16. Hence any rig-polynomial functor satisfies the assumptions of Theorem 5.4.5 and hence has a canonical fixed point. In the dagger case, one uses Theorem 5.4.4, and the argument above can be easily adapted to the dagger case. The only new thing to check is that (anti)symmetrized tensor powers preserve colimits of chains of isometries. This is the case by Theorem 4.8.5, since (anti)symmetrized tensor products are dagger limits of diagrams with only unitary morphisms, and hence the adjointability condition is automatically satisfied. Thus any generalized rig-polynomial endofunctor satisfies the assumptions of Theorem 5.4.4 and hence has a dagger fixed point. \square

Theorem 5.4.18. *The categories \mathbf{PInj} and $\mathbf{Hilb}_{\leq 1}$ are ω -continuous rig dagger categories. \mathbf{Pfn} and $\mathbf{Ban}_{\leq 1}$ are ω -continuous rig categories. Moreover, the functor $\ell^1: \mathbf{Pfn} \rightarrow \mathbf{Ban}_{\leq 1}$ preserves ambilimits of chains of embeddings.*

Proof. We start with **Pinj**. Clearly the empty set is both a zero object and a monoidal unit for disjoint unions. Moreover, **Pinj** is known to have colimits of chains [72, Proposition 3.6]: for a chain of isometries $D: \omega \rightarrow \mathbf{Pinj}$ the colimit L can be calculated as

$$L := \coprod D(i) / \approx$$

where the coproduct is taken in **Set** and the equivalence relation \approx is generated by $x \approx D(n \rightarrow n+1)x$. Hence it remains to check that $X \mapsto A \oplus X$ and $X \mapsto A \otimes X$ preserve colimits of such chains. This follows easily once one observes that a chain of isometries in **Pinj** is also a chain of monomorphisms in **Set** and that the construction of the colimit agrees in both categories. Since the disjoint union is a colimit in **Set** and colimits commute with colimits, $X \mapsto A \oplus X$ preserve colimits of chains of isometries. The functor $X \mapsto A \otimes X$ preserves such colimits since \otimes is the categorical product in set and filtered colimits commute with finite limits in **Set** [111, Theorem IX.2.1].

Next we move on to $\mathbf{Hilb}_{\leq 1}$. Clearly the zero-dimensional Hilbert space is both a zero object and a monoidal unit for the direct sum. It is known to be ω_1 -accessible [10, Example 2.3.9], which implies that it has colimits of isometries. Moreover, the functor $\ell^2: \mathbf{Pinj} \rightarrow \mathbf{Hilb}_{\leq 1}$ preserves the rig structure up to (unitary) isomorphism and preserves colimits of chains of isometries, see e.g. [72] and [18].

To prove that constant multiplication and addition preserve these colimits, an explicit description in terms of a (Schauder) basis will be useful. Let $D: \omega \rightarrow \mathbf{Hilb}_{\leq 1}$ be a chain of isometries. We may identify $D(n)$ with a closed subspace of $D(n+1)$, and using this identification, choose a Schauder basis I_n for each $D(n)$ such that $I_n \subset I_{n+1}$. Then the colimit of D is characterized by having a basis $\bigcup_n I_n$. From this description it is easy to check that the colimit is in fact a dagger colimit. Instead of computing directly in $\mathbf{Hilb}_{\leq 1}$, we use the functor ℓ^2 : the functor $n \mapsto I_n$ defines a functor $D': \omega \rightarrow \mathbf{Pinj}$ and moreover there is an obvious unitary isomorphism between D and $\ell^2 \circ D'$. Hence any colimit of a chain of isometries can be first computed in **Pinj**, whence ω -continuity of **Pinj** implies that of $\mathbf{Hilb}_{\leq 1}$.

In **Pfn** the colimit

$$A_0 \hookrightarrow A_0 \oplus A_1 \hookrightarrow A_0 \oplus A_1 \oplus A_2 \hookrightarrow \dots$$

is built as in **Pinj** (or **Set**) and is given by the disjoint union $L := \coprod_k A_k$ with the inclusions $i_k: \bigoplus_{i=1}^k A_i \rightarrow L$. Define projections $p_k: L \rightarrow \bigoplus_{i=1}^k A_i$, using the universal

property of the *coproduct* by defining

$$A_n \hookrightarrow L \xrightarrow{p_k} \bigoplus_{i=1}^k A_i := \begin{cases} A_n \hookrightarrow \bigoplus_{i=1}^k A_i & \text{if } n \leq k \\ A_n \xrightarrow{0} \bigoplus_{i=1}^k A_i & \text{otherwise.} \end{cases}$$

It is easy to check that $(L, (p_n, i_n))$ satisfies normalization, so it suffices to check that $(L, (p_n))$ is a limit of

$$A_0 \xleftarrow{p_0} A_0 \oplus A_1 \xleftarrow{p_{0,1}} A_0 \oplus A_1 \oplus A_2 \leftarrow \dots$$

So, consider a cone $(f_k : X \rightarrow \bigoplus_{i=1}^k A_i)_k$. Define $f : X \rightarrow L$ by

$$f(x) = \begin{cases} i_k f_k(x), & \text{where } k \text{ is least such that } f_k(x) \text{ is defined} \\ \text{undefined,} & \text{if no such } k \text{ exists.} \end{cases}$$

It is now straightforward to check that f satisfies $p_k f = f_k$ and is unique in that respect. Hence **Pfn** has ambilimits of chains of embeddings. Next we prove that the functors $X \mapsto A \oplus X$ and $X \mapsto A \otimes X$ preserve them. As both of these functors preserve embedding-projection chains by Lemma 5.4.15, it is enough to show that both of these functors preserve the universal property of the colimit by Lemma 5.4.9. This is done as in the case of **PInj**: $X \mapsto A \oplus X$ is cocontinuous since colimits commute with colimits, whereas $X \mapsto A \otimes X$ preserves colimits of countable chains since the colimit of a chain **Pfn** is computed as in **Set** and filtered colimits commute with finite limits in **Set** [111, Theorem IX.2.1].

For **Ban**_{≤1}, we first note that **Ban**_{≤1} is complete and cocomplete. It is locally presentable by [10, Example 1.48], hence cocomplete by definition and complete by [10, Remark 1.56]: alternatively, this has been proven directly in [134].

The colimit of

$$A_0 \xrightarrow{i_0} A_0 \oplus A_1 \xrightarrow{i_{0,1}} A_0 \oplus A_1 \oplus A_2 \rightarrow \dots$$

is in fact the coproduct $\coprod_k A_k$ and the limit of

$$A_0 \xleftarrow{p_0} A_0 \oplus A_1 \xleftarrow{p_{0,1}} A_0 \oplus A_1 \oplus A_2 \leftarrow \dots$$

can be seen via the usual construction as a certain subobject of the product $\prod_k (\bigoplus_{j=1}^k A_j)$. Since products in **Ban**_{≤1} are ℓ^∞ -sums, this is concretely given by the vector space

$$B := \{(b_k)_{k \in \mathbb{N}} \mid b_k \in \bigoplus_{j=1}^k A_j, \sup_k \|b_k\| < \infty, p_{0, \dots, k}(b_{k+1}) = b_k\}$$

with the ℓ^∞ -norm $\|(b_k)_{k \in \mathbb{N}}\| := \sup_k \|b_k\|$. The condition $p_{0, \dots, k}(b_{k+1}) = b_k$ guarantees that the sequence can be equivalently viewed as a sequence $(a_k)_{k \in \mathbb{N}}$ with $a_k \in A_k$. Since $\bigoplus_{j=1}^k A_k$ has the ℓ^1 -norm, the condition $\sup \|b_k\| < \infty$ then implies that

$$\sum_k \|a_k\| = \sup_k \sum_{j=1}^k \|a_j\| = \sup_k \|b_k\| < \infty$$

which amounts to saying that $(a_k)_{k \in \mathbb{N}} \in \coprod_k A_k$. This amounts to us having constructed an isometric inverse to the canonical map $\coprod_k A_k \rightarrow B$, so $\mathbf{Ban}_{\leq 1}$ has ambilimits of chains of embeddings by Theorem 5.4.6.

Next we need to show that the functors $X \mapsto A \oplus X$ and $X \mapsto A \otimes X$ preserve ambilimits of embedding-projection chains. Since colimits commute with colimits, $X \mapsto A \oplus X$ is cocontinuous. Because $\mathbf{Ban}_{\leq 1}$ is closed (see e.g. [35, Chapter II]), the functor $X \mapsto A \otimes X$ is also cocontinuous. Since both of these functors preserve embedding-projection chains, they preserve ambilimits by Lemmas 5.4.15 and 5.4.9.

For ℓ^1 , note first that the composite $\mathbf{Set} \hookrightarrow \mathbf{Pfn} \xrightarrow{\ell^1} \mathbf{Ban}_{\leq 1}$ is a left adjoint and hence cocontinuous. Since colimits of split monics are computed in \mathbf{Pfn} as they are in \mathbf{Set} and moreover ℓ^1 cooperates with the rig structure and hence preserves embedding-projection chains, it preserves ambilimits by Lemma 5.4.9.

□

One might be tempted to prove cocontinuity of $X \mapsto A \oplus X$ in $\mathbf{Hilb}_{\leq 1}$ by referring to Theorem 4.8.5. Unfortunately, this does not quite work as-is: the direct sum $H \oplus K$ of Hilbert spaces is a dagger product in \mathbf{Hilb} but not in $\mathbf{Hilb}_{\leq 1}$, whereas the colimits of isometries are colimits in the latter but not in the former category. However, one can get around this and provide an alternative proof as follows: the colimits of $\mathbf{Hilb}_{\leq 1}$ are *bounded* colimits in \mathbf{Hilb} , meaning that they satisfy a universal property for every jointly bounded cocone. One could then develop a theory of bounded dagger colimits and show that Theorem 4.8.5 still holds. We have chosen not to do so since doing so would take us too far afield. Moreover the proof above highlights an interesting similarity between \mathbf{PInj} and $\mathbf{Hilb}_{\leq 1}$, where both cases boil down to (ordinary) polynomial functors over \mathbf{Set} having initial algebras.

One could avoid a direct proof for $\mathbf{Hilb}_{\leq 1}$ by leveraging properties of \mathbf{PInj} and ℓ^2 . Similarly, one might hope to simplify the proof above by reducing the case of $\mathbf{Ban}_{\leq 1}$ to properties of \mathbf{Pfn} and ℓ^1 . However, there is a crucial difference that makes this impossible: every Hilbert space is unitarily isomorphic to a space of the form $\ell^2(A)$, but not every Banach space is isometrically isomorphic to a space of the form $\ell^1(A)$.

$\mathbf{Ban}_{\leq 1}$ is enriched over itself and hence also over metric spaces. However, ambilimits of embedding-projection chains in $\mathbf{Ban}_{\leq 1}$ do not fall under the theory covered in [12, 9, 22]: they wish the chain D to satisfy $\|e_n q_n - \text{id}\| \rightarrow 0$ as $n \rightarrow \infty$, whereas embedding-projection chains typically satisfy $\|e_n q_n - \text{id}\| = 1$ for all n . This corresponds with the fact that, for the ambilimit in question, the sequence $i_n p_n$ converges to id_L only strongly but not in the operator norm.

Example 5.4.19. • The Hilbert space $\ell^2(\mathbb{N})$ is a dagger fixed point for the endofunctor $-\oplus \mathbb{C}$. Indeed, by construction, the dagger fixed point is given as the dagger colimit of the sequence $\mathbb{C} \rightarrow \mathbb{C}^2 \rightarrow \mathbb{C}^3 \rightarrow \dots$, where $\mathbb{C}^n \rightarrow \mathbb{C}^{n+1}$ is given by inclusion into the first n coordinates. This colimit is given by $\ell^2(\mathbb{N})$. Being a dagger colimit of this sequence explains why the inclusions $i_n: \mathbb{C} \rightarrow \ell^2(\mathbb{N})$ satisfy infinitary biproduct equations, while not being a biproduct. This also clarifies the universal property of $\ell^2(\mathbb{N})$, which is weaker than that of a biproduct. This discussion lifts more generally to ℓ^2 -sums of any cardinality (though for a κ -ary sum one needs a κ -filtered colimit).

- Quantum lists of type A can be modelled as $\bigoplus_{i=1}^{\infty} A^{\otimes i}$, i.e. as the dagger fixed point for the functor $\mathbb{C} \oplus (A \otimes -)$.
- Quantum trees of type A can be modelled as the dagger initial algebra for $A \oplus (-)^{\otimes 2}$

Note that all of the polynomial endofunctors extend to \mathbf{Hilb} , and admit unitary solutions to $F(X) \cong X$. However, these solutions no longer satisfy a universal property. For instance, the codiagonal $\mathbb{C}^2 \rightarrow \mathbb{C}$ is an algebra of $-\oplus \mathbb{C}$, but there is no algebra homomorphism from $\ell^2(\mathbb{N})$ to it.

The symmetrized Fock space of type A is defined as $\bigoplus_{n \in \mathbb{N}} \text{Sym}^n A$ and the antisymmetrized Fock space is defined as $\bigoplus_{n \in \mathbb{N}} \text{Sym}_{-1}^n A$. Since the infinitary sum \bigoplus is the ℓ^2 -sum, which can be seen as a dagger colimit of a chain of isometries, one might hope to understand Fock spaces as dagger fixed points. This would result in another categorical understanding of Fock spaces, in contrast with [23, 47, 146]. Unfortunately, I have not been able to make this work. The issue is the following: while $\bigoplus_{n \in \mathbb{N}} A^{\otimes n}$ can be built as the dagger fixed point of $X \mapsto I \oplus (A \otimes -)$, there is no obvious way to replace the term $(A \otimes -)$ with something that would generate higher (anti)symmetrized powers of A . This is a shame since one would also like to use the Fock space of type A to model unordered lists i.e. sets of type A .

5.5 Traces

In this section we recall some background material on traced symmetric monoidal categories and then go on to show that the “standard sum formula” defines a trace on $\mathbf{Ban}_{\leq 1}$, with the functor $\ell^1: \mathbf{Pfn} \rightarrow \mathbf{Ban}_{\leq 1}$ being traced. We start by recalling the definition of a traced symmetric monoidal category [91].

Definition 5.5.1. A trace on a symmetric monoidal category consists of a family of functions $\mathrm{Tr}_{A,B}^U: \mathbf{C}(A \otimes U, B \otimes U) \rightarrow \mathbf{C}(A, B)$ satisfying the following axioms:

- Naturality in A and B : for all $g: A \otimes U \rightarrow B \otimes U$, $f: A' \rightarrow A$ and $h: B \rightarrow B'$,

$$h \circ \mathrm{Tr}_{A,B}^U(g) \circ f = \mathrm{Tr}_{A',B'}^U((h \otimes \mathrm{id}_U) \circ g \circ (f \otimes \mathrm{id}_U)).$$

- Dinaturality in U : for all $f: A \otimes U \rightarrow B \otimes U'$ and $g: U' \rightarrow U$,

$$\mathrm{Tr}_{A,B}^U((\mathrm{id}_B \otimes g) \circ f) = \mathrm{Tr}_{A,B}^{U'}(f \circ (\mathrm{id}_A \otimes g)).$$

- Strength: for all $f: A \otimes U \rightarrow B \otimes U$ and $g \otimes C \otimes D$,

$$g \otimes \mathrm{Tr}_{A,B}^U(f) = \mathrm{Tr}_{C \otimes A, C \otimes B}^U(g \otimes f).$$

- Vanishing I: for all $f: A \otimes I \rightarrow B \otimes I$,

$$f = \mathrm{Tr}_{A,B}^I(f).$$

- Vanishing II: for all $f: A \otimes U \otimes V \rightarrow B \otimes U \otimes V$,

$$\mathrm{Tr}_{A,B}^{U \otimes V}(f) = \mathrm{Tr}_{A,B}^U(\mathrm{Tr}_{A \otimes U, B \otimes U}^V(f)).$$

- Yanking: for all A ,

$$\mathrm{Tr}_{A,A}^A(\sigma_{A,A}) = \mathrm{id}_A.$$

Graphically the trace is usually depicted as follows:

$$\mathrm{Tr}_{A,B}^U \left(\begin{array}{c} B \quad U \\ | \quad | \\ \boxed{f} \\ | \quad | \\ A \quad U \end{array} \right) =: \begin{array}{c} B \\ | \\ \boxed{f} \\ | \\ A \end{array}$$

With this convention at hand, the axioms for a trace become quite natural. The reader is invited to either draw the axioms on their own or to consult one of [91, 131] for more details.

In a rig category $(\mathbf{C}, \oplus, \otimes)$ there are two monoidal products, either one of which might be traced. In such a setting, one often calls a trace on (\mathbf{C}, \oplus) “sum-like” or “particle-like”, especially when \oplus is an actual coproduct. Similarly, a trace on (\mathbf{C}, \otimes) is called a “product-like trace” or a “wave-like trace”, especially when \otimes is a categorical product. See [62, 1, 5] for further discussion.

In our context, we are primarily interested in particle-like traces. Even when \oplus isn’t a biproduct (let alone a coproduct), one might have enough structure to express each morphism $f: A \oplus U \rightarrow B \oplus U$ as a matrix

$$\begin{bmatrix} f_{A,B}: A \rightarrow B & f_{U,B}: U \rightarrow B \\ f_{A,U}: A \rightarrow U & f_{U,U}: U \rightarrow U \end{bmatrix}$$

and then hope to define a trace by the “standard trace formula”

$$\mathrm{Tr}_{A,B}^U(f) := f_{A,B} + \sum_{n \in \mathbb{N}} f_{U,B} \circ f_{U,U}^n \circ f_{A,U},$$

provided the sum is defined. In so-called unique decomposition categories, if these sums are always defined, then the standard formula defines a trace [59, Proposition 4.0.11]. Moreover, even when this sum isn’t always defined, it always defines a *partial trace*, (introduced [61], studied further in [112]). In fact every strong unique decomposition category can be embedded in a totally traced unique decomposition category [81].

The results on unique decomposition categories give sum-like traces for **Pfn** and **PInj**, and we cite these results below. Before doing so, we recall how sums are interpreted in these categories. In **Pfn**, a sum $\sum_{i \in I} f_i$ is defined iff the functions f_i have disjoint domains of definition, in which case the sum is defined to be the union of the morphisms f_i . In **PInj**, a sum $\sum_{i \in I} f_i$ is defined iff the functions f_i have disjoint domains of definition *and* disjoint images, in which case the sum is defined to be the union of the f_i .

Proposition 5.5.2 ([60]). **Pfn** and **PInj** have sum-like traces, given by the standard trace formula (5.5).

However, even if maps admit a matrix representation as above, and (some) infinite sums make sense, the standard formula need not define a trace. For instance, the standard trace formula does not define a (partial) trace on $(\mathbf{Vect}_{\mathbb{C}}, \oplus)$, when the sum is interpreted as weak convergence [112, Proposition 3.10].

Note that in **Ban**_{≤1} the underlying vector space of $A \oplus B$ is a biproduct in $\mathbf{Vect}_{\mathbb{C}}$ and hence morphisms $f: A \oplus U \rightarrow B \oplus U$ admit a matrix representation. Hence one might

wonder if the trace-formula defines a (partial) trace in $\mathbf{Ban}_{\leq 1}$. Given that the standard trace formula does not work in $(\mathbf{Vect}_{\mathbb{C}}, \oplus)$, it might be surprising that it does work in $\mathbf{Ban}_{\leq 1}$ when the sums involved are interpreted as convergent sums in the strong operator topology.

Proposition 5.5.3. $\mathbf{Ban}_{\leq 1}$ has a sum-like trace defined by standard trace formula (5.5). Moreover, the functor $\ell^1: \mathbf{Pfn} \rightarrow \mathbf{Ban}_{\leq 1}$ is traced.

Proof. First we will show that the sum in the formula (5.5) always converges strongly to a weak contraction. Since f is weakly contractive and the monoidal product we are working over is given by the ℓ^1 -sum, for any $v \in A$ and $k \in U$ we have

$$\|v\| \geq \|f_{A,B}v\| + \|f_{A,U}v\| \quad \|w\| \geq \|f_{U,B}w\| + \|f_{U,U}w\|$$

Piecing these together, we see that for any v in A and $k \in \mathbb{N}$ we have

$$\begin{aligned} \|v\| &\geq \|f_{A,B}v\| + \|f_{A,U}v\| \geq \|f_{A,B}v\| + \|f_{U,B}f_{A,U}v\| + \|f_{U,U}f_{A,U}v\| \\ &\geq \cdots \geq \|f_{A,B}v\| + \sum_{n=0}^k \|f_{U,B}f_{U,U}^n f_{A,U}v\| \end{aligned}$$

This implies that the series $\sum_{n \in \mathbb{N}} f_{U,B} \circ f_{U,U}^n \circ f_{A,U}v$ is absolutely convergent and hence convergent. Moreover, the triangle inequality gives us

$$\|f_{A,B}v\| + \sum_{n=0}^k \|f_{U,B}f_{U,U}^n f_{A,U}v\| \geq \left\| f_{A,B} + \sum_{n=0}^k f_{U,B} \circ f_{U,U}^n \circ f_{A,U} \right\|$$

whence

$$\|v\| \geq \|f_{A,B}v\| + \sum_{n \in \mathbb{N}} \|f_{U,B}f_{U,U}^n f_{A,U}v\| \geq \left\| f_{A,B} + \sum_{n \in \mathbb{N}} f_{U,B} \circ f_{U,U}^n \circ f_{A,U} \right\| = \|\mathrm{Tr}_{A,B}^U(f)v\|,$$

as desired.

Proving the axioms for the trace amounts to straightforward calculations. In fact, the proof of [59, Proposition 4.0.11] works verbatim, even though the proposition itself does not apply since $\mathbf{Ban}_{\leq 1}$ is not a so-called unique decomposition category in an obvious manner.

We conclude by proving that the functor ℓ^1 is traced. Since ℓ^1 preserves coproducts and hence the matrix representation of a morphism, it suffices to prove that it preserves countable sums, i.e. that whenever $\sum_{n \in \mathbb{N}} f_n$ is defined in \mathbf{Pfn} , the sum $\sum_{n \in \mathbb{N}} \ell^1(f_n)$ converges strongly to $\ell^1(\sum_{n \in \mathbb{N}} f_n)$.

So, let $f_n : A \rightarrow B$ be given. For the sum $\sum_{n \in \mathbb{N}} f_n$ to be defined, the partial functions f_n must have disjoint domain. Write $A_n := \text{dom } f_n$ and $A_{-1} := A \setminus \bigcup_{n \in \mathbb{N}} A_n$. For $k \geq 1$ define $p_n : \ell^1(A) \rightarrow \ell^1(A)$ by

$$p_n(\varphi)(a) = \begin{cases} \varphi(a) & \text{if } a \in A_n \\ 0 & \text{otherwise.} \end{cases}$$

Since $\{A_k | k \geq -1\}$ is a partition of A , we see that $\sum_{k \geq 1} p_k = \text{id}$ in the strong operator topology. Moreover, $\ell^1(\sum_{n \in \mathbb{N}} f_n)p_k = \ell^1(f_k)$ for any $k \geq 0$, and $\ell^1(\sum_{n \in \mathbb{N}} f_n)p_{-1} = 0$. Hence

$$\ell^1\left(\sum_{n \in \mathbb{N}} f_n\right) = \ell^1\left(\sum_{n \in \mathbb{N}} f_n\right) \sum_{k \geq -1} p_k = \sum_{k \geq -1} \left(\ell^1\left(\sum_{n \in \mathbb{N}} f_n\right)p_k\right) = \sum_{k \in \mathbb{N}} \ell^1(f_k),$$

as desired. \square

5.6 A language for traced ω -continuous rig categories

In this section we show how to interpret the Π^o -language studied in [26, 33, 89] in any traced ω -continuous rig category.

We start from the syntax, by listing our syntactic values and value types

$$\begin{array}{ll} \text{(Value Types)} & a := 0 \mid 1 \mid (a_1 \oplus a_2) \mid (a_1 \otimes a_2) \mid x \mid \mu x.a \\ \text{(Values)} & v := () \mid \text{inl } v \mid \text{inr } v \mid (v, v) \mid \langle v \rangle \end{array}$$

The values are typed according to the following rules.

$$\frac{}{(): 1} \quad \frac{v : a \quad w : b}{(v, w) : a \otimes b} \quad \frac{v : a}{\text{inl}(v) : a \oplus b} \quad \frac{v : a}{\text{inr}(v) : b \oplus a} \quad \frac{v : a[\mu x.a/x]}{\langle v \rangle : \mu x.a}$$

Where the notation $a[b/x]$ means the type obtained by replacing each free occurrence of x by b . Thinking categorically, our value types corresponds to objects. Besides value types, our language has combinators, which then correspond to morphisms. The collection of combinator types is given by $a \leftrightarrow b$ for value types a, b . We write \leftrightarrow instead of \rightarrow because our combinators are intended to be partial isomorphisms. The notation $f : a \leftrightarrow b : g$ means that f is of type $a \leftrightarrow b$ and g is of type $b \leftrightarrow a$. The basic

combinators, along with their types are given below.

$$\begin{array}{ll}
\text{id}_a: a \leftrightarrow a & \\
\text{assocl}^{\oplus}_{a,b,c}: a \oplus (b \oplus c) \leftrightarrow (a \oplus b) \oplus c: \text{assocr}^{\oplus}_{a,b,c} & \\
\text{unitl}^{\oplus}_a: 0 \oplus a \leftrightarrow a: \text{unitr}^{\oplus}_a & \\
\text{swap}^{\oplus}_{a,b}: a \oplus b \leftrightarrow b \oplus a & \\
\text{assocl}^{\otimes}_{a,b,c}: a \otimes (b \otimes c) \leftrightarrow (a \otimes b) \otimes c: \text{assocr}^{\otimes}_{a,b,c} & \\
\text{unitl}^{\otimes}_a: 1 \otimes a \leftrightarrow a: \text{unitr}^{\otimes}_a & \\
\text{swap}^{\otimes}_{a,b}: a \otimes b \leftrightarrow b \otimes a & \\
\text{distr}_{a,b,c}: (a \oplus b) \otimes c \leftrightarrow (a \otimes c) \oplus (b \otimes c): \text{factor}_{a,b,c} & \\
\text{absorb}_a: 0 \otimes a \leftrightarrow 0: \text{unabsorb}_a & \\
\text{fold}_{\mu x.a}: a[\mu x.a/x] \leftrightarrow \mu x.a: \text{unfold}_{\mu x.a} &
\end{array}$$

Basic combinators can be combined, using \circ , \oplus and \otimes . The typing rules for the combinators built using these are as one might guess:

$$\frac{f: a \leftrightarrow b \quad g: b \leftrightarrow c}{g \circ f: a \leftrightarrow c} \quad \frac{f: a \leftrightarrow b \quad g: c \leftrightarrow d}{f \oplus g: a \oplus c \leftrightarrow b \oplus d} \quad \frac{f: a \leftrightarrow b \quad g: c \leftrightarrow d}{f \otimes g: a \otimes c \leftrightarrow b \otimes d}$$

Moreover, we have a trace operator Tr , together with the typing rule

$$\frac{f: a \oplus c \leftrightarrow b \oplus c}{\text{Tr}(f): a \leftrightarrow c}$$

We list the full description of the syntax in a single table to make it easier to refer back to.

(Value Types)	$a := 0 \mid 1 \mid (a_1 \oplus a_2) \mid (a_1 \otimes a_2) \mid x \mid \mu x.a$
(Values)	$v := () \mid \text{inl } v \mid \text{inr } v \mid (v, v) \mid \langle v \rangle$
(Combinator Types)	$a_1 \leftrightarrow a_2$
(Basic Combinators)	$g := \text{id} \mid \text{assocl}^{\oplus} \mid \text{assocr}^{\oplus} \mid \text{unitl}^{\oplus} \mid \text{unitr}^{\oplus} \mid$ $\text{swap}^{\oplus} \mid \text{assocl}^{\otimes} \mid \text{assocr}^{\otimes} \mid \text{unitl}^{\otimes} \mid \text{unitr}^{\otimes} \mid \text{swap}^{\otimes} \mid$ $\text{distr} \mid \text{factor} \mid \text{absorb} \mid \text{unabsorb} \mid \text{fold} \mid \text{unfold}$
(Combinators)	$f := g \mid f_1 \circ f_2 \mid f_1 \oplus f_2 \mid f_1 \otimes f_2 \mid \text{Tr}(f)$

Next we move on to semantics. Given a traced ω -continuous rig category \mathbf{C} , it is now easy to define inductively an interpretation $\llbracket - \rrbracket$ of Π^o into \mathbf{C} . This will then be a functor from the syntactic category to the category \mathbf{C} , at least as long as we interpret

the syntactic category as the category that has *closed types* as its objects, i.e. those types with no free occurrences of variables x . However, to interpret $\mu x.b$ we need to be able to interpret variables x . Formally, we need a finite assignment (function) $s = \{x_1 \mapsto A_1, \dots, x_n \mapsto A_n\}$. For such a function s , we define $s[x \mapsto A]$ to mean the function defined by

$$s[x \mapsto A](y) = \begin{cases} A & \text{if } y = x \\ s(y) & \text{otherwise.} \end{cases}$$

Then, we define a partial interpretation $\llbracket - \rrbracket_s$ on all value types inductively.

$$\llbracket 0 \rrbracket_s = 0_{\mathbf{C}}$$

$$\llbracket 1 \rrbracket_s = 1_{\mathbf{C}}$$

$$\llbracket a \oplus b \rrbracket_s = \llbracket a \rrbracket_s \oplus \llbracket b \rrbracket_s$$

$$\llbracket a \otimes b \rrbracket_s = \llbracket a \rrbracket_s \otimes \llbracket b \rrbracket_s$$

$$\llbracket x \rrbracket_s = \begin{cases} s(x) & \text{if defined,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

$$\llbracket \mu x.b \rrbracket_s = \text{the canonical fixed point of the polynomial functor } A \mapsto \llbracket b \rrbracket_{s[x \mapsto a]}$$

Next we extend $\llbracket - \rrbracket_s$ to basic combinators. To avoid cluttering the notation, we omit s from the indices.

$$\llbracket \text{id}_a \rrbracket = \text{id}_{\llbracket a \rrbracket}$$

$$\llbracket \text{assocr}^{\oplus}_{a,b,c} \rrbracket = \alpha^{\oplus}_{\llbracket a \rrbracket, \llbracket b \rrbracket, \llbracket c \rrbracket}$$

$$\llbracket \text{unitl}^{\oplus}_a \rrbracket = \lambda^{\oplus}_{\llbracket a \rrbracket}$$

$$\llbracket \text{swap}^{\oplus}_{a,b} \rrbracket = \sigma^{\oplus}_{\llbracket a \rrbracket}$$

$$\llbracket \text{assocr}^{\otimes}_{a,b,c} \rrbracket = \alpha^{\otimes}_{\llbracket a \rrbracket, \llbracket b \rrbracket, \llbracket c \rrbracket}$$

$$\llbracket \text{unitl}^{\otimes}_a \rrbracket = \lambda^{\otimes}_{\llbracket a \rrbracket}$$

$$\llbracket \text{swap}^{\otimes}_{a,b} \rrbracket = \sigma^{\otimes}_{\llbracket a \rrbracket}$$

$$\llbracket \text{distr}_{a,b,c} \rrbracket = \delta^l_{\llbracket a \rrbracket, \llbracket b \rrbracket, \llbracket c \rrbracket}$$

$$\llbracket \text{absorb}_a \rrbracket = \tau^r_{\llbracket a \rrbracket}$$

$$\llbracket \text{fold} \rrbracket_{\mu x.a} = \text{the canonical isomorphism } \llbracket a[\mu x.a/x] \rrbracket \rightarrow \llbracket \mu x.a \rrbracket$$

$$\llbracket \text{assocl}^{\oplus}_{a,b,c} \rrbracket = \llbracket \text{assocr}^{\oplus}_{a,b,c} \rrbracket^{-1}$$

$$\llbracket \text{unitr}^{\oplus}_a \rrbracket = \llbracket \text{unitl}^{\oplus}_a \rrbracket^{-1}$$

$$\llbracket \text{assocl}^{\otimes}_{a,b,c} \rrbracket = \llbracket \text{assocr}^{\otimes}_{a,b,c} \rrbracket^{-1}$$

$$\llbracket \text{unitr}^{\otimes}_a \rrbracket = \llbracket \text{unitl}^{\otimes}_a \rrbracket^{-1}$$

$$\llbracket \text{factor}_{a,b,c} \rrbracket = \llbracket \text{distr}_{a,b,c} \rrbracket^{-1}$$

$$\llbracket \text{unabsorb}_a \rrbracket = \llbracket \text{absorb}_a \rrbracket^{-1}$$

$$\text{unfold}_{\mu x.a} = \llbracket \text{fold} \rrbracket_{\mu x.a}^{-1}$$

The extension to all combinators is as one would guess:

$$\begin{aligned} \llbracket (g \oplus f) \rrbracket &= \llbracket g \rrbracket \oplus \llbracket f \rrbracket & \llbracket g \circ f \rrbracket &= \llbracket g \rrbracket \circ \llbracket f \rrbracket \\ \llbracket (g \otimes f) \rrbracket &= \llbracket g \rrbracket \otimes \llbracket f \rrbracket & \llbracket \text{Tr}(f) \rrbracket &= \text{Tr}(\llbracket f \rrbracket) \end{aligned}$$

For types without free variables the choice of an assignment does not matter. An easy induction shows the following.

Proposition 5.6.1. *If $f: a \leftrightarrow b$ can be derived, then $\llbracket f \rrbracket: \llbracket a \rrbracket \rightarrow \llbracket b \rrbracket$.*

Proposition 5.6.2. *If \mathbf{C} and \mathbf{D} are traced ω -continuous rig categories and $F: \mathbf{C} \rightarrow \mathbf{D}$ is a traced ω -continuous rig functor, then $\llbracket - \rrbracket_{\mathbf{D}} \cong F \circ \llbracket - \rrbracket_{\mathbf{C}}$*

The last proposition implies that in each of our examples, the semantics $\llbracket - \rrbracket$ factors via **Pinj**. This is not very surprising, since the language is designed to capture reversible features, and indeed it is logically reversible [89, Proposition 4.2]. However, this might also be seen as a drawback or as a sign of low expressive power. Certainly if one wishes to use a variant of the language for naming morphisms in $\mathbf{Hilb}_{\leq 1}$ or $\mathbf{Ban}_{\leq 1}$ one might hope to customize the language to suit better either of these categories. Both of these categories might benefit from having more inbuilt functions (reversible or not) coming with given interpretations and perhaps new ways of combining them. We conclude by discussing how to possibly change the language, treating the changes suggested by $\mathbf{Hilb}_{\leq 1}$ and $\mathbf{Ban}_{\leq 1}$ separately.

From the construction, it is clear that dropping parts of the syntax corresponds to dropping requirements for the models. For instance, if one drops the trace operator from the language, the result is then readily interpretable in $\mathbf{Hilb}_{\leq 1}$. One could then add new operations e.g. superpositions of values, to get a language better suited for discussing quantum programming in $\mathbf{Hilb}_{\leq 1}$. Indeed, this is essentially what is done in [125]. Moreover, one could easily add a primitive operator $(-)^{\dagger}$ with the rule

$$\frac{f: a \leftrightarrow b}{f^{\dagger}: b \leftrightarrow a}$$

and extend the semantics by $\llbracket f^{\dagger} \rrbracket = \llbracket f \rrbracket^{\dagger}$. Of course, every combinator already has a canonical dagger as noticed in [89], but this need not be the case for further inbuilt combinators unless one adds a primitive like $(-)^{\dagger}$.

One might even hope to extend the full language to $\mathbf{Hilb}_{\leq 1}$ by finding an appropriate (partial?) trace on it, so that ℓ^2 becomes a traced functor. However, I have not been

able to do so: ℓ^2 being traced seems to strongly suggest using the sum formula, but it is unclear if it works in $\mathbf{Hilb}_{\leq 1}$. It is easy to find examples showing that it's not always defined in contrast with $\mathbf{Ban}_{\leq 1}$ but I haven't been able to modify the counterexample of [112, Proposition 3.10] to $\mathbf{Hilb}_{\leq 1}$ in order to show that it does not define a valid partial trace. Alternatively, one might hope to show conclusively that no such trace exists, by e.g. adapting the results of [82]. However, those results apply to Abelian categories and hence don't work as-is for either of $\mathbf{Hilb}_{\leq 1}$ or \mathbf{Hilb} . Hence I haven't been able to resolve the question whether there is a partial trace on $\mathbf{Hilb}_{\leq 1}$ making ℓ^2 traced. Besides being a shortcoming in research, this might also be taken as a further sign of problems with the idea of quantum control (see e.g. [16] for a discussion).

For $\mathbf{Ban}_{\leq 1}$, the most obvious way of extending the language is by adding a richer family of inbuilt functions. However, one might also add function types since $\mathbf{Ban}_{\leq 1}$ is closed. To do this successfully, one should first check that the closure works well with the canonical fixed points, which we leave for further work.

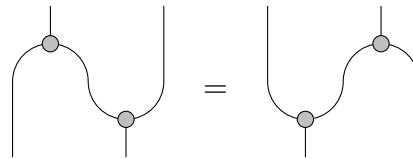
In both of these cases, while one might get a better syntactic handle of the categories in question, one is unlikely to get an internal language in the sense of an equivalence between models and syntactic theories. This is essentially because the syntactic $\mu x.a$ acts more like a "chosen fixed point of a functor" rather than as one satisfying a universal property. With no further inbuilt functions and types, this is probably not a problem but if the user is free to add types and functions, there is no obvious guarantee that the type $\mu x.a$ is an initial algebra in the syntactic category.

Chapter 6

Monads

6.1 Introduction

If T is a monad and a dagger functor, then it is automatically also a comonad. We contend that the theory of a monad on a dagger category works best when the monad and comonad satisfy the following *Frobenius law*, depicted in a graphical calculus reviewed in Section 2.2.



This law has the following satisfactory consequences.

- Any dagger adjunction induces a monad satisfying the law; see Section 6.2.
- The Kleisli category of a monad that preserves daggers and satisfies the law inherits a dagger; see Section 6.3.
- For such a monad, the category of those Eilenberg-Moore algebras that satisfy the law inherits a dagger; see Section 6.3.
- In fact, this Kleisli category and Frobenius-Eilenberg-Moore category are the initial and final resolutions of such a monad as dagger adjunctions in the dagger 2-category of dagger categories; see Section 6.4.
- Any monoid in a monoidal dagger category satisfying the law induces a monad satisfying the law; see Section 6.2.

- Moreover, the adjunction between monoids and strong monads becomes an equivalence in the dagger setting; see Section 6.5.

Additionally, Section 6.6 characterizes the Frobenius law as a natural coherence property between the dagger and closure of a monoidal category.

Because of these benefits, it is tempting to simply call such monads ‘dagger monads’. However, many of these results also work without daggers, see [140, 106, 28]. This chapter is related to those works, but not a straightforward extension. Daggers and monads have come together before in coalgebra [87, 86], quantum programming languages [57, 133], and matrix algebra [44]. The current chapter differs by taking the dagger into account as a fundamental principle from the beginning. Finally, Section 6.5 is a noncommutative generalization of [121, Theorem 4.5]. It also generalizes the classic Eilenberg-Watts theorem [150], that characterizes certain endofunctors on abelian categories as being of the form $- \otimes B$ for a monoid B , to monoidal dagger categories; note that there are monoidal dagger categories that are not abelian [71, Appendix A].

6.2 Frobenius monads

We now come to our central notion: the *Frobenius law* for monads. It is the dagger version of a similar notion in [140]. The monads of [140] correspond to ambijunctions, whereas our monads correspond to dagger adjunctions.

Definition 6.2.1. A *dagger Frobenius monad* on a dagger category \mathbf{C} is a dagger Frobenius monoid in $[\mathbf{C}, \mathbf{C}]$; explicitly, a monad (T, μ, η) on \mathbf{C} with $T(f^\dagger) = T(f)^\dagger$ and

$$T(\mu_A) \circ \mu_{T(A)}^\dagger = \mu_{T(A)} \circ T(\mu_A^\dagger). \quad (6.2.1)$$

The following family is our main source of examples of dagger Frobenius monads. We will see in Example 6.3.5 below that it includes quantum measurement.

Example 6.2.2. A monoid (B, \circ, \circ) in a monoidal dagger category \mathbf{C} is a dagger Frobenius monoid if and only if the monad $- \otimes B: \mathbf{C} \rightarrow \mathbf{C}$ is a dagger Frobenius monad.

Proof. The monad laws become the monoid laws.

$$\mu_A = \begin{array}{c} A \\ | \\ A \end{array} \quad \begin{array}{c} B \\ | \\ \circ \\ \curvearrowright \\ B \end{array} \quad \eta_A = \begin{array}{c} A \\ | \\ A \end{array} \quad \begin{array}{c} B \\ | \\ \circ \end{array}$$

The Frobenius law of the monoid implies the Frobenius law of the monad:

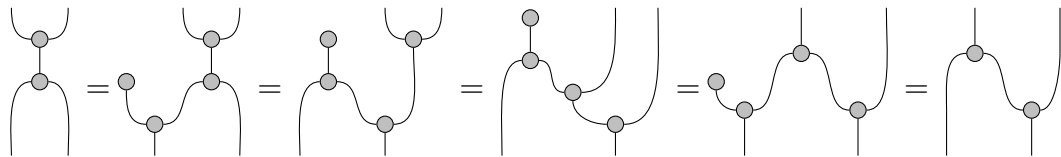
$$T\mu \circ \mu^\dagger_T = \begin{array}{c} \begin{array}{ccc} A & B & B \\ | & | & | \\ \text{---} & \text{---} & \text{---} \\ | & | & | \\ A & B & B \end{array} \\ \text{---} \\ \begin{array}{ccc} A & B & B \\ | & | & | \\ \text{---} & \text{---} & \text{---} \\ | & | & | \\ A & B & B \end{array} \end{array} = \begin{array}{c} \begin{array}{ccc} A & B & B \\ | & | & | \\ \text{---} & \text{---} & \text{---} \\ | & | & | \\ A & B & B \end{array} \\ \text{---} \\ \begin{array}{ccc} A & B & B \\ | & | & | \\ \text{---} & \text{---} & \text{---} \\ | & | & | \\ A & B & B \end{array} \end{array} = \mu_T \circ T\mu^\dagger.$$

The converse follows by taking $A = I$. □

For another example: if T is a dagger Frobenius monad on a dagger category \mathbf{C} , and \mathbf{D} is any other dagger category, then $T \circ -$ is a dagger Frobenius monad on $[\mathbf{D}, \mathbf{C}]$.

Lemma 6.2.3. *If T is a dagger Frobenius monad on a dagger category, $\mu^\dagger \circ \mu = \mu T \circ T\mu^\dagger$.*

Proof. The following graphical derivation holds for any dagger Frobenius monoid.



These equalities use the unit law, the Frobenius law, and associativity. □

The following lemma shows that dagger Frobenius monads have the same relationship to dagger adjunctions as ordinary monads have to ordinary adjunctions.

Lemma 6.2.4. *If $F \dashv G$ is a dagger adjunction, then $G \circ F$ is a dagger Frobenius monad.*

Proof. It is clear that $T = G \circ F$ is a dagger functor. The Frobenius law follows from applying [106, Corollary 2.22] to **DagCat**. We will be able to give a self-contained proof after Theorem 6.3.9 below. □

For example, in **Rel** and **Hilb**, the dagger biproduct monad induced by the dagger adjunction of Example 3.2.2 is of the form $- \otimes (I \oplus I)$ as in Example 6.2.2. However, not all dagger Frobenius monads are of this form: the Frobenius monad induced by the dagger adjunction of Example 3.2.3 in general decreases the dimension of the underlying space, and hence cannot be of the form $- \otimes B$ for a fixed B .

6.3 Algebras

Next we consider algebras for dagger Frobenius monads. We start by showing that Kleisli categories of dagger Frobenius monads inherit a dagger.

Lemma 6.3.1. *If T is a dagger Frobenius monad on a dagger category \mathbf{C} , then $\mathcal{Kl}(T)$ carries a dagger that commutes with the canonical functors $\mathcal{Kl}(T) \rightarrow \mathbf{C}$ and $\mathbf{C} \rightarrow \mathcal{Kl}(T)$.*

Proof. A straightforward calculation establishes that

$$(A \xrightarrow{f} T(B)) \mapsto (B \xrightarrow{\eta} T(B) \xrightarrow{\mu^\dagger} T^2(B) \xrightarrow{T(f^\dagger)} T(A))$$

is a dagger on $\mathcal{Kl}(T)$ commuting with the functors $\mathbf{C} \rightarrow \mathcal{Kl}(T)$ and $\mathcal{Kl}(T) \rightarrow \mathbf{C}$. \square

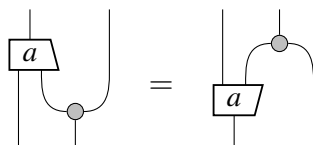
If we want algebras to form a dagger category, it turns out that the category of all Eilenberg-Moore algebras is too large. The crucial law is the following *Frobenius law* for algebras.

Definition 6.3.2. Let T be a monad on a dagger category \mathbf{C} . A *Frobenius-Eilenberg-Moore algebra*, or *FEM-algebra* for short, is an Eilenberg-Moore algebra $a: T(A) \rightarrow A$ that makes the following diagram commute.

$$\begin{array}{ccc} T(A) & \xrightarrow{T(a)^\dagger} & T^2(A) \\ \mu^\dagger \downarrow & & \downarrow \mu \\ T^2(A) & \xrightarrow{T(a)} & T(A) \end{array} \tag{6.3.1}$$

Denote the category of FEM-algebras (A, a) and algebra homomorphisms by $\text{FEM}(T)$.

In effect, equation (6.3.1) says that the map $T(a) \circ \mu^\dagger: T(A) \rightarrow T^2(A) \rightarrow T(A)$ is self-adjoint. When the dagger Frobenius monad is of the form $T(A) = A \otimes B$ as in Example 6.2.2, the Frobenius law (6.3.1) for an algebra $a: T(A) \rightarrow A$ becomes the following equation, which resembles the Frobenius law (2.2.1) for monoids and monads.



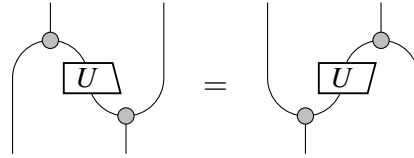
Example 6.3.3. Any free algebra $\mu_A: T^2(A) \rightarrow T(A)$ of a dagger Frobenius monad T on a dagger category \mathbf{C} is a FEM-algebra. Hence there is an embedding $\mathcal{K}\ell(T) \rightarrow \text{FEM}(T)$.

Proof. The Frobenius law for the free algebra is the Frobenius law of the monad. \square

There are many EM-algebras that are not FEM-algebras; a family of examples can be derived from [121, Theorem 6.4]. Here is a concrete example.

Example 6.3.4. The complex n -by- n -matrices form a Hilbert space A with inner product $\langle a, b \rangle = \frac{1}{n} \text{Tr}(a^\dagger \circ b)$. Matrix multiplication $m: A \otimes A \rightarrow A$ makes A into a dagger Frobenius monoid in \mathbf{FHilb} , and hence $T = - \otimes A$ into a dagger Frobenius monad on \mathbf{FHilb} . Define a monoid homomorphism $U: A \rightarrow A$ by conjugation $a \mapsto u^\dagger \circ a \circ u$ with a unitary matrix $u \in A$. Then $m \circ (\text{id}_A \otimes U)$ is an EM-algebra that is a FEM-algebra if and only if $u = u^\dagger$.

Proof. Because $U^\dagger = u \circ - \circ u^\dagger$, the Frobenius law (6.3.1) for T unfolds to the following.



This comes down to $U = (U^*)^\dagger$, that is, $u = u^\dagger$. \square

Before we calculate $\text{FEM}(- \otimes B)$ for a dagger Frobenius monoid B induced by an arbitrary groupoid, we work out an important special case.

Example 6.3.5. Let B be a dagger Frobenius monoid in \mathbf{FHilb} induced by a finite discrete groupoid \mathbf{G} as in Example 2.2.4. A FEM-algebra structure on a Hilbert space A for $- \otimes B$ consists of *quantum measurements* on A : orthogonal projections on A that sum to the identity.

For more information about quantum measurements, see [66, Section 3.2].

Proof. A FEM-algebra structure on A consists of a map $a: A \otimes B \rightarrow A$ subject to the FEM-laws. Since B has a basis indexed by objects of \mathbf{G} , it suffices to understand the maps $P_G: A \rightarrow A$ defined by $v \mapsto a(v \otimes \text{id}_G)$. The associative law implies that each P_G satisfies $P_G \circ P_G = P_G$, and from the Frobenius law we get that each P_G is self-adjoint, so that each P_G is an orthonormal projection. The unit law says that $\sum_G P_G = \text{id}_A$.

There is also another, graphical, way of seeing this. Quantum measurements can also be characterized as ‘ B -self-adjoint’ coalgebras for the comonad $- \otimes B$, where being B -self-adjoint means that the following equation holds [39].

(6.3.2)

Such coalgebras correspond precisely to FEM-algebras, as we will now show. Because of the dagger, coalgebras of the comonad $- \otimes B$ are just algebras of the monad $- \otimes B$. Thus it suffices to show that an algebra is FEM if and only if it satisfies (6.3.2). The implication from left to right is easy.

The other implication can be proven as follows.

This finishes the alternative proof. □

Example 6.3.6. Let a groupoid \mathbf{G} induce a dagger Frobenius monoid B in $\mathbf{C} = \mathbf{Rel}$ or $\mathbf{C} = \mathbf{FHilb}$ as in Example 2.2.4. There is an equivalence $\text{FEM}(- \otimes B) \simeq [\mathbf{G}, \mathbf{C}]$.

Proof. Separate the cases **Rel** and **FHilb**.

- In **Rel**, a FEM-algebra is a set A with a *relation* $g: A \rightarrow A$ for each $g \in B$ satisfying several equations. For each object G of \mathbf{G} , define $A_G = \{a \in A \mid \text{id}_G a = \{a\}\}$. The unit law implies that each $a \in A$ is in at least one A_G , and the other EM-law implies that no $a \in A$ can be in more than one A_G . Now if $g: G \rightarrow H$ in \mathbf{G} , then g defines a *function* $A_G \rightarrow A_H$ and maps everything outside of A_G to the empty set. Thus the FEM-algebra A defines an *action* of \mathbf{G} in **Rel**: a functor $F_A: \mathbf{G} \rightarrow \mathbf{Rel}$ making the sets $F_A(G)$ pairwise disjoint for distinct objects of \mathbf{G} .

Conversely, each such functor F defines a FEM-algebra A_F by setting $A_F = \bigcup_G F(G)$. But the category of such functors is equivalent to $[\mathbf{G}, \mathbf{Rel}]$.

- In **FHilb**, a FEM-algebra is a Hilbert space A with a morphism $g: A \rightarrow A$ for each $g \in B$. For each object G of \mathbf{G} , define $A_G = \{a \in A \mid \text{id}_G a = a\}$. As above,

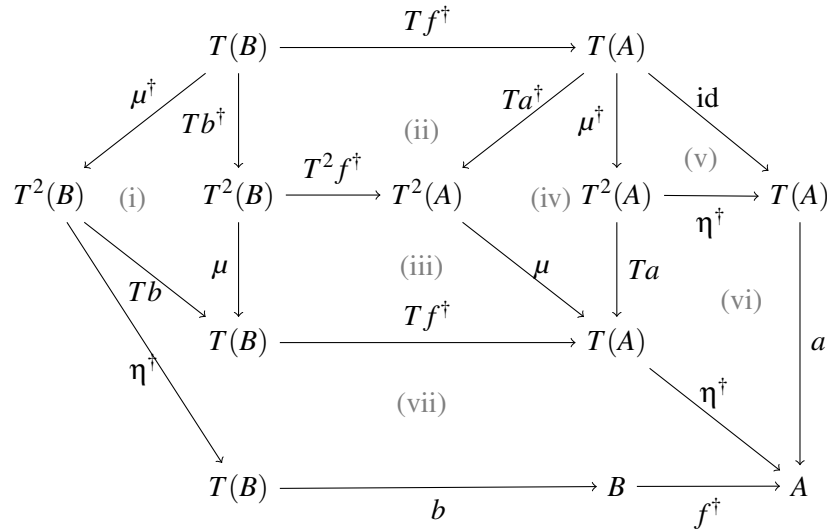
A is a direct sum of the A_G and $g: G \rightarrow H$ in \mathbf{G} defines a morphism $A_G \rightarrow A_H$ and annihilates A_G^\perp . This defines a representation of \mathbf{G} in \mathbf{FHilb} . The Frobenius law implies that this representation is unitary. □

In both of these examples, the same reasoning goes through over all of \mathbf{Hilb} . Moreover, the same constructions give rise to an equivalence between all EM-algebras for $- \otimes B$ and *all* (=also non-dagger) functors $\mathbf{G} \rightarrow (\mathbf{F})\mathbf{Hilb}$: hence any non-unitary presentation of a groupoid gives rise to a EM-algebra that is not FEM, providing us with more examples than those in Example 6.3.4.

The fact that all of the categories of FEM-algebras from the previous example had daggers is no accident.

Lemma 6.3.7. *Let T be a dagger Frobenius monad on a dagger category \mathbf{C} . The dagger on \mathbf{C} induces one on $\mathbf{FEM}(T)$.*

Proof. Let $f: (A, a) \rightarrow (B, b)$ be a morphism of FEM-algebras; we are to show that f^\dagger is a morphism $(B, b) \rightarrow (A, a)$. It suffices to show that $b \circ T(f) = f \circ a$ implies $a \circ T(f^\dagger) = f^\dagger \circ b$. Consider the following diagram:



Region (i) is the Frobenius law of (B, b) ; commutativity of (ii) follows from the assumption that f is a morphism $(A, a) \rightarrow (B, b)$ by applying T and the dagger; (iii) is naturality of μ ; (iv) is the Frobenius law of (A, a) ; (v) commutes since T is a comonad; (vi) and (vii) commute by naturality of η^\dagger . □

In fact, the equivalence of Example 6.3.6 is a dagger equivalence.

Lemma 6.3.8. *Let T be a dagger Frobenius monad. An EM-algebra (A, a) is FEM if and only if a^\dagger is a homomorphism $(A, a) \rightarrow (TA, \mu_A)$.*

Proof. If (A, a) is a FEM-algebra, its associativity means that a is a homomorphism $(TA, \mu_A) \rightarrow (A, a)$. Here $T(A)$ is a FEM-algebra too because T is a dagger Frobenius monad. Thus a^\dagger is a homomorphism $(A, a) \rightarrow (TA, \mu_A)$ by Lemma 6.3.7.

For the converse, assume a^\dagger is a homomorphism $(A, a) \rightarrow (TA, \mu_A)$, so the diagram

$$\begin{array}{ccc} T(A) & \xrightarrow{Ta^\dagger} & T^2(A) \\ a \downarrow & & \downarrow \mu \\ A & \xrightarrow{a^\dagger} & T(A) \end{array}$$

commutes. Hence $\mu \circ Ta^\dagger$ is self-adjoint as $a^\dagger \circ a$ is, giving the Frobenius law (6.3.1). \square

Interpreting the associative law for algebras as saying that $a: TA \rightarrow A$ is a homomorphism $(TA, \mu_A) \rightarrow (A, a)$, Lemmas 6.3.7 and 6.3.8 show that this morphism is universal, in the sense that if its dagger is an algebra homomorphism, then so is the dagger of any other algebra homomorphism to A (whose domain satisfies the Frobenius law).

Theorem 6.3.9. *Let F and G be dagger adjoints, and write $T = G \circ F$ for the induced dagger Frobenius monad. There are unique dagger functors K and J making the following diagram commute.*

$$\begin{array}{ccccc} \mathcal{Kl}(T) & \overset{K}{\dashrightarrow} & \mathbf{D} & \overset{J}{\dashrightarrow} & \text{FEM}(T) \\ & \swarrow & \updownarrow \begin{array}{c} F \\ G \end{array} & \searrow & \\ & & \mathbf{C} & & \end{array}$$

Moreover, J is full, K is full and faithful, and $J \circ K$ is the canonical inclusion.

Proof. It suffices to show that the comparisons $K: \mathcal{Kl}(T) \rightarrow \mathbf{D}$ and $J: \mathbf{D} \rightarrow \text{EM}(T)$ (see [111, VI.3, IV.5]) are dagger functors, and that J factors through $\text{FEM}(T)$.

Let (A, a) be in the image of J . As $J \circ K$ equals the canonical inclusion, $(T(A), \mu_A)$ is also in the image of J . Because J is full, the homomorphism $a: (TA, \mu_A) \rightarrow (A, a)$ is in the image of J , say $a = J(f)$. But then $G(f) = a$, and so $G(f^\dagger) = a^\dagger$. This implies that a^\dagger is a homomorphism $(A, a) \rightarrow (TA, \mu_A)$. Lemma 6.3.8 guarantees (A, a) is in $\text{FEM}(T)$.

Clearly J is a dagger functor. It remains to show that K is a dagger functor. As K is full and faithful, Lemma 2.1.16 gives $\mathcal{K}\ell(T)$ a unique dagger making it a dagger functor. This also makes $J \circ K$ a dagger functor, and since $J \circ K$ equals the canonical inclusion, the induced dagger on $\mathcal{K}\ell(T)$ must equal the canonical one from Lemma 6.3.1. \square

The previous theorem leads to a direct proof of Lemma 6.2.4 above, as follows. The definition of $\text{FEM}(T)$ makes sense for arbitrary monads (that might not satisfy the Frobenius law), and the proof above still shows that the image of $J: \mathbf{D} \rightarrow \text{EM}(T)$ lands in $\text{FEM}(T)$. Hence every free algebra is a FEM-algebra. This implies the Frobenius law (6.2.1) for T .

6.4 Formal monads on dagger categories

Both ordinary monads [137] and Frobenius monads [106] have been treated formally in 2-categories. This section establishes the counterpart for dagger 2-categories. Its main contribution is to show that the category of FEM-algebras satisfies a similar universal property for dagger Frobenius monads as EM-algebras do for ordinary monads. Recall from Definition 2.1.8 that a dagger 2-category is a category enriched in **DagCat**.

Definition 6.4.1. An adjunction in a dagger 2-category is just an adjunction in the underlying 2-category. A *dagger 2-adjunction* consists of two **DagCat**-enriched functors that form a 2-adjunction in the usual sense.

Adjunctions in a dagger 2-category need not specify left and right, just like the dagger adjunctions they generalize, but dagger 2-adjunctions need to specify left and right.

Definition 6.4.2. Let \mathbf{C} be a dagger 2-category. A *dagger Frobenius monad* consists of an object C , a morphism $T: C \rightarrow C$, and 2-cells $\eta: \text{id}_C \rightarrow T$ and $\mu: T^2 \rightarrow T$ that form a monad in the underlying 2-category of \mathbf{C} and satisfy the Frobenius law

$$\mu T \circ T \mu^\dagger = T \mu \circ \mu^\dagger T.$$

A *morphism of dagger Frobenius monads* $(C, S) \rightarrow (D, T)$ is a morphism $F: C \rightarrow D$

with a 2-cell $\sigma: TF \rightarrow FS$ making the following diagrams commute.

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \eta^T F & \nearrow & TF \\
 F & & \downarrow \sigma \\
 F\eta^S & \searrow & FS
 \end{array} &
 \begin{array}{ccc}
 T\sigma & \nearrow & TFS \\
 TTF & & \downarrow F\mu^S \\
 \mu^T F & \searrow & TF \xrightarrow{\sigma} FS
 \end{array} &
 \begin{array}{ccc}
 \sigma S & \nearrow & FSS \\
 TFS & & \downarrow \sigma^\dagger \\
 T\sigma^\dagger & \searrow & TTF \xrightarrow{\mu^T F} TF
 \end{array} &
 \begin{array}{ccc}
 & & FSS \xrightarrow{F\mu^S} FS \\
 & & \downarrow \sigma^\dagger \\
 & & TF
 \end{array}
 \end{array} \quad (6.4.1)$$

A transformation between morphisms of dagger Frobenius monads $(F, \sigma) \rightarrow (G, \tau)$ is a 2-cell $\phi: F \rightarrow G$ making the following diagrams commute.

$$\begin{array}{ccc}
 TF & \xrightarrow{T\phi} & TG \\
 \sigma \downarrow & & \downarrow \tau \\
 FS & \xrightarrow{\phi S} & GS
 \end{array}
 \qquad
 \begin{array}{ccc}
 TG & \xrightarrow{T\phi^\dagger} & TF \\
 \sigma \downarrow & & \downarrow \tau \\
 GS & \xrightarrow{\phi^\dagger S} & FS
 \end{array}$$

Define the composition of morphisms to be $(G, \tau) \circ (F, \sigma) = (GF, G\sigma \circ \tau F)$. Horizontal and vertical composition of 2-cells in \mathbf{C} define horizontal and vertical composition of transformations of morphisms of dagger Frobenius monads, and the dagger on 2-cells of \mathbf{C} gives a dagger on these transformations. This forms a dagger 2-category $\mathbf{DFMonad}(\mathbf{C})$.

Omitting the third diagram of (6.4.1) gives the usual definition of a monad morphism. We require this coherence with the dagger for the following reason: just as the first two diagrams of (6.4.1) ensure that $(C, T) \mapsto \mathbf{EM}(T)$ is a 2-functor $\mathbf{Monad}(\mathbf{Cat}) \rightarrow \mathbf{Cat}$, the third one ensures that $(C, T) \mapsto \mathbf{FEM}(T)$ is a dagger 2-functor $\mathbf{DFMonad}(\mathbf{DagCat}) \rightarrow \mathbf{DagCat}$.

There is an inclusion dagger 2-functor $\mathbf{C} \rightarrow \mathbf{DFMonad}(\mathbf{C})$ given by $C \mapsto (C, \text{id}_C)$, $F \mapsto (F, \text{id})$, and $\psi \mapsto \psi$. There is also a forgetful dagger 2-functor $\mathbf{DFMonad}(\mathbf{C}) \rightarrow \mathbf{C}$ given by $(C, T) \mapsto C$, $(F, \sigma) \mapsto F$, and $\psi \mapsto \psi$. As with the formal theory of monads [137], the forgetful functor is the left adjoint to the inclusion functor. To see this, it suffices to exhibit a natural isomorphism of dagger categories $[C, D] \cong [(C, T), (D, \text{id}_D)]$, such as sending $F: C \rightarrow D$ to $(F, F\eta^T)$ and ψ to ψ .

Definition 6.4.3. A dagger 2-category \mathbf{C} admits the construction of FEM-algebras when the inclusion $\mathbf{C} \rightarrow \mathbf{DFMonad}(\mathbf{C})$ has a right adjoint $\mathbf{FEM}: \mathbf{DFMonad}(\mathbf{C}) \rightarrow \mathbf{C}$.

We will abbreviate $\mathbf{FEM}(C, T)$ to $\mathbf{FEM}(T)$ when no confusion can arise.

Theorem 6.4.4. *If \mathbf{C} admits the construction of FEM-algebras, then dagger Frobenius monads factor as dagger adjunctions.*

Proof. We closely follow the proof of the similar theorem for ordinary monads in [137], but need to verify commutativity of some additional diagrams. To verify that (T, μ) is a morphism of dagger Frobenius monads $(C, \text{id}) \rightarrow (C, T)$, the first two diagrams are exactly as for ordinary monads, and the third diagram commutes by Lemma 6.2.3. Denoting the counit of the adjunction of the assumption by $(E, \varepsilon): (\text{FEM}(T), \text{id}) \rightarrow (C, T)$, the universal property gives a unique morphism $(J, \text{id}): (C, \text{id}) \rightarrow (\text{FEM}(T), \text{id})$ making the following diagram commute.

$$\begin{array}{ccc} (C, \text{id}) & \xrightarrow{(T, \mu)} & (C, T) \\ & \searrow (J, \text{id}) & \nearrow (E, \varepsilon) \\ & & (\text{FEM}(T), \text{id}) \end{array}$$

Thus $T = EJ$ and $\mu = \varepsilon J$. Next we verify ε is a transformation of morphisms of dagger Frobenius monads $(EJE, \mu E) \rightarrow (E, \varepsilon)$, by showing the following diagrams commute.

$$\begin{array}{ccc} TEJE & \xrightarrow{T\varepsilon} & TE \\ \mu E \downarrow & & \downarrow \varepsilon \\ TE & \xrightarrow{\varepsilon} & E \end{array} \quad \begin{array}{ccc} TE & \xrightarrow{T\varepsilon^\dagger} & TEJE \\ \varepsilon \downarrow & & \downarrow \mu E \\ E & \xrightarrow{\varepsilon^\dagger} & TE \end{array}$$

These diagrams are instances of the second and third equations of (6.4.1).

The rest of the proof proceeds exactly as in [137]. As ε is a transformation, the adjunction gives a unique 2-cell $\xi: JE \rightarrow \text{id}_{\text{FEM}(T)}$ in \mathbf{C} with $E\xi = \varepsilon$. Now

$$E(\xi J \circ J\eta) = E\xi J \circ EJ\eta = \varepsilon J \circ T\eta = \mu \circ T\eta = \text{id},$$

and so $\xi J \circ J\eta = \text{id}$ by the universal property of the counit. Furthermore

$$E\xi \circ \eta E = \varepsilon \circ \eta E = \text{id},$$

so (E, J, μ, ξ) is an adjunction generating (C, T) . □

Theorem 6.4.5. *The dagger 2-category DagCat admits the construction of FEM-algebras.*

Proof. We first show $(\mathbf{C}, T) \mapsto (\text{FEM}(T))$ extends to a well-defined dagger 2-functor $\text{FEM}: \text{DFMonad}(\mathbf{DagCat}) \rightarrow \mathbf{DagCat}$. On a 1-cell $(F, \sigma): (\mathbf{C}, S) \rightarrow (\mathbf{D}, T)$, define a dagger functor $\text{FEM}(F, \sigma): \text{FEM}(S) \rightarrow \text{FEM}(T)$ as follows: map the algebra $a: SA \rightarrow A$ to $F(a) \circ \sigma_A: TFA \rightarrow FSA \rightarrow FA$ and the homomorphism $f: (A, a) \rightarrow (B, b)$ to $F(f)$. The laws for FEM-algebras of (F, σ) now show that $\text{FEM}(F, \sigma)$ sends FEM-algebras to FEM-algebras. On a 2-cell $\psi: (F, \sigma) \rightarrow (G, \tau)$, define $\text{FEM}(\psi)$ by $\text{FEM}(\psi)_A = \psi_A$; each ψ_A is a morphism of FEM-algebras.

There is a natural isomorphism of dagger categories $[\mathbf{C}, \text{FEM}(T)] \cong [(\mathbf{C}, \text{id}_{\mathbf{C}}), (\mathbf{D}, T)]$ as follows. A dagger functor $\mathbf{C} \rightarrow \text{FEM}(T)$ consists of a dagger functor $F\mathbf{C} \rightarrow \mathbf{D}$ and a family of maps $\sigma_A: TFA \rightarrow FA$ making each (FA, σ_A) into a FEM-algebra. Then $(F, \sigma): (\mathbf{C}, \text{id}_{\mathbf{C}}) \rightarrow (\mathbf{D}, T)$ is a well-defined morphism of dagger Frobenius monads. Similarly, any $(F, \sigma): (\mathbf{C}, \text{id}_{\mathbf{C}}) \rightarrow (\mathbf{D}, T)$ defines a dagger functor $\mathbf{C} \rightarrow \text{FEM}(T)$. On the level of natural transformations both of these operations are obvious. \square

In [137] this last result is proved as follows: instead of starting with the definition of \mathbf{C}^T , the construction is recovered from the assumption that the right adjoint exists and considering functors from the categories $\mathbf{1}$ and $\mathbf{2}$ to the right adjoint, thus recovering the objects and arrows of \mathbf{C}^T . It is unclear how to write a similar proof in our case: while $[\mathbf{2}, \mathbf{C}]$ classifies arrows and commutative squares for an ordinary category, it is not obvious how to replace $\mathbf{2}$ with a dagger category playing an analogous role.

6.5 Strength

As we saw in Example 6.2.2, (Frobenius) monoids in a monoidal dagger category induce (dagger Frobenius) monads on the category. This in fact sets up an adjunction between monoids and strong monads [151]. This section shows that the Frobenius law promotes this adjunction to an equivalence. Most of this section generalizes to the non-dagger setting.

A pedestrian approach to the adjunction between monoids in \mathbf{C} and strong monads on \mathbf{C} proceeds by defining the relevant functors and verifying that they work and form an adjunction directly. Our official approach to the equivalence between dagger Frobenius monoids and strong dagger Frobenius monads proceeds similarly. However, there is a higher-level approach to this that we'll explain briefly afterwards. Making the higher-level picture as detailed as the rest of the thesis would use more 2-categories than needed elsewhere, which is why it is not our official approach.

The abstract viewpoint will explain why – in the Frobenius case – one would expect to get an equivalence of categories rather than an adjunction. This will then justify our definition of a strong Frobenius monoid, which assumes that the strength natural transformation $st_{A,B}: A \otimes F(B) \rightarrow F(A \otimes B)$ is unitary, which then directly implies that $F(-) \cong F(- \otimes I) \cong - \otimes F(I)$.

Definition 6.5.1. A dagger functor F between monoidal dagger categories is *strong* if it is equipped with natural unitary morphisms $st_{A,B}: A \otimes F(B) \rightarrow F(A \otimes B)$ satisfying $st \circ \alpha = F(\alpha) \circ st \circ (\text{id} \otimes st)$ and $F(\lambda) \circ st = \lambda$. A (dagger Frobenius) monad on a monoidal dagger category is *strong* if it is a strong dagger functor with $st \circ (\text{id} \otimes \mu) = \mu \circ T(st) \circ st$ and $st \circ (\text{id} \otimes \eta) = \eta$. A morphism β of dagger Frobenius monads is *strong* if $\beta \circ st = st \circ (\text{id} \otimes \beta)$.

To prove the equivalence between dagger Frobenius monoids and strong dagger Frobenius monads, we need two lemmas.

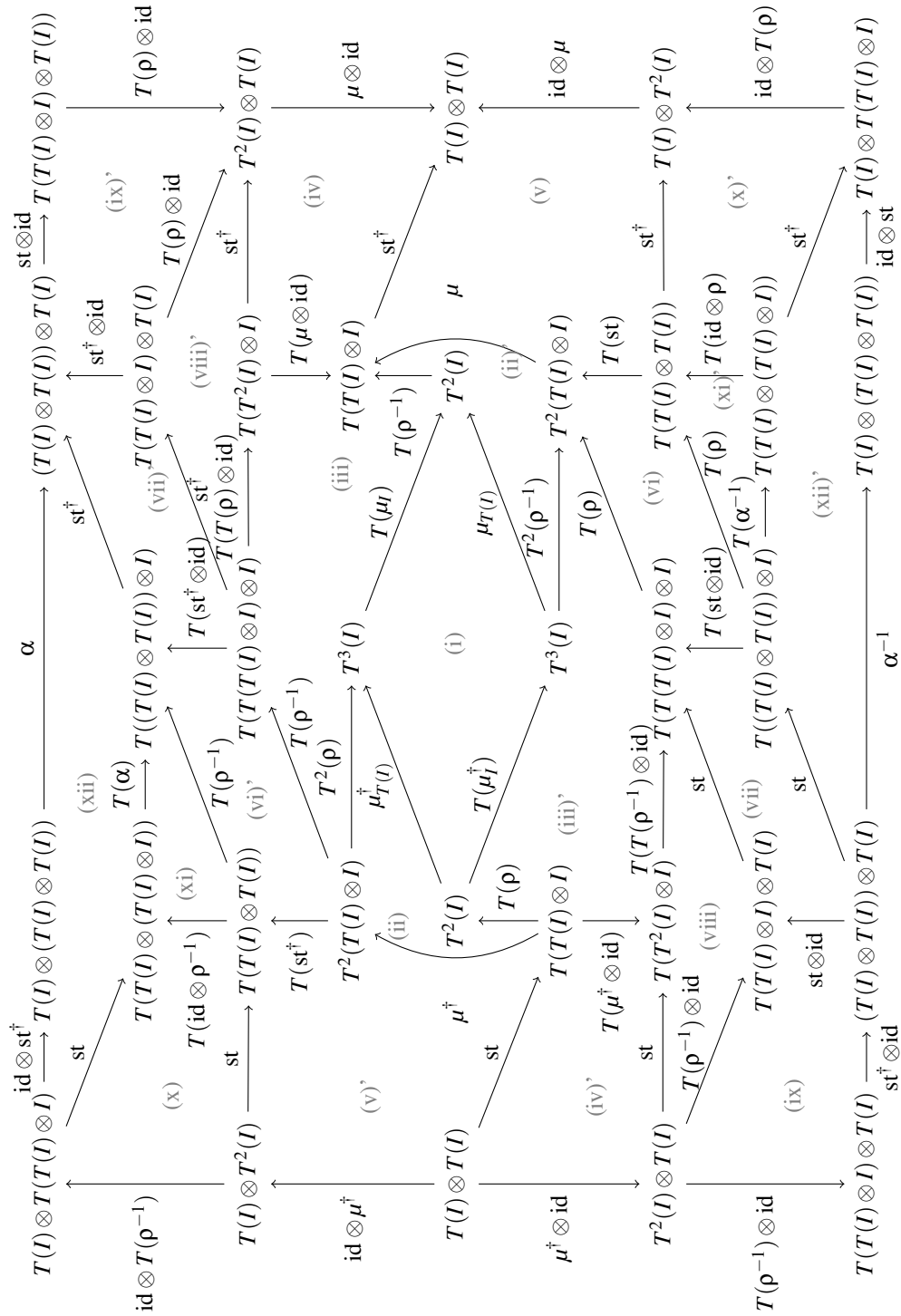
Lemma 6.5.2. *If T is a strong dagger Frobenius monad on a monoidal dagger category, then $T(I)$ is a dagger Frobenius monoid.*

Proof. Consider the diagram in Figure 6.1. Region (i) commutes because T is a dagger Frobenius monad, (ii) because μ^\dagger is natural, (iii) because ρ^{-1} is natural, (iv) because st^\dagger is natural, (v) is a consequence of T being a strong monad, (vi) commutes as ρ is natural, (vii) and (viii) because st is natural, (ix) commutes trivially and (x) because st is natural. Regions (ii)'-(x)' commute for dual reasons. Hence the outer diagram commutes. \square

Lemma 6.5.3. *If T is a strong dagger Frobenius monad on a monoidal dagger category, then $T\rho \circ st: A \otimes T(I) \rightarrow T(A)$ preserves η^\dagger and μ^\dagger .*

Proof. To show that η^\dagger is preserved, it suffices to see that

$$\begin{array}{ccc}
 A \otimes T(I) & & \\
 st_{A,I} \downarrow & \searrow \text{id} \otimes \eta_I^\dagger & \\
 T(A \otimes I) & \xrightarrow{\eta_{A \otimes I}^\dagger} & A \otimes I \\
 T(\rho_A) \downarrow & & \downarrow \rho_A \\
 T(A) & \xrightarrow{\eta_A^\dagger} & A
 \end{array}$$

Figure 6.1: Diagram proving that $T \mapsto T(I)$ preserves the Frobenius law.

commutes. But the rectangle commutes because η^\dagger is natural, and the triangle commutes because T is a strong monad and strength is unitary.

To see that μ^\dagger is preserved, consider the following diagram:

$$\begin{array}{ccccccc}
A \otimes T(I) & \xrightarrow{\text{id} \otimes \mu^\dagger} & A \otimes T^2(I) & \xrightarrow{\text{id} \otimes T(\rho^{-1})} & A \otimes T(T(I) \otimes I) & \xrightarrow{\text{id} \otimes \text{st}^\dagger} & A \otimes (T(I) \otimes T(I)) \\
\downarrow \text{st} & & \downarrow \text{id} & & \swarrow \text{id} \otimes \text{st} & & \downarrow \alpha \\
& & A \otimes T^2(I) & \xleftarrow{\text{id} \otimes T(\rho)} & A \otimes T(T(I) \otimes I) & & (A \otimes T(I)) \otimes T(I) \\
& & \downarrow \text{id} & & \downarrow \text{st} & & \downarrow \text{st} \\
& & & & T(A \otimes (T(I) \otimes I)) & \xrightarrow{T(\alpha)} & T((A \otimes T(I)) \otimes I) \\
& & \downarrow \text{id} & & \downarrow \text{st} & & \downarrow T(\rho) \\
& & & & & \searrow T(\text{id} \otimes \rho) & T(A \otimes T(I)) \\
& & & & & & \downarrow T(\text{st}) \\
T(A \otimes I) & \xrightarrow{\hspace{10em}} & T^2(A \otimes I) & & & & \\
\downarrow T(\rho) & & \downarrow T^2(\rho) & & & & \\
T(A) & \xrightarrow{\hspace{10em}} & T^2(A) & & & & \\
& & \mu^\dagger & & & &
\end{array}$$

(i) (ii) (iii) (iv) (v) (vi)

Commutativity of region (i) is a consequence of strength being unitary, (ii) commutes by definition, (iii) commutes as strength is natural, (iv) because T is a strong functor, (v) by coherence and finally (vi) by naturality of μ^\dagger . \square

Theorem 6.5.4. *Let \mathbf{C} be a monoidal dagger category. The operations $B \mapsto - \otimes B$ and $T \mapsto T(I)$ form an equivalence between dagger Frobenius monoids in \mathbf{C} and strong dagger Frobenius monads on \mathbf{C} .*

Proof. It is well-known that $B \mapsto - \otimes B$ is left adjoint to $T \mapsto T(I)$, when considered as maps between ordinary monoids and ordinary strong monads, see [151]; the unit of the adjunction is $\lambda: I \otimes B \rightarrow B$, and the counit is determined by $T\rho \circ \text{st}: A \otimes T(I) \rightarrow T(A)$.

Example 6.2.2 already showed that $B \mapsto - \otimes B$ preserves the Frobenius law. Moreover, Lemma 6.5.2 shows that $T \mapsto T(I)$ preserves the Frobenius law as well. It remains to prove that they form an equivalence. Clearly the unit of the adjunction is a natural isomorphism. To see that the counit is also a natural isomorphism, combine Lemmas 6.5.3 and 2.2.5. \square

It follows from the previous theorem that not every dagger Frobenius monad is strong: as discussed in Section 6.2, the monad induced by the dagger adjunction of Example 3.2.3 is not of the form $- \otimes B$ for fixed B .

Let us now sketch the higher-level viewpoint promised at the start of this section in order to justify our assumption that st is unitary. If A is an object of a monoidal category \mathbf{C} , then $- \otimes A$ is an endofunctor on \mathbf{C} . Conversely, any endofunctor F on \mathbf{C} gives an object $F(I)$ of \mathbf{C} . The category of endofunctors is monoidal (with composition of functors as \otimes) and the functor $A \mapsto - \otimes A$ is strong monoidal. However, for arbitrary endofunctors, there are no obvious maps between $F(I) \otimes G(I)$ and $FG(I)$, so the converse isn't in general monoidal unless one imposes further conditions on the endofunctors. Similarly, while $A \mapsto - \otimes A \mapsto I \otimes A$ results in a functor isomorphic to $\text{id}_{\mathbf{C}}$, there is no obvious natural transformation between $- \otimes F(I)$ and F . A reasonable way of getting a natural transformation $- \otimes F(I) \rightarrow F$ is by requiring F to be strong: i.e. asking for a natural transformation $\text{st}_{A,B}: A \otimes F(B) \rightarrow F(A \otimes B)$ subject to some further conditions. Hence, if one restricts to strong functors and transformations of strong functors, the map $F \mapsto F(I)$ becomes (lax) monoidal and in fact it is the right adjoint to $A \mapsto - \otimes A$.

Once this has been verified, it is not hard to obtain the adjunction between monoids in \mathbf{C} and strong monads on \mathbf{C} : monoidal categories, lax monoidal functors and monoidal transformations organize themselves into a 2-category. If $\mathbf{1}$ is the terminal monoidal category, then $\text{hom}(\mathbf{1}, -)$ is a 2-functor into the 2-category of categories. Any 2-functor preserves adjunctions inside a 2-category, so applying $\text{hom}(\mathbf{1}, -)$ to the adjunction between \mathbf{C} and strong endofunctors on \mathbf{C} results in an adjunction. Moreover, a lax monoidal functor $\mathbf{1} \rightarrow \mathbf{C}$ is the same thing as a monoid in \mathbf{C} (and a monoid homomorphism is the same thing as a monoidal transformation between the functors). Hence the resulting adjunction is the adjunction between monoids in \mathbf{C} and monoids in the category of strong endofunctors on \mathbf{C} , i.e. the category of strong monads on \mathbf{C} .

Dualizing this, one obtains an adjunction between comonoids in \mathbf{C} and comonads on \mathbf{C} , the functor $A \mapsto - \otimes A$ now being the right adjoint. Since a Frobenius algebra is both a monoid and a comonoid, one might then expect a similar story, resulting in an ambijunction. For such a story to work, one first needs to find the correct category of monoidal categories, in which a functor $\mathbf{1} \rightarrow \mathbf{C}$ corresponds to a Frobenius algebra in \mathbf{C} . This is given by the notion of a Frobenius monoidal functor [114, 43] – a functor that is both lax monoidal and oplax monoidal and satisfies a coherence law. Then to get an ambijunction between Frobenius algebras and (strong) Frobenius monads, one only needs to find an ambijunction between \mathbf{C} and suitable endofunctors on it, so that both adjoints are Frobenius monoidal. Since $A \mapsto - \otimes A$ is strong monoidal there is nothing to worry about. For $F \mapsto F(I)$ to be Frobenius monoidal,

one needs to consider functors that are both strong, costrong and satisfy a coherence law. A morphism of such functors will then at the very least be a morphism of the underlying strong and costrong functors. But this is enough to ensure that applying $\text{hom}(1, -)$ to the adjunction results in an equivalence: the counit $- \otimes F(I) \rightarrow F(-)$ has to be a morphism of strong and costrong functors, and this is enough to ensure that Lemma 6.5.3 goes through. Hence the counit is a strict morphism of Frobenius structures, and thus an isomorphism by Lemma 2.2.5. Since the counit is built from the monoidal unitor and strength, the strength has to be an isomorphism as well. The meaning of this is that, if an endofunctor F is strong, costrong and satisfies the required coherence law, and moreover admits a Frobenius monad structure in which the (co)multiplication and (co)unit are morphisms of strong and costrong functors, then the strength and costrength transformations are necessarily inverses to each other. Hence, if one just wants to obtain the resulting equivalence and doesn't care too much about the adjunction between \mathbf{C} and coherently strong and costrong endofunctors on it, one might as well start by restricting to those functors for which the strength natural transformation is an isomorphism. Since we are working in the dagger setting, it is natural to assume that the strength map is in fact unitary.

A Frobenius monoid in a monoidal dagger category is *special* when $\mu \circ \eta' = \eta \circ \mu'$. Theorem 6.5.4 restricts to an equivalence between special dagger Frobenius monoids and special strong dagger Frobenius monads.

For symmetric monoidal dagger categories, there is also a notion of commutativity for strong monads [102, 85]. Given a strong dagger Frobenius monad T , one can define a natural transformation $st'_{A,B}: T(A) \otimes B \rightarrow T(A \otimes B)$ by $T(\sigma_{B,A}) \circ st_{B,A} \circ \sigma_{T(A),B}$, and

$$\begin{aligned} dst_{A,B} &= \mu_{A \otimes B} \circ T(st'_{A,B}) \circ st_{T(A),B}, \\ dst'_{A,B} &= \mu_{A \otimes B} \circ T(st_{A,B}) \circ st'_{A,T(B)}. \end{aligned}$$

The strong dagger Frobenius monad is *commutative* when these coincide. Theorem 6.5.4 restricts to an equivalence between commutative dagger Frobenius monoids and commutative strong dagger Frobenius monads. Kleisli categories of commutative monads on symmetric monoidal categories are again symmetric monoidal [41]. This extends to dagger categories.

Theorem 6.5.5. *If T is a commutative strong dagger Frobenius monad on a symmetric monoidal dagger category \mathbf{C} , then $\mathcal{Kl}(T)$ is a symmetric monoidal dagger category.*

Proof. The monoidal structure on $\mathcal{Kl}(T)$ is given by $A \otimes_T B = A \otimes B$ on objects and by $f \otimes_T g = \text{dst} \circ (f \otimes g)$ on morphisms. The coherence isomorphisms of $\mathcal{Kl}(T)$ are

images of those in \mathbf{C} under the functor $\mathbf{C} \rightarrow \mathcal{K}\ell(T)$. This functor preserves daggers and hence unitary isomorphisms, making all coherence isomorphisms of $\mathcal{K}\ell(T)$ unitary. It remains to check that the dagger on $\mathcal{K}\ell(T)$ satisfies $(f \otimes_T g)^\dagger = f^\dagger \otimes_S g^\dagger$. Theorem 6.5.4 makes T isomorphic to $S = - \otimes T(I)$, and this induces an isomorphism between the respective Kleisli categories that preserves daggers and monoidal structure on the nose. Thus it suffices to check that this equation holds on $\mathcal{K}\ell(S)$:

$$(f \otimes_S g)^\dagger = \begin{array}{c} \text{---} \\ | \\ \boxed{f} \\ | \\ \text{---} \end{array} \otimes_S \begin{array}{c} \text{---} \\ | \\ \boxed{g} \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \boxed{f} \\ | \\ \text{---} \end{array} \otimes_S \begin{array}{c} \text{---} \\ | \\ \boxed{g} \\ | \\ \text{---} \end{array} = f^\dagger \otimes_S g^\dagger.$$

But this is a straightforward graphical argument

using associativity, commutativity, the unit law, and Lemma 6.2.3. □

6.6 Closure

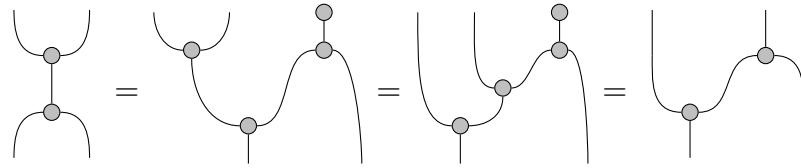
This final section justifies the Frobenius law from first principles, by explaining it as a coherence property between daggers and closure. In a monoidal dagger category that is closed, monoids and daggers interact in two ways. First, any monoid picks up an involution by internalizing the dagger. Second, any monoid embeds into an endohomset by closure, and the dagger is an involution on the endohomset. The Frobenius law is equivalent to the property that these two canonical involutions coincide.

We start by giving an equivalent formulation of the Frobenius law.

Lemma 6.6.1. *A monoid (A, \cdot, \circ) in a monoidal dagger category is a dagger Frobenius monoid if and only if it satisfies the following equation.*

$$\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \quad (6.6.1)$$

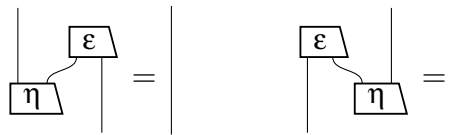
Proof. The Frobenius law (2.2.1) directly implies (6.6.1). Conversely, (6.6.1) gives



by associativity. But since the left-hand side is self-adjoint, so is the right-hand side, giving the Frobenius law (2.2.1). \square

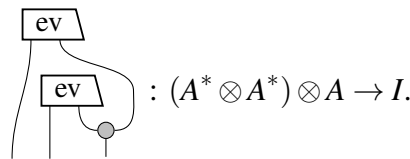
Any dagger Frobenius monoid forms a duality with itself in the following sense.

Definition 6.6.2. Morphisms $\eta: I \rightarrow A \otimes B$ and $\varepsilon: B \otimes A \rightarrow I$ in a monoidal category form a duality when they satisfy the following equations.



In the categories **Rel** and **FHilb**, every object A is part of a duality $(A, B, \eta, \varepsilon)$: they are compact categories. Moreover, in those categories we may choose $\varepsilon = \eta^\dagger \circ \sigma$: they are compact dagger categories.

Let \mathbf{C} be a monoidal dagger category that is also a closed monoidal category, so that there is a correspondence of morphisms $A \otimes B \rightarrow C$ and $A \rightarrow [B, C]$ called currying. Write A^* for $[A, I]$, and write ev for the counit $A^* \otimes A \rightarrow I$. If $(A, \cdot, \circ, \delta)$ is a monoid in \mathbf{C} , then (A, \cdot, \circ) is a comonoid in \mathbf{C} and A^* becomes a monoid with unit and multiplication given by currying $\circ: I \otimes A \rightarrow I$ and



A and A^* are related by a map $i: A \rightarrow A^*$ obtained by currying δ .

Proposition 6.6.3. A monoid $(A, \cdot, \circ, \delta)$ in a monoidal dagger category that is also a closed monoidal category is a dagger Frobenius monoid if and only if $i: A \rightarrow A^*$ is a monoid homomorphism and $\text{ev}: A^* \otimes A \rightarrow I$ forms a duality with

(6.6.2)

Proof. The morphism i always preserves units: $ev \circ (i \otimes id) \circ (\circlearrowleft \otimes id) = \circlearrowleft \circ \circlearrowright \circ (\circlearrowleft \otimes id) = \circlearrowleft$. It preserves multiplication precisely when:

(6.6.3)

Furthermore, $ev: A^* \otimes A \rightarrow I$ and (6.6.2) form a duality precisely when:

(6.6.4)

By evaluating both sides, it is easy to see that the left equation implies the right one.

Now, assuming the Frobenius law (2.2.1), Lemma 6.2.3 and the unit law guarantee that (the left equation of) (6.6.4) is satisfied, as well as (6.6.3):

Conversely, equations (6.6.3) and (6.6.4) imply:

Lemma 6.6.1 now finishes the proof. □

In any closed monoidal category, $[A, A]$ is canonically a monoid. The ‘way of the dagger’ suggests that there should be interaction between the dagger and closure in categories that have both.

Definition 6.6.4. A *sheathed dagger category* is a monoidal dagger category that is also closed monoidal, such that

$ev_{[A,A]} : [A, A] \otimes A \rightarrow A$

for all objects A , and for all morphisms $f, g: B \rightarrow C \otimes [A, A]$:

Any compact dagger category is a sheathed dagger category: the first axiom there says that $A^* \otimes A$ with its canonical monoid structure is a dagger Frobenius monoid, and the second axiom then holds because the evaluation morphism is invertible. In principle, the definition of sheathed dagger categories is much weaker. Although we have no uncontrived examples of sheathed dagger categories that are not compact dagger categories, we will work with the more general sheathed dagger categories because they are the natural home for the following arguments. The second axiom merely says that partial evaluation is faithful, which is the case in any well-pointed monoidal dagger category. In any closed monoidal category, the evaluation map canonically makes A into an algebra for the monad $- \otimes [A, A]$. The first axiom merely says that A is $- \otimes [A, A]$ -self-adjoint, as in (6.3.2). It does not assume $[A, A]$ is a dagger Frobenius monoid, nor that A is a FEM-algebra. No other plausible conditions are imposed, such as the bifunctor $[-, -]$ being a dagger functor, which does hold in compact dagger categories. Nevertheless, the following example shows that being a sheathed dagger category is an essentially monoidal notion that degenerates for cartesian categories.

Example 6.6.5. If a Cartesian closed category has a dagger, every homset is a singleton.

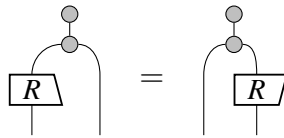
Proof. Because the terminal object is in fact a zero object, there are natural bijections $\text{hom}(A, B) \cong \text{hom}(0 \times A, B) \cong \text{hom}(0, B^A) \cong \{*\}$. \square

Currying the multiplication of a monoid (A, μ, η) in a closed monoidal category gives a monoid homomorphism $R: A \rightarrow [A, A]$. This is the abstract version of Cayley's embedding theorem, which states that any group embeds into the symmetric group on itself. If the category also has a dagger, there is also a monoid homomorphism $R_* = [R^\dagger, \text{id}_I]: A^* \rightarrow [A, A]^*$.

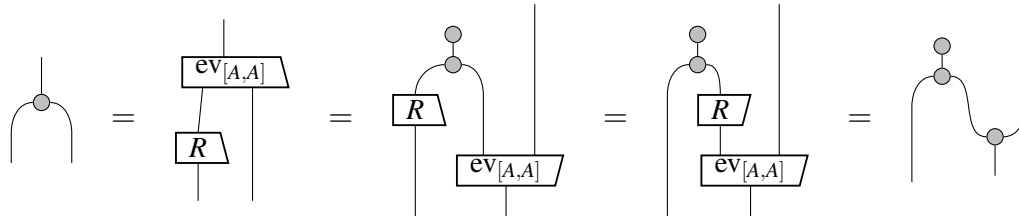
Theorem 6.6.6. In a sheathed dagger category, (A, μ, η) is a dagger Frobenius monoid if and only if the following diagram commutes.

$$\begin{array}{ccc}
 A & \xrightarrow{R} & [A, A] \\
 i_A \downarrow & & \downarrow i_{[A, A]} \\
 A^* & \xrightarrow{R_*} & [A, A]^*
 \end{array}$$

Proof. Evaluating both sides shows that R and i commute precisely when:



But this is equivalent to



Lemma 6.6.1 now finishes the proof. □

Corollary 6.6.7. *The following are equivalent for a monoid (A, \circ, \bullet) in a compact dagger category:*

- (A, \circ, \bullet) is a dagger Frobenius monoid;
- the canonical morphism $i: A \rightarrow A^*$ is an involution: $i_* \circ i = \text{id}_A$;
- the canonical Cayley embedding is involutive: $i \circ R = R_* \circ i$.

Proof. Combine Proposition 6.6.3 and Theorem 6.6.6. □

6.7 Limits of algebras

In this section we show that the forgetful functor $\text{FEM}(T) \rightarrow \mathbf{C}$ creates dagger limits. This is analogous to the non-dagger case. However, this then will imply that the functor also creates dagger colimits, whereas in the non-dagger case only those colimits preserved by T and T^2 are created by the forgetful functor. In a sense, this is not very striking since a dagger monad T is automatically both continuous and cocontinuous due to it factoring as an ambidextrous adjunction. However, it is still pleasing to see that the notion of dagger limit cooperates well with dagger monads, providing evidence that both notions are on the right track.

Theorem 6.7.1. *Let $D: \mathbf{J} \rightarrow \text{FEM}(T)$ be a diagram and $\Omega \subset \mathbf{J}$ be weakly initial. Write $U: \text{FEM}(T) \rightarrow \mathbf{C}$ for the forgetful functor. If (UD, Ω) has a dagger limit, so does (D, Ω) .*

Proof. For each $A \in \mathbf{J}$ write $D(A)$ as $(D(A), a_A)$, and let (L, l_A) be the dagger limit of (UD, Ω) . The maps $TL \xrightarrow{Tl_A} TD(A) \xrightarrow{a_A} D(A)$ make TL into a cone for UA ; let $l: TL \rightarrow L$ be the unique arrow factorizing this cone. We will prove that (L, l) is the desired dagger limit. Note that the normalization and independence properties of L imply those of (L, l) . Hence we only need to prove that (L, l) is a limiting FEM-algebra.

That (L, l) is a limiting EM-algebra is proved as in ordinary category theory, see e.g. [25, 4.3] We reproduce the proof for the sake of completeness: For the unit law, note that the square on the left of the diagram

$$\begin{array}{ccccc}
 L & \xrightarrow{\eta} & T(L) & \xrightarrow{l} & L \\
 l_A \downarrow & & \downarrow Tl_A & & \downarrow l_A \\
 D(A) & \xrightarrow{\eta} & TD(A) & \xrightarrow{a_A} & D(A) \\
 & \searrow & & \nearrow & \\
 & & \text{id} & &
 \end{array}$$

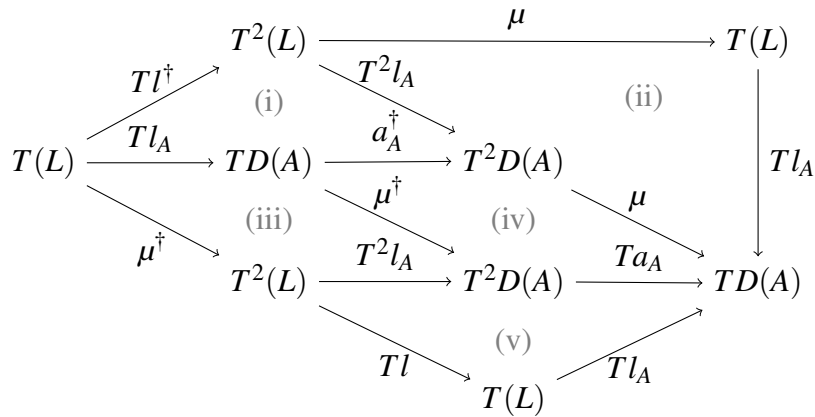
commutes by naturality of η and the square on the right by definition of l . The bottom part of the diagram is just the unit law of (DA, a_A) . Since A was arbitrary, the unit axiom is satisfied by (L, l) . The associative law is proved similarly.

To prove the Frobenius law, note first that the monad T factors as the composite of two dagger adjoints and hence is continuous, so that (TL, Tl_A) is the dagger limit of TUD with support Ω . Next we show that the diagram

$$\begin{array}{ccccc}
 TD(A) & \xrightarrow{Tl_A^\dagger} & T(L) & \xrightarrow{l} & L \\
 a_A \downarrow & & \downarrow Tl_B & & \downarrow l_B \\
 D(A) & \xrightarrow{l_A^\dagger} & T(A) & \xrightarrow{l_B} & D(B) \\
 & & & \nearrow a_B &
 \end{array}$$

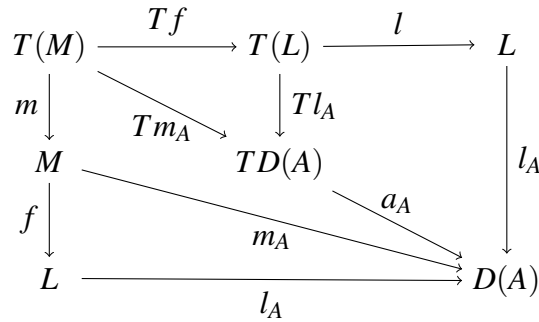
commutes for any $A, B \in \Omega$. The right half commutes by definition of l . For the left half, note that the algebra maps induce a natural transformation $TUD \rightarrow UD$. Since each of these algebras is FEM, this transformation is adjointable. Hence we can apply Lemma 4.3.7 to conclude that the left half commutes. Hence the whole diagram commutes. As it does so for each B , we conclude that l_A^\dagger is an algebra homomorphism

$(D(A), a_A) \rightarrow (L, l)$ for each A . Hence region (i) in the diagram



commutes. Regions (ii) and (iii) commute by naturality of μ , and region (iv) commutes by the Frobenius law of (DA, a_A) . Finally, region (v) commutes by definition of l . Hence the whole diagram commutes. Since (TL, Tl_A) is a limit of TUD , this implies the Frobenius law of (L, l) .

Since $FEM(T) \rightarrow \mathbf{C}$ is faithful, (L, l) is a cone for D . We conclude by proving that it is a limiting cone. If $m_A : (M, m) \rightarrow (D(A), a_A)$ then $m_A : M \rightarrow D(A)$ is a cone for UD and hence factors through L via an unique $f : M \rightarrow L$. It suffices to prove that f is an algebra homomorphism $(M, m) \rightarrow (L, l)$. This follows once we show that the diagram



commutes for each A . The triangles on the top and bottom commute by definition of f and the triangle in the middle commutes because each m_A is an algebra homomorphism. The trapezoid on the right commutes by definition of l , concluding the proof. \square

6.8 Monadicity

Next we will prove a version of Beck's monadicity theorem [19] for dagger monads.

Definition 6.8.1. A dagger functor $G: \mathbf{D} \rightarrow \mathbf{C}$ is *dagger monadic* if there is a dagger Frobenius monad (T, μ, η) on \mathbf{C} and a dagger equivalence $\mathbf{D} \rightarrow \text{FEM}(T)$ such that G equals the composite $\mathbf{D} \rightarrow \text{FEM}(T) \rightarrow \mathbf{C}$.

Beck's monadicity theorem characterizes monadic functors. The usual proof rests on the fact that an algebra (A, a) can be canonically viewed as the coequalizer of

$$(\mu_A, Ta): (T^2(A), \mu_{TA}) \rightrightarrows (T(A), \mu_A)$$

and this coequalizer splits in \mathbf{C} . Moreover, one can show that the forgetful functor $\mathbf{C}^T \rightarrow \mathbf{C}$ creates those coequalizers that split in \mathbf{C} , and this in fact characterizes \mathbf{C}^T up to equivalence. In the dagger setting, it turns out that there is no need to discuss split coequalizers. The central observation is that the Frobenius law gives additional properties to a FEM-algebra when viewed as a coequalizer.

Definition 6.8.2. A coequalizer $e: B \rightarrow C$ of $f, g: A \rightrightarrows B$ is a *Frobenius coequalizer* if $g^\dagger f = e^\dagger e$.

Note that Frobenius coequalizers are unique up to unitary isomorphism by Theorem 4.3.6 whenever they exist. However, a consequence of the Frobenius law for a coequalizer is that $f^\dagger g = g^\dagger f$, so Frobenius coequalizers might not exist even when f, g has dagger coequalizers. Moreover, if say $f = 0$, then the existence of a Frobenius coequalizer implies that $e = 0$ for the coequalizer, whence $g = 0$. Hence Frobenius coequalizers are rare and do not seem to provide an interesting class of dagger (co)limits apart from their use in dagger monadicity. The name is explained by the canonical example below.

Example 6.8.3. Let T be a dagger Frobenius monad on \mathbf{C} . The associativity law of an EM-algebra says that a is an algebra homomorphism $a: (T(A), \mu_A) \rightarrow (A, a)$. In fact, it is a coequalizer of $(\mu_A, Ta): (T^2(A), \mu_{TA}) \rightrightarrows (T(A), \mu_A)$. This coequalizer is Frobenius precisely when (A, a) is.

Theorem 6.8.4. Let $G: \mathbf{D} \rightarrow \mathbf{C}$ be a dagger functor. Then G is dagger monadic iff the following conditions hold:

- (i) G has a dagger adjoint.
- (ii) G reflects unitary isomorphisms
- (iii) If $f, g: A \rightrightarrows B$ in \mathbf{D} is a pair such that $(U(f), U(g))$ has Frobenius coequalizer in \mathbf{C} , then (f, g) has a Frobenius coequalizer in \mathbf{D} preserved by G .

Proof. Note that conditions (i)-(iii) are preserved by dagger equivalence. Hence to prove the implication from left to right, it suffices to prove that $\text{FEM}(T) \rightarrow \mathbf{C}$ satisfies (i)-(iii) for any dagger Frobenius monad T . Conditions (i) and (ii) are clear, so we move on to (iii). So, consider a pair $f, g: (A, a) \rightrightarrows (B, b)$ in $\text{FEM}(T)$ that has a Frobenius coequalizer in \mathbf{C} given by $e: B \rightarrow C$. As T factors as a dagger adjunction, it is cocontinuous so that Te is the coequalizer of Tf, Tg . This implies the existence of a map $c: T(C) \rightarrow C$ making the square

$$\begin{array}{ccc} T(B) & \xrightarrow{Te} & T(C) \\ b \downarrow & & \downarrow c \\ B & \xrightarrow{e} & C \end{array}$$

commute. Next, we observe that since T is the composite of two dagger adjoints, it preserves all limits. Hence Te is a coequalizer and in particular an epimorphism, and ditto for T^2e .¹ Using this, the FEM-properties of (C, c) can be deduced from the corresponding properties of (B, b) and (A, a) .

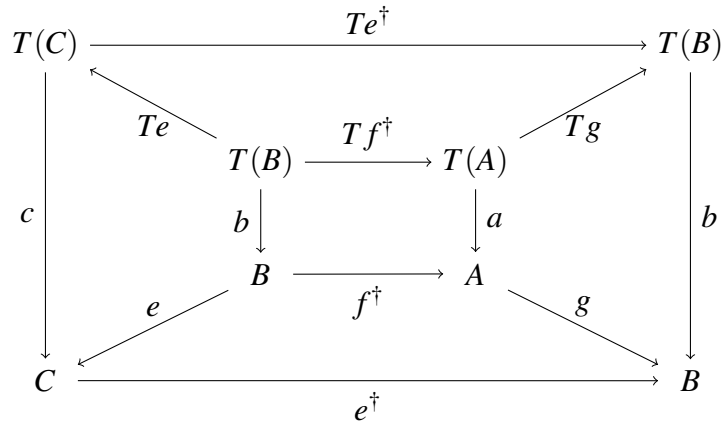
For associativity (C, c) , it suffices to prove that the diagram

$$\begin{array}{ccccc} T^2(C) & \xrightarrow{Tc} & & & T(C) \\ & \swarrow T^2e & & & \nearrow Te \\ & & T^2(B) & \xrightarrow{Tb} & T(B) \\ & & \mu \downarrow & & \downarrow b \\ & & T(B) & \xrightarrow{b} & B \\ & \swarrow Te & & & \searrow e \\ T(C) & \xrightarrow{c} & & & C \end{array}$$

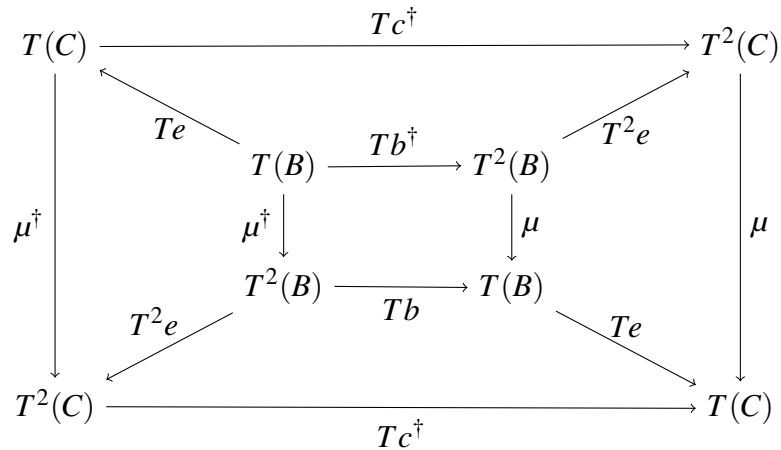
commutes. The trapezoid on the left commutes by naturality of μ and the square in the center is the associative law of (B, b) . The other trapezoids commute by definition of c . The unit law is proved similarly.

¹The usual proof assumes that e is a split coequalizer and use this to infer that Te and T^2e are epi.

To prove that (C, c) satisfies the Frobenius law, consider first the diagram:



The trapezoids on the top and bottom commute because the coequalizer is Frobenius. The trapezoid on the left commutes because e is an algebra homomorphism and the one on the right because g is. The square in the middle commutes because f^\dagger is the dagger of a FEM-algebra homomorphism. As Te is epi the outer rectangle commutes. In other words, e^\dagger is a homomorphism $(C, c) \rightarrow (B, b)$. This is the reason the trapezoid on the top of the diagram



commutes, and the bottom one commutes because e is an algebra homomorphism. The left and right trapezoids commute because μ^\dagger and μ are natural. The diagram in the middle commutes because (B, b) is FEM. Since Te is epi, this implies that the outer rectangle commutes, giving us the Frobenius law for (C, c) . Hence

$$f, g: (A, a) \rightrightarrows (B, b) \xrightarrow{e} (C, c)$$

is a diagram in $FEM(T)$, and it is easy to see that it is a Frobenius coequalizer in $FEM(T)$.

For the other direction, let G satisfy (i)-(iii) and let $F: \mathbf{C} \rightarrow \mathbf{D}$ be its dagger adjoint. Write (T, μ, η) for the dagger Frobenius monad on \mathbf{C} generated by this dagger adjunction, and let $J: \mathbf{D} \rightarrow \text{FEM}(T)$ be the comparison functor from Theorem 6.3.9. We wish to show that J forms a part of a dagger equivalence, so by Lemma 3.1.5 it suffices to show that it is full, faithful and unitarily essentially surjective. We already know that it is full.

To show that J is unitarily essentially surjective, fix a FEM-algebra (A, a) . Now $\varepsilon_{FA}, F(a): FGF(A) \rightrightarrows F(A)$ is a pair in \mathbf{D} which G maps to $\mu_A, Ta: T^2(A) \rightrightarrows T(A)$ in \mathbf{C} . The latter pair has a Frobenius coequalizer in \mathbf{C} , so by (iii) the pair

$$\varepsilon_{FA}, F(a): FGF(A) \rightrightarrows F(A)$$

has a Frobenius coequalizer $e: F(A) \rightarrow B$ in \mathbf{D} preserved by G . As seen in the first part of the proof GB has a unique FEM-algebra structure $b: T(B) \rightarrow B$ such that UJe is a homomorphism, and this in fact makes $(GB, b) = JB$ into a Frobenius coequalizer of $(\mu_A, Ta): (T^2(A), \mu_{TA}) \rightrightarrows (T(A), \mu_A)$ in $\text{FEM}(T)$. As Frobenius coequalizers are unique up to unitary iso, this implies that JB is unitarily isomorphic to (A, a) , as desired.

It remains to prove faithfulness of J , for which it suffices to prove faithfulness of G . So let $f, g: A \rightrightarrows B$ be a pair in \mathbf{D} with $G(f) = G(g)$. This is done exactly as in the non-dagger case: by naturality of the counit $\varepsilon: FG \rightarrow \text{id}_{\mathbf{D}}$ we have

$$f\varepsilon_A = \varepsilon_A FG(f) = \varepsilon_A FG(g) = g\varepsilon_A$$

so it suffices to prove that ε_A is epi. The pair $\varepsilon_{FGA}, FG\varepsilon_A: FGF(A) \rightrightarrows FG(A)$ in \mathbf{D} is mapped by G to the pair $G\varepsilon_{FGA}, GF\varepsilon_A: GF(A) \rightrightarrows GF(A)$ in \mathbf{C} having a Frobenius coequalizer $G\varepsilon_A: GF(A) \rightarrow G(A)$. Hence (iii) implies that

$$\varepsilon_{FGA}, FG\varepsilon_A: FGF(A) \rightrightarrows FG(A)$$

has a Frobenius coequalizer $q: FGA \rightarrow Q$ in \mathbf{C} preserved by G , and hence $\varepsilon_A: GF(A) \rightarrow A$ factors through q as $\varepsilon_A = fq$ for some F . As this coequalizer is preserved by G , the morphism $G(f)$ is a unitary and hence by (ii) f is unitary in \mathbf{D} , which implies that ε_A is epic, completing the proof. □

Corollary 6.8.5. *Dagger monadic functors are closed under composition*

Proof. Conditions (i)-(iii) from Theorem 6.8.4 are clearly closed under composition. □

Here dagger category theory deviates from ordinary category theory: ordinary monadic functors are not closed under composition (an example is that Torsion-free abelian groups are monadic over abelian groups which are monadic over sets, but the composite is not monadic [25, Counterexample 4.6.4]) but dagger monadic functors are.

Chapter 7

Arrows

7.1 Introduction

Reversible computing studies settings in which all processes can be reversed: programs can be run backwards as well as forwards. Its history goes back at least as far as 1961, when Landauer formulated his physical principle that logically irreversible manipulation of information costs work. This sparked the interest in developing reversible models of computation as a means to making them more energy efficient. Reversible computing has since also found applications in high-performance computing [126], process calculi [40], probabilistic computing [136], quantum computing [129], and robotics [127].

There are various theoretical models of reversible computations. The most well-known ones are perhaps Bennett’s reversible Turing machines [21] and Toffoli’s reversible circuit model [142]. There are also various other models of reversible automata [117, 103] and combinator calculi [2, 89].

We are interested in models of reversibility suited to functional programming languages. Functional languages are interesting in a reversible setting for two reasons. First, they are easier to reason and prove properties about, which is a boon if we want to understand the logic behind reversible programming. Second, they are not stateful by definition, which eases reversing programs. It is fair to say that existing reversible functional programming languages [90, 154] still lack various desirable constructs familiar from the irreversible setting.

Irreversible functional programming languages like Haskell naturally take semantics in categories. The objects interpret types, and the morphisms interpret functions. Functional languages are by definition not stateful, and their categorical semantics only

models pure functions. However, sometimes it is useful to have non-functional side-effects, such as exceptions, input/output, or indeed even state. Irreversible functional languages can handle this elegantly using monads [116] or more generally arrows [84].

A word on terminology. We call a computation $a: X \rightarrow Y$ *reversible* when it comes with a specified partner computation $a^\dagger: Y \rightarrow X$ in the opposite direction. This implies nothing about possible side-effects. Saying that a computation is *partially invertible* is stronger, and requires $a \circ a^\dagger \circ a = a$. Saying that it is *invertible* is even stronger, and requires $a \circ a^\dagger$ and $a^\dagger \circ a$ to be identities. We call this partner of a reversible computation its *dagger*. In other words, reversible computing for us concerns dagger arrows on dagger categories, and is modelled using involutions [77]. In an unfortunate clash of terminology, categories of partially invertible maps are called inverse categories [36], and categories of invertible maps are called groupoids [50]. Thus, inverse arrows on inverse categories concern partially invertible maps.

We develop *dagger arrows* and *inverse arrows*, which are useful in two ways:

- We illustrate the reach of these notions by exhibiting many fundamental reversible computational side-effects that are captured (in Section 7.3), including: pure reversible functions, information effects, reversible state, serialization, vector transformations, dagger Frobenius monads [76, 77], recursion [93], and super-operators. To remain within the scope of the thesis, we treat each example informally from the perspective of programming languages, but formally from the perspective of category theory.
- We prove that these notions behave well mathematically (in Section 7.4): whereas arrows are monoids in a category of profunctors [88], dagger arrows and inverse arrows are involutive monoids.

This chapter aims to inform design principles of sound reversible programming languages. The main contribution is to match desirable programming concepts to precise category theoretic constructions. As such, it is written from a theoretical perspective. To make examples more concrete for readers with a more practical background, we adopt the syntax of a typed first-order reversible functional programming language with type classes. We begin with preliminaries on reversible base categories (in Section 7.2).

7.2 Inverse categories

Definition 7.2.1. A (monoidal) inverse category is a (monoidal) dagger category of partial isometries where positive maps commute: $f \circ f^\dagger \circ f = f$ and $f^\dagger \circ f \circ g^\dagger \circ g = g^\dagger \circ g \circ f^\dagger \circ f$ for all maps $f: X \rightarrow Y$ and $g: X \rightarrow Z$.

Every groupoid is an inverse category. Another example of an inverse category is **Pinj**, whose objects are sets, and morphisms $X \rightarrow Y$ are partial injections: $R \subseteq X \times Y$ such that for each $x \in X$ there exists at most one $y \in Y$ with $(x, y) \in R$, and for each $y \in Y$ there exists at most one $x \in X$ with $(x, y) \in R$. It is a monoidal inverse category under either Cartesian product or disjoint union.

Definition 7.2.2. A dagger category is said to have *inverse products* [53] if it is a symmetric monoidal dagger category with a natural transformation $\Delta_X: X \rightarrow X \otimes X$ making the following diagrams commute:

$$\begin{array}{ccc}
 X & \xrightarrow{\Delta_X} & X \otimes X \\
 & \searrow \Delta_X & \downarrow \sigma_{X,X} \\
 & & X \otimes X
 \end{array}
 \quad
 \begin{array}{ccc}
 X & \xrightarrow{\Delta_X} & X \otimes X \\
 \downarrow \Delta_X & & \downarrow \Delta_X \otimes \text{id} \\
 X \otimes X & \xrightarrow{\text{id} \otimes \Delta_X} & X \otimes (X \otimes X) \xrightarrow{\alpha} (X \otimes X) \otimes X
 \end{array}$$

$$\begin{array}{ccc}
 X & \xrightarrow{\Delta_X} & X \otimes X \\
 & \searrow \text{id} & \downarrow \Delta_X^\dagger \\
 & & X
 \end{array}
 \quad
 \begin{array}{ccc}
 X \otimes X & \xrightarrow{\text{id} \otimes \Delta_X} & X \otimes (X \otimes X) \\
 \downarrow \Delta \otimes \text{id} & \searrow \Delta_X^\dagger & \downarrow (\Delta_X^\dagger \otimes \text{id}) \circ \alpha \\
 (X \otimes X) \otimes X & \xrightarrow{(\text{id} \otimes \Delta_X^\dagger) \circ \alpha^\dagger} & X \otimes X
 \end{array}$$

These diagrams express cocommutativity, coassociativity, speciality and the Frobenius law.

Another useful monoidal product, here on inverse categories, is a disjointness tensor, defined in the following way (see [53]):

Definition 7.2.3. An inverse category is said to have a *disjointness tensor* if it is equipped with a symmetric monoidal tensor product $- \oplus -$ such that its unit 0 is a zero object, and the canonical *quasi-injections*

$$i_1 := X \xrightarrow{\rho_X^{-1}} X \oplus 0 \xrightarrow{X \oplus 0, Y} X \oplus Y \quad i_2 := Y \xrightarrow{\lambda_Y^{-1}} 0 \oplus Y \xrightarrow{0, X \oplus Y} X \oplus Y$$

are jointly epic.

For example, **Pinj** has inverse products $\Delta_X: X \rightarrow X \otimes X$ with $x \mapsto (x, x)$, and a disjointness tensor where $X \oplus Y$ is given by the tagged disjoint union of X and Y (the unit of which is \emptyset).

Inverse categories can also be seen as certain instances of restriction categories. Informally, a restriction category models partially defined morphisms, by assigning to each $f: A \rightarrow B$ a morphism $\bar{f}: A \rightarrow A$ that is the identity on the domain of definition of f and undefined otherwise. For more details, see [36].

Definition 7.2.4. A *restriction category* is a category equipped with an operation that assigns to each $f: A \rightarrow B$ a morphism $\bar{f}: A \rightarrow A$ such that:

- $f \circ \bar{f} = f$ for every f ;
- $\bar{f} \circ \bar{g} = \bar{g} \circ \bar{f}$ whenever $\text{dom } f = \text{dom } g$;
- $\overline{g \circ f} = \bar{g} \circ \bar{f}$ whenever $\text{dom } f = \text{dom } g$;
- $\bar{g} \circ f = f \circ \overline{g \circ f}$ whenever $\text{dom } g = \text{cod } f$.

A *restriction functor* is a functor F between restriction categories with $F(\bar{f}) = \overline{F(f)}$.

A *monoidal restriction category* is a restriction category with a monoidal structure for which $\otimes: \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ is a restriction functor.

A morphism f in a restriction category is a *partial isomorphism* if there is a morphism g such that $g \circ f = \bar{f}$ and $f \circ g = \bar{g}$. Given a restriction category \mathbf{C} , define $\text{Inv}(\mathbf{C})$ to be the wide subcategory of \mathbf{C} having all partial isomorphisms of \mathbf{C} as its morphisms.

An example of a monoidal restriction category is **PFn**, whose objects are sets, and whose morphisms $X \rightarrow Y$ are partial functions: $R \subseteq X \times Y$ such that for each $x \in X$ there is at most one $y \in Y$ with $(x, y) \in R$. The restriction \bar{R} is given by $\{(x, x) \mid \exists y \in Y: (x, y) \in R\}$.

Remark 7.2.5. Inverse categories could equivalently be defined as either categories in which every morphism f satisfies $f = f \circ g \circ f$ and $g = g \circ f \circ g$ for a unique morphism g , or as restriction categories in which all morphisms are partial isomorphisms [36, Theorem 2.20]. It follows that functors between inverse categories automatically preserve daggers and that $\text{Inv}(\mathbf{C})$ is an inverse category.

It follows, in turn, that an inverse category with inverse products is a monoidal inverse category: because $X \otimes -$ and $- \otimes Y$ are endofunctors on an inverse category, they preserve daggers, so that by bifactoriality $- \otimes -$ does as well.

7.3 Arrows as an interface for reversible effects

Arrows are a standard way to encapsulate computational side-effects in a functional (irreversible) programming language [83, 84]. This section extends the definition to reversible settings, namely to dagger arrows and inverse arrows. We argue that these notions are “right”, by exhibiting a large list of fundamental reversible side-effects that they model. We start by recalling irreversible arrows.

Definition 7.3.1. An *arrow* on a symmetric monoidal category \mathbf{C} is a functor $A: \mathbf{C}^{\text{op}} \times \mathbf{C} \rightarrow \mathbf{Set}$ with operations

$$\begin{aligned} \text{arr} &: (X \rightarrow Y) \rightarrow A X Y \\ (\ggg) &: A X Y \rightarrow A Y Z \rightarrow A X Z \\ \text{first}_{X,Y,Z} &: A X Y \rightarrow A (X \otimes Z) (Y \otimes Z) \end{aligned}$$

that satisfy the following laws:

$$(a \ggg b) \ggg c = a \ggg (b \ggg c) \quad (7.3.1)$$

$$\text{arr}(g \circ f) = \text{arr } f \ggg \text{arr } g \quad (7.3.2)$$

$$\text{arr id} \ggg a = a = a \ggg \text{arr id} \quad (7.3.3)$$

$$\text{first}_{X,Y,I} a \ggg \text{arr } \rho_Y = \text{arr } \rho_X \ggg a \quad (7.3.4)$$

$$\text{first}_{X,Y,Z} a \ggg \text{arr}(\text{id}_Y \otimes f) = \text{arr}(\text{id}_X \otimes f) \ggg \text{first}_{X,Y,Z} a \quad (7.3.5)$$

$$(\text{first}_{X,Y,Z \otimes V} a) \ggg \text{arr } \alpha_{Y,Z,V} = \text{arr } \alpha_{X,Z,V} \ggg \text{first}(\text{first } a) \quad (7.3.6)$$

$$\text{first}(\text{arr } f) = \text{arr}(f \otimes \text{id}) \quad (7.3.7)$$

$$\text{first}(a \ggg b) = (\text{first } a) \ggg (\text{first } b) \quad (7.3.8)$$

where we use the functional programming convention to write $A X Y$ for $A(X, Y)$ and $X \rightarrow Y$ for $\text{hom}(X, Y)$. The *multiplicative fragment* consists of above data except first , satisfying all laws except those mentioning first ; we call this a *weak arrow*.

Define $\text{second}(a)$ by $\text{arr}(\sigma) \ggg \text{first}(a) \ggg \text{arr}(\sigma)$, using the symmetry, so analogues of (7.3.4)–(7.3.8) are satisfied. Arrows makes sense for (nonsymmetric) monoidal categories if we add this operation and these laws.

Definition 7.3.2. A *dagger arrow* is an arrow on a monoidal dagger category with an

additional operation $\text{inv} : A X Y \rightarrow A Y X$ satisfying the following laws:

$$\text{inv}(\text{inv } a) = a \quad (7.3.9)$$

$$\text{inv } a \gg\gg \text{inv } b = \text{inv}(b \gg\gg a) \quad (7.3.10)$$

$$\text{arr}(f^\dagger) = \text{inv}(\text{arr } f) \quad (7.3.11)$$

$$\text{inv}(\text{first } a) = \text{first}(\text{inv } a) \quad (7.3.12)$$

An *inverse arrow* is a dagger arrow on a monoidal inverse category such that:

$$(a \gg\gg \text{inv } a) \gg\gg a = a \quad (7.3.13)$$

$$(a \gg\gg \text{inv } a) \gg\gg (b \gg\gg \text{inv } b) = (b \gg\gg \text{inv } b) \gg\gg (a \gg\gg \text{inv } a) \quad (7.3.14)$$

The *multiplicative fragment* consists of above data except `first`, satisfying all laws except those mentioning `first`.

Remark 7.3.3. There is some redundancy in the definition of an inverse arrow: (7.3.13) and (7.3.14) imply (7.3.11) and (7.3.12); and (7.3.11) implies $\text{inv}(\text{arrid}) = \text{arrid}$.

Like the arrow laws (7.3.1)–(7.3.8), in a programming language with inverse arrows, the burden is on the programmer to guarantee (7.3.9)–(7.3.14) for their implementation. If that is done, the language guarantees arrow inversion.

Remark 7.3.4. Now follows a long list of examples of inverse arrows, described in a typed first-order reversible functional pseudocode with type classes, inspired by Theseus [90, 89], the revised version of Rfun (briefly described in [94]), and Haskell. Type classes are a form of interface polymorphism: A type class is defined by a class specification containing the signatures of functions that a given type must implement in order to be a member of that type class (often, type class membership also informally requires the programmer to ensure that certain equations are required of their implementations). For example, the *Functor* type class (in Haskell) is given by the class specification

class Functor f where

$fmap : (a \rightarrow b) \rightarrow f a \rightarrow f b$

with the additional informal requirements that $fmap \text{ id} = \text{id}$ and $fmap (g \circ f) = (fmap g) \circ (fmap f)$ must be satisfied for all instances. For example, lists in Haskell satisfy these equations when defining *fmap* as the usual *map* function, i.e.:

instance *Functor List* **where**

$$\begin{aligned} fmap & : (a \rightarrow b) \rightarrow List\ a \rightarrow List\ b \\ fmap\ f\ [] & = [] \\ fmap\ f\ (x::xs) & = (f\ x)::(fmap\ f\ xs) \end{aligned}$$

While higher-order reversible functional programming is fraught, aspects of this can be mimicked by means of parametrized functions. A parametrized function is a function that takes parts of its input statically (i.e., no later than at compile time), in turn lifting the first-order requirement on these inputs. To separate static and dynamic inputs from one another, two distinct function types are used: $a \rightarrow b$ denotes that a must be given statically, and $a \leftrightarrow b$ (where a and b are first-order types) denotes that a is passed dynamically. As the notation suggests, functions of type $a \leftrightarrow b$ are reversible. For example, a parametrized variant of the reversible map function can be defined as a function $map : (a \leftrightarrow b) \rightarrow ([a] \leftrightarrow [b])$. Thus, map itself is *not* a reversible function, but given statically any reversible function $f : a \leftrightarrow b$, the parametrized $map\ f : ([a] \leftrightarrow [b])$ is.

Given this distinction between static and dynamic inputs, the signature of arr becomes $(X \leftrightarrow Y) \rightarrow A\ X\ Y$. We will see later that Arrows on \mathbf{C} can be modelled categorically as monoids in the functor category $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$ [88]. Definition 7.3.1 uses the original signature, because this distinction is not present in the irreversible case. Fortunately, the semantics of arrows remain the same whether or not this distinction is made.

Example 7.3.5. (*Pure functions*) A trivial example of an arrow is the identity arrow $\text{hom}(-, +)$ which adds no computational side-effects at all. This arrow is not as boring as it may look at first. If the identity arrow is an inverse arrow, then the programming language in question is both *invertible* and *closed under program inversion*: any program p has a semantic inverse $\llbracket p \rrbracket^\dagger$ (satisfying certain equations), and the semantic inverse coincides with the semantics $\llbracket \text{inv}(p) \rrbracket$ of another program $\text{inv}(p)$. As such, inv must be a sound and complete *program inverter* (see also [97]) on pure functions; not a trivial matter at all.

Example 7.3.6. (*Information effects*) James and Sabry's *information effects* [89] explicitly expose creation and erasure of information as effects. This type-and-effect system captures irreversible computation inside a pure reversible setting.

Categorically, they start by working in a model of the language Π , a fragment of the language Π^o studied in Section 5.6 without recursive types or traces. Since the language can only denote isomorphisms, we may assume that the model is a rig groupoid \mathbf{C} .

If \mathbf{C} is small, it carries an arrow, where $A(X, Y)$ is the disjoint union of $\text{hom}(X \otimes H, Y \otimes G)$ where G and H range over all objects, and morphisms $X \otimes H \rightarrow Y \otimes G$ and $X \otimes H' \rightarrow Y \otimes G'$ are identified when they are equal up to coherence isomorphisms. This is an inverse arrow, where $\text{inv}(a)$ is simply a^{-1} . It supports the following additional operations:

$$\begin{aligned} \text{erase} &= [\sigma : X \otimes 1 \rightarrow 1 \otimes X]_{\simeq} \in A(X, 1), \\ \text{create}_X &= [\sigma : 1 \otimes X \rightarrow X \otimes 1]_{\simeq} \in A(1, X). \end{aligned}$$

James and Sabry show how a simply-typed first order functional irreversible language translates into π using this inverse arrow to build implicit communication with a global heap H and garbage dump G . They then augment the irreversible language with iteration and natural numbers, and extend the original translation, now translating into Π^o .

Example 7.3.7. (Reversible state) Perhaps the prototypical example of an effect is computation with a mutable store of type S . In the irreversible case, such computations are performed using the state monad $\text{State } S X = S \multimap (X \otimes S)$, where $S \multimap -$ is the right adjoint to $- \otimes S$, and can be thought of as a function type. Morphisms in the corresponding Kleisli category are morphisms of the form $X \rightarrow S \multimap (Y \otimes S)$ in the ambient monoidal closed category. In this formulation, the current state is fetched by $\text{get} : \text{State } S S$ defined as $\text{get } s = (s, s)$, while the state is (destructively) updated by $\text{put} : S \rightarrow \text{State } S 1$ defined as $\text{put } x s = ((), x)$.

Such arrows can not be used as-is in inverse categories, however, as canonical examples (such as **PInj**) fail to be monoidal closed. To get around this, note that it follows from monoidal closure that $\text{hom}(X, S \multimap (Y \otimes S)) \simeq \text{hom}(X \otimes S, Y \otimes S)$, so that $\text{hom}(- \otimes S, - \otimes S)$ is an equivalent arrow that does not depend on closure. With this in mind, we define the *reversible state arrow* with a store of type S :

type $R\text{State } S X Y = X \otimes S \leftrightarrow Y \otimes S$

instance $\text{Arrow } (R\text{State } S)$ **where**

$$\begin{aligned} \text{arr } f (x, s) &= (f \ x, \ s) \\ (a \gggg \ b) (x, s) &= b \ (a \ (x, s)) \\ \text{first } a \ ((x, z), s) &= \text{let } (x', s') = a \ (x, s) \ \text{in } ((x', z), s') \end{aligned}$$

instance $\text{InverseArrow } (R\text{State } S)$ **where**

$$\text{inv } a (y, s) = a^\dagger (y, s)$$

This satisfies the inverse arrow laws. To access the state, we use reversible duplication of values (categorically, this requires the monoidal product to have a natural diagonal $\Delta_X : X \rightarrow X \otimes X$, as inverse products do). Syntactically, this corresponds to the following arrow:

$$\begin{aligned} \text{get} & : RState\ S\ X\ (X \otimes S) \\ \text{get}\ (x, s) & = ((x, s), s) \end{aligned}$$

The inverse to this arrow is $\text{assert} : RState\ S\ (X \otimes S)\ X$, which asserts that the current state is precisely what is given in its second input component; if this fails, the result is undefined. For changing the state, while we cannot destructively update it reversibly, we *can* reversibly update it by a given reversible function with signature $S \leftrightarrow S$. This gives:

$$\begin{aligned} \text{update} & : (S \leftrightarrow S) \rightarrow RState\ S\ X\ X \\ \text{update}\ f\ (x, s) & = (x, f\ s) \end{aligned}$$

This is analogous to how variable assignment works in the reversible programming language Janus [155]: Since destructive updating is not permitted, state is updated by means of built-in reversible update operators, e.g., updating a variable by adding a constant or the contents of another variable to it, etc.

Example 7.3.8. (*Computation in context*) Related to computation with a mutable store is computation with an immutable one; that is, computation within a larger context that remains invariant across execution. In an irreversible setting, this job is typically handled by the *reader monad* (with context of type C), defined as $Reader\ C\ X = C \rightarrow X$. This approach is fundamentally irreversible, however, as the context is “forgotten” whenever a value is computed by supplying it with a context. Even further, it relies on the reversibly problematic notion of monoidal closure.

A reversible version of this idea is one that remembers the context, giving us the reversible Reader arrow:

$$\text{type } Reader\ C\ X\ Y = X \otimes C \leftrightarrow Y \otimes C$$

This is precisely the same as the state arrow – indeed, the instance declarations for arr , $(\gg\gg)$, first , and inv are the same – save for the fact that we additionally require *all* Reader arrows r to satisfy $c = c'$ whenever $r\ (x, c) = (y, c')$. We notice that $\text{arr}\ f$ satisfies this property for all f , whereas $(\gg\gg)$, first , and inv all preserve it. This

resembles the “slice” construction on inverse categories with inverse products; see [53, Sec. 4.4].

As such, while we can provide access to the context via a function defined exactly as *get* for the reversible state arrow, we cannot provide an update function without (potentially) breaking this property – as intended. In practice, the property that the context is invariant across execution can be aided by appropriate interface hiding, i.e. exposing the *Reader* type and appropriate instance declarations and helpers (such as *get* and *assert*) but leaving the *constructor* for *Reader* arrows hidden.

Example 7.3.9. (*Rewriter*) A particularly useful special case of the reversible state arrow is when the store S forms a group. While group multiplication if seen as a function $G \otimes G \leftrightarrow G$ is invertible only in degenerate cases, we can use parametrization to fix the first argument of the multiplication, giving it a much more reasonable signature of $G \rightarrow (G \leftrightarrow G)$. In this way, groups can be expressed as instances of the type class

```
class Group G where
  gunit : G
  gmul  : G → (G ↔ G)
  ginv  : G ↔ G
```

subject to the usual group axioms. This gives us an arrow of the form

```
type Rewriter G X Y = X ⊗ G ↔ Y ⊗ G
```

with instance declarations identical to that of *RState* G , save that we require G to be an instance of the *Group* type class. With this, adding or removing elements from state of type G can then be performed by

```
rewrite      : G → Rewriter G X X
rewrite a (x, b) = (x, gmul a b)
```

which “rewrites” the state by the value a of type G . Note that while the name of this arrow was chosen to be evocative of the *Writer* monad known from irreversible functional programming, as it may be used for similar practical purposes, its construction is substantially different (i.e., irreversible *Writer* arrows are maps of the form $X \rightarrow Y \times M$ where M is a monoid).

Example 7.3.10. (*Vector transformation*) Vector transformations, that is, functions on lists that preserve the length of the list, form another example of inverse arrows. The *Vector* arrow is defined as follows:

type $Vector\ X\ Y = [X] \leftrightarrow [Y]$

instance $Arrow\ (Vector)$ **where**

$arr\ f\ xs = map\ f\ xs$

$(a \gg\gg b)\ xs = b\ (a\ xs)$

$first\ a\ ps = \mathbf{let}\ (xs, zs) = zip^\dagger\ ps\ \mathbf{in}\ zip\ (a\ xs, zs)$

instance $InverseArrow\ (Vector)$ **where**

$inv\ a\ ys = a^\dagger\ ys$

The definition of *first* relies on the usual *map* and *zip* functions, which are defined as follows:

map	$: (a \leftrightarrow b) \rightarrow ([a] \leftrightarrow [b])$	zip	$: ([a], [b]) \leftrightarrow [(a, b)]$
$map\ f\ []$	$= []$	$zip\ ([], [])$	$= []$
$map\ f\ (x::xs)$	$= (f\ x)::(map\ f\ xs)$	$zip\ (x::xs, y::ys)$	$= (x, y)::(zip\ (xs, ys))$

Notice that preservation of length is required for *first* to work: if the arrow *a* does not preserve the length of *xs*, then $zip\ (a\ xs, zs)$ is undefined. However, since *arr* lifts a pure function *f* to a *map* (which preserves length), and ($\gg\gg$) and *inv* are given by the usual composition and inversion, the interface maintains this property.

Example 7.3.11. (*Reversible error handling*) An inverse *weak arrow* comes from reversible computation with a possibility for failure. The weak *Error* arrow is defined using disjointness tensors as follows:

type $Error\ E\ X\ Y = X \oplus E \leftrightarrow Y \oplus E$

instance $WeakArrow\ (Error\ E)$ **where**

$arr\ f\ (InL\ x) = InL\ (f\ x)$

$arr\ f\ (InR\ e) = InR\ e$

$(a \gg\gg b)\ x = b\ (a\ x)$

instance $InverseWeakArrow\ (Error\ E)$ **where**

$inv\ a\ y = a^\dagger\ y$

In this definition, we think of the type *E* as the type of *errors* that could occur during computation. As such, a pure function *f* lifts to a weak arrow which always succeeds with value $f(x)$ when given a non-erroneous input of *x*, and always propagates errors that may have occurred previously.

Raising an error reversibly requires more work than in the irreversible case, as the effectful program that produces an error must be able to *recover* from it in the converse direction. In this way, a reversible *raise* requires two pieces of data: a function $f : X \leftrightarrow E$ that transforms problematic inputs into appropriate errors; and a choice function $p : E \leftrightarrow E \oplus E$ that decides if the error came from this site, injecting it to the left if it did, and to the right if it did not. This can be formalized in an inverse category with a disjointness tensor by saying that the morphisms f and p should satisfy the equations

$$pf = i_1f \quad i_2^\dagger p = p^\dagger i_2 i_2^\dagger p \quad p^\dagger p = \text{id}$$

The choice function p is critical, as in the converse direction it decides whether the error should be handled immediately or later. Thus we define *raise* as follows:

$$\begin{aligned} \text{raise} & : (X \leftrightarrow E) \rightarrow (E \leftrightarrow E \oplus E) \rightarrow \text{Error } E \ X \ Y \\ \text{raise } f \ p \ x & = \text{InR } (p^\dagger (\text{arr } f \ x)) \end{aligned}$$

The converse of *raise* is *handle*, an (unconditional) error handler that maps matching errors back to successful output values. Since unconditional error handling is seldom required, this can be combined with control flow (see Example 7.3.15) to perform conditional error handling, i.e. to only handle errors if they occur.

Example 7.3.12. (*Serialization*) When restricting our attention, as we do here, to only first-order reversible functional programming languages, another example of inverse arrows arises in the form of *serializers*. A serializer is a function that transforms an internal data representation into one more suitable for storage, or for transmission to other running processes. To transform serialized data back into an internal representation, a suitable deserializer is used.

When restricting ourselves to the first-order case, it seems reasonable to assume that all types are serializable, as we thus avoid the problematic case of how to serialize data of function type. As such, assuming that all types X admit a function $\text{serialize} : X \leftrightarrow \text{Serialized } X$ (where $\text{Serialized } X$ is the type of serializations of data of type X), we define the *Serializer* arrow as follows:

$$\text{type } \text{Serializer } X \ Y = X \leftrightarrow \text{Serialized } Y$$

instance *Arrow* (*Serializer*) **where**

$$\begin{aligned} \text{arr } f \ x & = \text{serialize } (f \ x) \\ (a \gggg \ b) \ x & = b \ (\text{serialize}^\dagger (a \ x)) \\ \text{first } a \ (x, z) & = \text{serialize } (\text{serialize}^\dagger (a \ x), z) \end{aligned}$$

instance *InverseArrow* (*Serializer*) **where**
 $inv\ a\ y = serialize\ (a^\dagger\ (serialize\ y))$

Notice how $serialize^\dagger : Serialized\ X \leftrightarrow X$ takes the role of a (partial) deserializer, able to recover the internal representation from serialized data as produced by the serializer. A deserializer of the form $serialize^\dagger$ will often only be partially defined, since many serialization methods allow many different serialized representations of the same data (for example, many textual serialization formats are whitespace insensitive). Of course, all this serializing and unserializing is somewhat inefficient. In spite of these shortcomings, partial deserializers produced by inverting serializers are sufficient for the above definition to satisfy the inverse arrow laws.

Example 7.3.13. (*Dagger Frobenius monads*) Monads are also often used to capture computational side-effects. Arrows are more general. If T is a strong monad, then $A = \text{hom}(-, T(+))$ is an arrow: arr is given by the unit, \ggg is given by Kleisli composition, and first is given by the strength maps. When the base category is a dagger or inverse category modelling reversible pure functions, we can use dagger Frobenius monads from chapter 6.

The Kleisli category of such a monad is again a dagger category by Lemma 6.3.1, giving rise to an operation inv satisfying (7.3.9)–(7.3.11). A dagger Frobenius monad is strong when the strength maps are unitary. In this case (7.3.12) also follows. If the underlying category is an inverse category, then $\mu \circ \mu^\dagger \circ \mu = \mu$, whence $\mu \circ \mu^\dagger = \text{id}$, and (7.3.13)–(7.3.14) follow. Thus, if T is a strong dagger Frobenius monad on a dagger/inverse category, then A is a dagger/inverse arrow.

Example 7.3.14. (*Restriction monads*) There is a notion in between the dagger and inverse arrows of the previous example. A (*strong*) *restriction monad* is a (strong) monad on a (monoidal) restriction category whose underlying endofunctor is a restriction functor. The Kleisli-category of a restriction monad T has a natural restriction structure: just define the restriction of $f : X \rightarrow T(Y)$ to be $\eta_X \circ \bar{f}$. The functors between the base category and the Kleisli category then become restriction functors. If T is a strong restriction monad on a monoidal restriction category \mathbf{C} , then $\text{Inv}(\mathbf{C})$ has an inverse arrow $(X, Y) \mapsto (\text{Inv}(\mathcal{K}\ell(T)))(X, Y)$.

Example 7.3.15. (*Control flow*) While only trivial inverse categories have coproducts [53], less structure suffices for reversible control structures. When the domain and codomain

of an inverse arrow both have disjointness tensors (see Definition 7.2.3), it can often be used to implement *ArrowChoice*, i.e. an arrow with an additional operation

$$\mathit{left} : A\ X\ Y \rightarrow A(X \oplus Z)(Y \oplus Z)$$

subject to various laws enabling conditional constructs. For a simple example, the pure arrow on an inverse category with disjointness tensors implements $\mathit{left} : A\ X\ Y \rightarrow A(X \oplus Z)(Y \oplus Z)$ as

$$\mathit{left}\ f(x, z) = (f\ x, z)$$

The laws of *ArrowChoice* [83] simply reduce to $-\oplus-$ being a bifunctor with natural quasi-injections. More generally, the laws amount to preservation of the disjointness tensor. For the reversible state arrow (Example 7.3.7), this hinges on \otimes distributing over \oplus .

From these libraries for *ArrowChoice* derive additional operations. The splitting combinator

$$(+++) : A\ X\ Y \rightarrow A\ Z\ W \rightarrow A(X \otimes Z)(Y \oplus W)$$

is unproblematic for reversibility, but the fan-in combinator

$$(|||) : A\ X\ Y \rightarrow A\ Z\ Y \rightarrow A(X \oplus Z) \rightarrow Y$$

cannot be defined reversibly, as it explicitly deletes information about which branch was chosen. Reversible conditionals thus require two predicates: one determining the branch to take, and one asserted to join the branches after execution. The branch-joining predicate must be chosen carefully to ensure that it is always true after the *then*-branch, and false after the *else*-branch. This is a standard way of handling branch joining reversibly [155, 154, 54].

Example 7.3.16. (*Superoperators*) Quantum information theory has to deal with environments. The basic category **FHilb** is that of finite-dimensional Hilbert spaces and linear maps. But because a system may be entangled with its environment, the only morphisms that preserve states are the so-called superoperators, or *completely positive* maps [129, 37]: they are not just positive, but stay positive when tensored with an arbitrary ancillary object. In a sense, information about the system may be stored in the environment without breaking the (reversible) laws of nature. This leads to the so-called CPM construction. It is infamously known *not* to be a monad. But it *is* a dagger arrow on **FHilb**, where $A\ X\ Y$ is the set of completely positive maps $X^* \otimes X \rightarrow Y^* \otimes Y$, $\mathit{arr}\ f = f_* \otimes f$, $a \gg b = b \circ a$, $\mathit{first}_{X,Y,Z}\ a = a \otimes \mathit{id}_{Z^* \otimes Z}$, and $\mathit{inv}\ a = a^\dagger$.

Aside from these, other examples do fit the interface of inverse arrows, though they are less syntactically interesting as they must essentially be “built in” to a particular programming language. These include reversible IO, which functions very similarly to irreversible IO, and reversible recursion, which could be used to give a type-level separation between terminating and potentially non-terminating functions, by only allowing fixed points of parametrized functions between arrows rather than between (pure) functions.

7.4 Inverse arrows, categorically

This section explicates the categorical structure of inverse arrows. Arrows on \mathbf{C} can be modelled categorically as monoids in the functor category $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$ [88]. They also correspond to certain identity-on-objects functors $J: \mathbf{C} \rightarrow \mathbf{D}$. The category \mathbf{D} for an arrow A is built by $\mathbf{D}(X, Y) = A X Y$, and `arr` provides the functor J . We will only consider the multiplicative fragment, apart from remark 7.4.6. The operation first can be incorporated in a standard way using strength [88, 13], and poses no added difficulty in the reversible setting.

Clearly, dagger arrows correspond to \mathbf{D} being a dagger category and J a dagger functor, whereas inverse arrows correspond to both \mathbf{C} and \mathbf{D} being inverse categories and J a (dagger) functor. This section addresses the following question: which monoids correspond to dagger arrows and inverse arrows? In the dagger case, the answer is quite simple: the dagger makes $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$ into an involutive monoidal category, and then dagger arrows correspond to involutive monoids. Inverse arrows furthermore require certain diagrams to commute.

Definition 7.4.1. An *involutive monoidal category* is a monoidal category \mathbf{C} equipped with an *involution*: a functor $\overline{(\)}: \mathbf{C} \rightarrow \mathbf{C}$ satisfying $\overline{\overline{f}} = f$ for all morphisms f , together with a natural isomorphism $\chi_{X, Y}: \overline{X} \otimes \overline{Y} \rightarrow \overline{Y \otimes X}$ that makes the following diagrams

commute¹:

$$\begin{array}{ccc}
 \overline{X} \otimes (\overline{Y} \otimes \overline{Z}) & \xrightarrow{\alpha} & (\overline{X} \otimes \overline{Y}) \otimes \overline{Z} \\
 \text{id} \otimes \chi \downarrow & & \downarrow \chi \otimes \text{id} \\
 \overline{X} \otimes \overline{Z} \otimes \overline{Y} & & \overline{Y} \otimes \overline{X} \otimes \overline{Z} \\
 \alpha \downarrow & & \downarrow \chi \\
 \overline{(Z \otimes Y)} \otimes \overline{X} & \xleftarrow{\overline{\alpha}} & \overline{Z} \otimes \overline{(Y \otimes X)}
 \end{array}
 \qquad
 \begin{array}{ccc}
 \overline{\overline{X}} \otimes \overline{\overline{Y}} & \xrightarrow{\chi} & \overline{\overline{Y}} \otimes \overline{\overline{X}} \\
 \text{id} \downarrow & & \downarrow \overline{\chi} \\
 X \otimes Y & \xrightarrow{\text{id}} & \overline{\overline{X}} \otimes \overline{\overline{Y}}
 \end{array}$$

Just like monoidal categories are the natural setting for monoids, involutive monoidal categories are the natural setting for involutive monoids. Any involutive monoidal category has a canonical isomorphism $\phi: I \rightarrow \bar{I}$ [46, Lemma 2.3]:

$$I = \bar{\bar{I}} \xrightarrow{\overline{\rho_{\bar{I}}}^{-1}} \overline{\bar{I}} \otimes I \xrightarrow{\chi_{I, \bar{I}}^{-1}} \bar{I} \otimes \bar{I} = \bar{I} \otimes I \xrightarrow{\rho_{\bar{I}}} \bar{I}$$

Moreover, any monoid M with multiplication m and unit u induces a monoid on \overline{M} with multiplication $\overline{m} \circ \chi_{M, M}$ and unit $\overline{u} \circ \phi$. This monoid structure on \overline{M} allows us to define involutive monoids.

Definition 7.4.2. An *involutive monoid* is a monoid (M, m, u) together with a monoid homomorphism $i: \overline{M} \rightarrow M$ satisfying $i \circ \bar{i} = \text{id}$. A *morphism* of involutive monoids is a monoid homomorphism $f: M \rightarrow N$ making the following diagram commute:

$$\begin{array}{ccc}
 \overline{M} & \xrightarrow{\overline{f}} & \overline{N} \\
 i_M \downarrow & & \downarrow i_N \\
 M & \xrightarrow{f} & N
 \end{array}$$

Our next result lifts the dagger on \mathbf{C} to an involution on the category $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$ of profunctors. First we recall the monoidal structure on that category. It categorifies the dagger monoidal category \mathbf{Rel} of relations of Section 7.2 [24, Section 7.8].

Definition 7.4.3. If \mathbf{C} is small, then $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$ has a monoidal structure, which is a special case of composition of profunctors. In terms of coends [24, 109], the monoidal product is

$$F \otimes G(X, Z) = \int^Y F(X, Y) \times G(Y, Z);$$

¹There is a more general definition allowing a natural isomorphism $\overline{\overline{X}} \rightarrow X$ (see [46] for details), but we only need the strict case.

concretely, $F \otimes G(X, Z) = \coprod_{Y \in \mathbf{C}} F(X, Y) \times G(Y, Z) / \approx$, where \approx is the equivalence relation generated by $(y, F(f, \text{id})(x)) \approx (G(\text{id}, f)(y), x)$, and the action on morphisms is given by $F \otimes G(f, g) := [y, x]_{\approx} \mapsto [F(f, \text{id})x, G(\text{id}, g)y]$. The unit of the tensor product is $\text{hom}_{\mathbf{C}}$.

Proposition 7.4.4. *If \mathbf{C} is a dagger category, then $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$ is an involutive monoidal category when one defines the involution on objects F by $\overline{F}(X, Y) = F(Y, X)$, $\overline{F}(f, g) = F(g^{\dagger}, f^{\dagger})$ and on morphisms $\tau: F \rightarrow G$ by $\overline{\tau}_{X, Y} = \tau_{Y, X}$.*

Proof. First observe that $\overline{(\quad)}$ is well-defined: For any natural transformation of profunctors τ , $\overline{\tau}$ is natural, and $\tau \mapsto \overline{\tau}$ is functorial. Define $\chi_{F, G}$ by the following composite of natural isomorphisms:

$$\begin{aligned} \overline{F} \otimes \overline{G}(X, Z) &\cong \int^Y \overline{F}(X, Y) \times \overline{G}(Y, Z) \text{ by definition of } \otimes \\ &= \int^Y F(Y, X) \times G(Z, Y) \text{ by definition of } \overline{(\quad)} \\ &\cong \int^Y G(Z, Y) \times F(Y, X) \text{ by symmetry of } \times \\ &\cong G \otimes F(Z, X) \text{ by definition of } \otimes \\ &= \overline{G \otimes F}(X, Z) \text{ by definition of } \overline{(\quad)} \end{aligned}$$

Checking that χ make the relevant diagrams commute is routine. \square

Theorem 7.4.5. *If \mathbf{C} is a dagger category, the multiplicative fragments of dagger arrows on \mathbf{C} correspond exactly to involutive monoids in $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$.*

Proof. It suffices to show that the dagger on an arrow corresponds to an involution on the corresponding monoid F . But this is easy: an involution on F corresponds to giving, for each X, Y a map $F(X, Y) \rightarrow F(Y, X)$ subject to some axioms. That this involution is a monoid homomorphism amounts to it being a contravariant identity-on-objects-functor, and the other axiom amounts to it being involutive. \square

Remark 7.4.6. If the operation first is modelled categorically as (internal) strength, axiom (7.3.12) for dagger arrows can be phrased in $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$ as follows: for each object Z of \mathbf{C} , and each dagger arrow M , the profunctor $M_Z = M((-) \otimes Z, (+) \otimes Z)$ is also a dagger arrow, and $\text{first}_{-, +, Z}$ is a natural transformation $M \Rightarrow M_Z$. The arrow laws (7.3.7) and (7.3.8) imply that it is a monoid homomorphism, and the new axiom just states that it is in fact a homomorphism of involutive monoids. For inverse arrows this law is not needed, as any functor between inverse categories is automatically a dagger functor and thus every monoid homomorphism between monoids corresponding to inverse arrows preserves the involution.

Next we set out to characterize which involutive monoids correspond to inverse arrows. Given an involutive monoid M , the obvious approach would be to just state that the map $M \rightarrow M$ defined by $a \mapsto a \circ a^\dagger \circ a$ is the identity. However, there is a catch: for an arbitrary involutive monoid, the map $a \mapsto a \circ a^\dagger \circ a$ is not natural transformation and therefore not a morphism in $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$. To circumvent this, we first require some conditions guaranteeing naturality. These conditions concern endomorphisms, and to discuss them we introduce an auxiliary operation on $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$.

Definition 7.4.7. Let \mathbf{C} be a dagger category. Given a profunctor $M: \mathbf{C}^{\text{op}} \times \mathbf{C} \rightarrow \mathbf{Set}$, define $LM: \mathbf{C}^{\text{op}} \times \mathbf{C} \rightarrow \mathbf{Set}$ by

$$\begin{aligned} LM(X, Y) &= M(X, X), \\ LM(f, g) &= f^\dagger \circ (-) \circ f. \end{aligned}$$

If M is an involutive monoid in $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$, define a subprofunctor of LM :

$$L^+M(X, Y) = \{a^\dagger \circ a \in M(X, X) \mid a \in M(X, Z) \text{ for some } Z\}.$$

Remark 7.4.8. The construction L is a functor $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}] \rightarrow [\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$. There is an analogous construction $RM(X, Y) = M(Y, Y)$ and R^+M , and furthermore $RM = \overline{LM}$. For any monoid M in $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$, LM is a right M -module (and RM a left M -module). Compare Example 7.3.16.

For the rest of this section, assume the base category \mathbf{C} to be an inverse category. This lets us multiply positive arrows by positive pure morphisms. If M is an involutive monoid in $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$, then the map $LM \times L^+(\text{hom}_{\mathbf{C}}) \rightarrow LM$ defined by $(a, g^\dagger \circ g) \mapsto a \circ g^\dagger \circ g$ is natural:

$$\begin{aligned} & LM \times L^+(\text{hom})(f, \text{id}_Y)(a, g^\dagger \circ g) \\ &= (f^\dagger \circ a \circ f, f^\dagger \circ g^\dagger \circ g \circ f) \\ &\mapsto f^\dagger \circ a \circ f \circ f^\dagger \circ g^\dagger \circ g \circ f \\ &= f^\dagger \circ a \circ g^\dagger \circ g \circ f \circ f^\dagger \circ f && \text{because } \mathbf{C} \text{ is an inverse category} \\ &= f^\dagger \circ a \circ g^\dagger \circ g \circ f && \text{because } \mathbf{C} \text{ is an inverse category} \\ &= LM(f, \text{id}_Y)(a \circ g^\dagger \circ g) \end{aligned}$$

Similarly there is a map $L^+(\text{hom}) \times LM \rightarrow LM$ defined by $(g^\dagger \circ g, a) \mapsto g^\dagger \circ g \circ a$. Now the category corresponding to M satisfies $a^\dagger \circ a \circ g^\dagger \circ g = g^\dagger \circ g \circ a^\dagger \circ a$ for all a

and pure g if and only if the following diagram commutes:

$$\begin{array}{ccc}
 L^+M \times L^+(\text{hom}) & \xrightarrow{\quad\quad\quad} & LM \times L^+(\text{hom}) \\
 \sigma \downarrow & & \downarrow \\
 L^+(\text{hom}) \times L^+M & \xrightarrow{\quad\quad\quad} & L^+(\text{hom}) \times LM \xrightarrow{\quad\quad\quad} LM
 \end{array} \tag{7.4.1}$$

If this is satisfied for an involutive monoid M in $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$, then positive arrows multiply. In other words, the map $L^+M \times L^+M \rightarrow LM$ defined by $(a^\dagger \circ a, b^\dagger \circ b) \mapsto a^\dagger \circ a \circ b^\dagger \circ b$ is natural:

$$\begin{aligned}
 & D_M(f, g)(a, a^\dagger, a) \\
 &= (g \circ a \circ f, f^\dagger \circ a^\dagger \circ g^\dagger, g \circ a \circ f) \\
 &\mapsto g \circ a \circ f \circ f^\dagger \circ a^\dagger \circ g^\dagger \circ g \circ a \circ f \\
 &= g \circ a \circ a^\dagger \circ g^\dagger \circ g \circ a \circ f \circ f^\dagger \circ f && \text{by (7.4.1)} \\
 &= g \circ a \circ a^\dagger \circ g^\dagger \circ g \circ a \circ f && \text{because } \mathbf{C} \text{ is an inverse category} \\
 &= g \circ g^\dagger \circ g \circ a \circ a^\dagger \circ a \circ f && \text{by (7.4.1)} \\
 &= g \circ a \circ a^\dagger \circ a \circ f && \text{because } \mathbf{C} \text{ is an inverse category} \\
 &= M(f, g)(a \circ a^\dagger \circ a)
 \end{aligned}$$

This multiplication is commutative iff the following diagram commutes:

$$\begin{array}{ccc}
 L^+M \times L^+M & \xrightarrow{\quad \sigma \quad} & L^+M \times L^+M \\
 & \searrow & \downarrow \\
 & & LM
 \end{array} \tag{7.4.2}$$

Finally, let $D_M \hookrightarrow M \times \overline{M} \times M$ be the diagonal $D_M(X, Y) = \{(a, a^\dagger, a) \mid a \in M(X, Y)\}$.

If M satisfies (7.4.1), then the map $D_M \rightarrow M$ defined by $(a, a^\dagger, a) \mapsto a \circ a^\dagger \circ a$ is natural:

$$\begin{aligned}
& D_M(f, g)(a, a^\dagger, a) \\
&= (g \circ a \circ f, f^\dagger \circ a^\dagger \circ g^\dagger, g \circ a \circ f) \\
&\mapsto g \circ a \circ f \circ f^\dagger \circ a^\dagger \circ g^\dagger \circ g \circ a \circ f \\
&= g \circ a \circ a^\dagger \circ g^\dagger \circ g \circ a \circ f \circ f^\dagger \circ f && \text{by (7.4.1)} \\
&= g \circ a \circ a^\dagger \circ g^\dagger \circ g \circ a \circ f && \text{because } \mathbf{C} \text{ is an inverse category} \\
&= g \circ g^\dagger \circ g \circ a \circ a^\dagger \circ a \circ f && \text{by (7.4.1)} \\
&= g \circ a \circ a^\dagger \circ a \circ f && \text{because } \mathbf{C} \text{ is an inverse category} \\
&= M(f, g)(a \circ a^\dagger \circ a)
\end{aligned}$$

Thus M satisfies $a \circ a^\dagger \circ a = a$ if and only if the following diagram commutes:

$$\begin{array}{ccc}
M & \longrightarrow & D_M \\
& \searrow \text{id} & \downarrow \\
& & M
\end{array} \tag{7.4.3}$$

Hence we have established the following theorem.

Theorem 7.4.9. *Let \mathbf{C} be an inverse category. Then the multiplicative fragments of inverse arrows on \mathbf{C} correspond exactly to involutive monoids in $[\mathbf{C}^{\text{op}} \times \mathbf{C}, \mathbf{Set}]$ making the diagrams (7.4.1)–(7.4.3) commute. \square*

7.5 Applications and related work

As we have seen, inverse arrows capture a variety of fundamental reversible effects. An immediate application of our results would be to retrofit existing typed reversible functional programming languages (e.g., Theseus [90]) with inverse arrows to accommodate reversible effects while maintaining a type-level separation between pure and effectful programs. Another approach could be to design entirely new such programming languages, taking inverse arrows as the fundamental representation of reversible effects. While the Haskell approach to arrows uses typeclasses [83], these are not a priori necessary to reap the benefits of inverse arrows. For example, special syntax for defining inverse arrows could also be used, either explicitly, or implicitly by means of an effect system that uses inverse arrows “under the hood”.

To aid programming with ordinary arrows, a handy notation due to Paterson [119, 120] may be used. The simplest form of this notation is based on process combinators, the central one being

$$p \rightarrow e_1 \leftarrow e_2 = \begin{cases} \text{arr}(\lambda p.e_2) \gggg e_1 & \text{if } p \text{ is fresh for } e_1, \\ \text{arr}(\lambda p.(e_1, e_2)) \gggg \text{app} & \text{otherwise.} \end{cases}$$

Note that if the second branch is used, the arrow must additionally be an instance of *ArrowApply* (so that it is, in fact, a monad). Though we only know of degenerate examples where inverse arrows are instances of *ArrowApply*, this definition is conceptually unproblematic (from the point of view of guaranteeing reversibility) so long as the pure function $\lambda p.e_2$ is first-order and reversible. A more advanced style of this notation is the *do*-notation for arrows, which additionally relies on the arrow combinator

$$\begin{aligned} \text{bind} & : A X Y \rightarrow A (X \otimes Y) Z \rightarrow A X Z \\ f \text{'bind'} g & = (\text{arr}(\text{id}) \&\&\& f) \gggg g . \end{aligned}$$

If the underlying monoidal dagger category has natural coassociative diagonals, for example when it has inverse products, this combinator does exist: the arrow combinator ($\&\&\&$) can be defined as

$$\begin{aligned} (\&\&\&) & : A X Y \rightarrow A X Z \rightarrow A X (Y \otimes Z) \\ f \&\&\& g & = \text{arr}(\text{copy}) \gggg \text{first}(f) \gggg \text{second}(g) \end{aligned}$$

where $\text{copy} : X \rightarrow X \otimes X$ is the natural diagonal (given in pseudocode by $\text{copy } x = (x, x)$), and the combinator second is derived from first in the usual way, i.e., as

$$\begin{aligned} \text{second} & : A X Y \rightarrow A (Z \otimes X) (Z \otimes Y) \\ \text{second } f & = \text{arr}(\text{swap}) \gggg \text{first}(f) \gggg \text{arr}(\text{swap}) \end{aligned}$$

with $\text{swap} : X \otimes Y \leftrightarrow Y \otimes X$ given by $\text{swap } (x, y) = (y, x)$. This allows *do*-notation of the form

$$\mathbf{do} \{p \leftarrow c ; A\} \equiv c \text{'bind'} (\kappa p. \mathbf{do} \{A\}),$$

so soon as the κ -calculus [65] expression $\kappa p. \mathbf{do} \{A\}$ is reversible. Note, however, that *do*-expressions of the form $\mathbf{do} \{c ; A\}$ (i.e., where the output of c is discarded entirely) will fail to be reversible in all but the most trivial cases. Since $\mathbf{do} \{p \leftarrow c ; A\}$ produces a value of an inverse arrow type, closure under program inversion provides a program we might call

$$\mathbf{undo} \{p \leftarrow c ; A\} \equiv \mathit{inv}(\mathbf{do} \{p \leftarrow c ; A\}) \quad .$$

Inverse arrow law (7.3.13) then guarantees that *doing*, then *undoing*, and then *doing* the same operation is the same as *doing* it once.

A pleasant consequence of the semantics of inverse arrows is that inverse arrows are safe: as long as the inverse arrow laws are satisfied, fundamental properties guaranteed by reversible functional programming languages (such as invertibility and closure under program inversion) are preserved. In this way, inverse arrows provide reversible effects as a conservative extension to pure reversible functional programming.

A similar approach to invertibility using arrows is given by bidirectional arrows [11]. However, while the goal of inverse arrows is to add effects to already invertible languages, bidirectional arrows arise as a means to add invertibility to an otherwise uninvertible language. As such, bidirectional arrows have different concerns than inverse arrows, and notably do not guarantee invertibility in the general case.

Bibliography

- [1] S. Abramsky. Retracing some paths in process algebra. In *International Conference on Concurrency Theory*, pages 1–17. Springer, 1996.
- [2] S. Abramsky. A structural approach to reversible computation. *Theoretical Computer Science*, 347(3):441–464, 2005.
- [3] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Logic in computer science, 2004. Proceedings of the 19th Annual IEEE Symposium on*, pages 415–425. IEEE, 2004.
- [4] S. Abramsky and B. Coecke. Categorical quantum mechanics. In K. Engesser, D. M. Gabbay, and D. Lehmann, editors, *Handbook of quantum logic and quantum structures: Quantum logic*, pages 261–324. Elsevier, 2008.
- [5] S. Abramsky, E. Haghverdi, and P. J. Scott. Geometry of interaction and linear combinatory algebras. *Mathematical Structures in Computer Science*, 12(5):625–665, 2002.
- [6] S. Abramsky and A. Jung. Domain theory. *Handbook of logic in computer science*, 3:1–168, 1994.
- [7] J. Adámek and V. Koubek. Least fixed point of a functor. *Journal of Computer and System Sciences*, 19(2):163–178, 1979.
- [8] J. Adámek, S. Milius, and L. S. Moss. Fixed points of functors. *Journal of Logical and Algebraic Methods in Programming*, 95:41–81, 2018.
- [9] J. Adámek and J. Reiterman. Banach’s fixed-point theorem as a base for data-type equations. *Applied Categorical Structures*, 2(1):77–90, 1994.
- [10] J. Adámek and J. Rosicky. *Locally presentable and accessible categories*, volume 189. Cambridge University Press, 1994.

- [11] A. Alimarine, S. Smetsers, A. van Weelden, M. van Eekelen, and R. Plasmeijer. There and back again: Arrows for invertible programming. In *Proceedings of the 2005 ACM SIGPLAN workshop on Haskell*, pages 86–97, 2005.
- [12] P. America and J. Rutten. Solving reflexive domain equations in a category of complete metric spaces. *Journal of Computer and System Sciences*, 39(3):343–375, 1989.
- [13] K. Asada. Arrows are strong monads. In *Proceedings of the third ACM SIGPLAN workshop on Mathematically Structured Functional Programming*, pages 33–42. ACM, 2010.
- [14] S. Awodey. *Category theory*. Oxford University Press, 2010.
- [15] G. Bachman and L. Narici. *Functional Analysis*. Dover Books on Mathematics. Dover Publications, 2012.
- [16] C. Bădescu and P. Panangaden. Quantum alternation: Prospects and problems. In *International Workshop on Quantum Physics and Logic*, volume 195 of *Electronic Proceedings in Theoretical Computer Science*, pages 33–42, 2015.
- [17] M. Barr. Exact categories. In *Exact categories and categories of sheaves*, pages 1–120. Springer, 1971.
- [18] M. Barr. Algebraically compact functors. *Journal of Pure and Applied Algebra*, 82(3):211–231, 1992.
- [19] J. M. Beck. *Triples, algebras and cohomology*. PhD thesis, Columbia University, 1967.
- [20] Marek A Bednarczyk, Andrzej M Borzyszkowski, and Wiesław Pawłowski. Generalized congruences–epimorphisms in cat. *Theory and Applications of Categories*, 5(11):266–280, 1999.
- [21] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, 1973.
- [22] L. Birkedal, K. Støvring, and J. Thamsborg. The category-theoretic solution of recursive metric-space equations. *Theoretical Computer Science*, 411(47):4102–4122, 2010.

- [23] R. Blute, P. Panangaden, and R. Seely. Fock space: a model of linear exponential types, revised version of holomorphic models of exponential types in linear logic. In *Proceedings of MFPS'93*, volume 802 of *Lecture Notes in Computer Science*, 1993.
- [24] F. Borceux. *Handbook of Categorical Algebra 1: Basic Category Theory*. Cambridge University Press, 1994.
- [25] F. Borceux. *Handbook of Categorical Algebra 2: Categories and Structures*. Cambridge University Press, 1994.
- [26] W. J. Bowman, R. P. James, and A. Sabry. Dagger traced symmetric monoidal categories and reversible programming. *Reversible Computation*, 7165:51–56, 2011.
- [27] H.-B. Brinkmann and D. Puppe. *Abelsche und exakte Kategorien, Korrespondenzen*, volume 96. Springer-Verlag, 1969.
- [28] D. Bulacu and B. Torrecillas. On Frobenius and separable algebra extensions in monoidal categories: applications to wreaths. *Journal of Noncommutative Geometry*, 9:707–774, 2015.
- [29] M. S. Burgin. γ -categories and categories with involution. *Uspekhi Matematicheskikh Nauk*, 24(2):221–222, 1969.
- [30] M. S. Burgin. Categories with involution and correspondences in γ -categories. *Transactions of the Moscow Mathematical Society*, 22:161–228, 1970.
- [31] P. Busch, P. J. Lahti, J. Pellonpää, and K. Ylisen. *Quantum measurement*, volume 890. Springer, 2016.
- [32] C. Butz. Regular categories and regular logic. *BRICS Lecture Series LS-98-2*, 1998.
- [33] J. Carette and A. Sabry. Computing with semirings and weak rig groupoids. In *European Symposium on Programming Languages and Systems*, pages 123–148. Springer, 2016.
- [34] D. Chikhladze. Barr's embedding theorem for enriched categories. *Journal of Pure and Applied Algebra*, 215(9):2148–2153, 2011.

- [35] J. Cigler, V. Losert, and P. W. Michor. *Banach modules and functors on categories of Banach spaces*. Dekker, 1979.
- [36] J. R. B. Cockett and S. Lack. Restriction categories I: categories of partial maps. *Theoretical Computer Science*, 270:223–259, 2002.
- [37] B. Coecke and C. Heunen. Pictures of complete positivity in arbitrary dimension. *Information and Computation*, 250:50–58, 2016.
- [38] B. Coecke and A. Kissinger. *Picturing quantum processes*. Cambridge University Press, 2017.
- [39] B. Coecke and D. Pavlovic. Quantum measurements without sums. In G. Chen, L. Kauffman, and S. J. Lomonaco, editors, *The mathematics of quantum computation and technology*, pages 559–596. Taylor and Francis, 2008.
- [40] I. Cristescu, J. Krivine, and D. Varacca. A compositional semantics for the reversible π -calculus. In *Logic in Computer Science*, pages 388–397. IEEE Computer Society, 2013.
- [41] B. Day. On closed categories of functors II. In *Sydney Category Theory Seminar*, number 420 in Lecture Notes in Mathematics, pages 20–54, 1974.
- [42] B. Day and R. Street. Localisation of locally presentable categories. *Journal of Pure and Applied Algebra*, 58(3):227–233, 1989.
- [43] Brian Day and Craig Pastro. Note on Frobenius monoidal functors. *New York J. Math*, 14(733-742):20, 2008.
- [44] A. de Vos and S. de Baerdemacker. Matrix calculus for classical and quantum circuits. *ACM Journal on Emerging Technologies in Computing Systems*, 11(2):9, 2014.
- [45] S. Doplicher and J. E. Roberts. A new duality theory for compact groups. *Inventiones mathematicae*, 98(1):157–218, 1989.
- [46] J. M. Egger. On involutive monoidal categories. *Theory and Applications of Categories*, 25(14):368–393, 2011.
- [47] M. Fiore. An axiomatics and a combinatorial model of creation/annihilation operators. *arXiv:1506.06402*, 2015.

- [48] P. J. Freyd. Algebraically complete categories. In *Category Theory*, pages 95–104. Springer, 1991.
- [49] P. J. Freyd and A. Scedrov. *Categories, allegories*, volume 39. Elsevier, 1990.
- [50] M. J. Gabbay and P. H. Kropholler. Imaginary groups: lazy monoids and reversible computation. *Mathematical Structures in Computer Science*, 23(5):1002–10031, 2013.
- [51] R. Garner. Diagrammatic characterisation of enriched absolute colimits. *Theory and Applications of Categories*, 29(26):775–780, 2014.
- [52] P. Ghez, R. Lima, and J. E. Roberts. w^* -categories. *Pacific Journal of Mathematics*, 120:79–109, 1985.
- [53] B. G. Giles. *An investigation of some theoretical aspects of reversible computing*. PhD thesis, University of Calgary, 2014.
- [54] R. Glück and R. Kaarsgaard. A categorical foundation for structured reversible flowchart languages. In *Mathematical Foundations of Programming Semantics (MFPS)*, volume 336 of *Electronic Notes in Theoretical Computer Science*, pages 155–171. Elsevier, 2018.
- [55] A. Goswami and Z. Janelidze. On the structure of zero morphisms in a quasi-pointed category. *Applied Categorical Structures*, 25(6):1037–1043, 2017.
- [56] J. W. Gray. *Formal category theory: adjointness for 2-categories*, volume 391. Springer, 1974.
- [57] A. S. Green, P. LeFanu Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron. A scalable quantum programming language. *ACM SIGPLAN Notices*, 48(6):333–342, 2013.
- [58] P. A. Grillet. Regular categories. In *Exact categories and categories of sheaves*, pages 121–222. Springer, 1971.
- [59] E. Haghverdi. *A Categorical Approach to Linear Logic, Geometry of Interaction and Full Completeness*. PhD thesis, University of Ottawa, 2000.
- [60] E. Haghverdi. Unique decomposition categories, geometry of interaction and combinatory logic. *Mathematical Structures in Computer Science*, 10(2):205–230, 2000.

- [61] E. Haghverdi and P. J. Scott. Towards a typed geometry of interaction. In *International Workshop on Computer Science Logic*, pages 216–231. Springer, 2005.
- [62] E. Haghverdi and P. J. Scott. A categorical model for the geometry of interaction. *Theoretical Computer Science*, pages 252–274, 2006.
- [63] P. Halmos. *A Hilbert space problem book*. Springer, second edition, 1982.
- [64] Hans Halvorson and Michael Müger. Algebraic quantum field theory. In J. Butterfield and J. Earman, editors, *Philosophy of physics*, volume 2. Elsevier, 2007.
- [65] M. Hasegawa. Decomposing typed lambda calculus into a couple of categorical programming languages. In D. Pitt, D. E. Rydeheard, and P. Johnstone, editors, *Category Theory and Computer Science*, volume 953 of *Lecture Notes in Computer Science*, pages 200–219. Springer, 1995.
- [66] T. Heinosaari and M. Ziman. *The mathematical language of quantum theory*. Cambridge University Press, 2012.
- [67] A. Y. Helemskii. *Lectures and exercises on functional analysis*, volume 233. American Mathematical Society Providence, RI, 2006.
- [68] S. Henry. Complete c^* -categories and a topos theoretic Green-Julg theorem. *arXiv:1512.03290*, 2015.
- [69] C. Heunen. Semimodule enrichment. In *Mathematical Foundations of Programming Semantics*, volume 218 of *Electronic Notes in Theoretical Computer Science*, pages 193–208, 2008.
- [70] C. Heunen. *Categorical quantum models and logics*. Amsterdam University Press, 2009.
- [71] C. Heunen. An embedding theorem for Hilbert categories. *Theory and Applications of Categories*, 22(13):321–344, 2009.
- [72] C. Heunen. On the functor ℓ^2 . *Computation, Logic, Games, and Quantum Foundations*, pages 107–121, 2013.
- [73] C. Heunen, I. Contreras, and A. Cattaneo. Relative Frobenius algebras are groupoids. *Journal of Pure and Applied Algebra*, 217:114–124, 2013.

- [74] C. Heunen and B. Jacobs. Quantum logic in dagger kernel categories. *Order*, 27(2):177–212, 2010.
- [75] C. Heunen, R. Kaarsgaard, and M. Karvonen. Reversible effects as inverse arrows. In *Mathematical Foundations of Programming Semantics (MFPS)*, volume 341 of *Electronic Notes in Theoretical Computer Science*, pages 179–199. Elsevier, 2018.
- [76] C. Heunen and M. Karvonen. Reversible monadic computing. In *Mathematical Foundations of Programming Semantics (MFPS)*, volume 319 of *Electronic Notes in Theoretical Computer Science*, pages 217–237, 2015.
- [77] C. Heunen and M. Karvonen. Monads on dagger categories. *Theory and Applications of Categories*, 31(35):1016–1043, 2016.
- [78] C. Heunen and M. Karvonen. Limits in dagger categories. *arXiv:1803.06651*, 2018.
- [79] C. Heunen and S. Tull. Categories of relations as models of quantum theory. In *International Workshop on Quantum Physics and Logic*, volume 195 of *Electronic Proceedings in Theoretical Computer Science*, pages 247–261, 2015.
- [80] C. Heunen and J. Vicary. *Categories for Quantum Theory: An Introduction*. Oxford University Press.
- [81] N. Hoshino. A representation theorem for unique decomposition categories. *Electronic Notes in Theoretical Computer Science*, 286:213–227, 2012.
- [82] N. Hoshino. Partial traces on additive categories. In *Mathematical Foundations of Programming Semantics (MFPS)*, volume 341 of *Electronic Notes in Theoretical Computer Science*, pages 219–237. Elsevier, 2018.
- [83] J. Hughes. Generalising monads to arrows. *Science of Computer Programming*, 37:67–111, 2000.
- [84] J. Hughes. *Advanced Functional Programming*, volume 3622 of *Lecture Notes in Computer Science*, chapter Programming with Arrows, pages 73–129. Springer, 2005.
- [85] B. Jacobs. Semantics of weakening and contraction. *Annals of Pure and Applied Logic*, 69(1):73–106, 1994.

- [86] B. Jacobs. Coalgebraic walks, in quantum and Turing computation. *FoSSaCS, Lecture Notes in Computer Science*, 6604:12–26, 2011.
- [87] B. Jacobs. Involutive categories and monoids, with a GNS-correspondence. *Foundations of Physics*, 42(7):874–895, 2012.
- [88] B. Jacobs, C. Heunen, and I. Hasuo. Categorical semantics for arrows. *Journal of Functional Programming*, 19(3-4):403–438, 2009.
- [89] R. P. James and A. Sabry. Information effects. In *Principles of Programming Languages*, pages 73–84. ACM, 2012.
- [90] R. P. James and A. Sabry. Theseus: A high level language for reversible computing, 2014. Work-in-progress report at RC 2014, available at <https://www.cs.indiana.edu/~sabry/papers/theseus.pdf>.
- [91] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3):447–468, 1996.
- [92] R. Kaarsgaard. *The Logic of Reversible Computing - Theory and Practice*. PhD thesis, Department of Computer Science, Faculty of Science, University of Copenhagen, 2017.
- [93] R. Kaarsgaard, H. B. Axelsen, and R. Glück. Join inverse categories and reversible recursion. *Journal of Logical and Algebraic Methods in Programming*, 87:33–50, 2017.
- [94] R. Kaarsgaard and M. K. Thomsen. RFun revisited. In M. Waldén, editor, *Proceedings of the 29th Nordic Workshop on Programming Theory*, volume 27 of *TUCS Lecture Notes*, pages 65–67. Turku Centre for Computer Science, 2017.
- [95] M. Karvonen. Biproducts without zero morphisms. *arXiv:1801.06488*, 2018.
- [96] J. Kastl. Inverse categories. *Studien zur Algebra und ihre Anwendungen*, 7:51–60, 1979.
- [97] M. Kawabe and R. Glück. The program inverter Inrv and its structure. In M. V. Hermenegildo and D. Cabeza, editors, *Practical Aspects of Declarative Languages*, volume 3350 of *Lecture Notes in Computer Science*, pages 219–234. Springer, 2005.

- [98] Y. Kawahara. Matrix calculus in i -categories and an axiomatic characterization of relations in a regular category. *Memoirs of the Faculty of Science, Kyushu University. Series A, Mathematics*, 27(2):249–273, 1973.
- [99] Y. Kawahara. Relations in categories with pullbacks. *Memoirs of the Faculty of Science, Kyushu University. Series A, Mathematics*, 27(1):149–173, 1973.
- [100] G. M. Kelly. Coherence theorems for lax algebras and for distributive laws. In *Lecture Notes in Mathematics 420*, pages 281–375. Springer, 1974.
- [101] G. M. Kelly. *Basic concepts of enriched category theory*, volume 64. CUP Archive, 1982.
- [102] A. Kock. Strong functors and monoidal monads. *Archiv der Mathematik*, 23(1):113–120, 1972.
- [103] M. Kutrib and M. Wendlandt. Reversible limited automata. In *Machines, Computations and Universality*, volume 9288 of *Lecture Notes in Computer Science*, pages 113–128, 2015.
- [104] J. Lambek. Diagram chasing in ordered categories with involution. *Journal of Pure and Applied Algebra*, 143(1-3):293–307, 1999.
- [105] M. L. Laplaza. Coherence for distributivity. In *Lecture Notes in Mathematics 281*, pages 29–72. Springer, 1972.
- [106] A. Lauda. Frobenius algebras and ambidextrous adjunctions. *Theory and Applications of Categories*, 16(4):84–122, 2006.
- [107] M. V. Lawson. *Inverse semigroups: the theory of partial symmetries*. World Scientific, 1998.
- [108] D. J. Lehmann and M. B. Smyth. Algebraic specification of data types: A synthetic approach. *Mathematical Systems Theory*, 14(1):97–139, 1981.
- [109] F. Loregian. This is the (co)end, my only (co)friend. *arXiv:1501.02503*, 2015.
- [110] S. Mac Lane. An algebra of additive relations. *Proceedings of the National Academy of Sciences*, 47:1043–1051, 1961.
- [111] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 2nd edition, 1971.

- [112] O. Malherbe, P. J. Scott, and P. Selinger. Partially traced categories. *Journal of Pure and Applied Algebra*, 216(12):2563–2585, 2012.
- [113] Are dagger categories truly evil? MathOverflow-discussion. <https://mathoverflow.net/questions/220032/are-dagger-categories-truly-evil>, 2015. Accessed: 24.10.2018.
- [114] Micah McCurdy and Ross Street. What separable Frobenius monoidal functors preserve? *Cahiers de topologie et géométrie différentielle catégoriques*, 51(1):29–50, 2010.
- [115] B. Mitchell. *Theory of categories*. Academic Press, 1965.
- [116] E. Moggi. Computational lambda-calculus and monads. *Logic in Computer Science*, 1989.
- [117] K. Morita. Two-way reversible multihead automata. *Fundamenta Informaticae*, 110(1–4):241–254, 2011.
- [118] nLab-entry on the principle of equivalence. <https://ncatlab.org/nlab/show/principle+of+equivalence>. Accessed: 24.10.2018.
- [119] R. Paterson. A new notation for arrows. In *International Conference on Functional Programming*, pages 229–240, 2001.
- [120] R. Paterson. Arrows and computation. In J. Gibbons and O. de Moor, editors, *The Fun of Programming*, pages 201–222. Palgrave, 2003.
- [121] D. Pavlovic. Geometry of abstraction in quantum computation. In *Classical and Quantum Information Assurance Foundations and Practice*, number 09311 in Dagstuhl Seminar Proceedings, 2010.
- [122] E. Prugovečki. *Quantum mechanics in Hilbert space*, volume 92. Academic Press, 1982.
- [123] D. Puppe. Korrespondenzen in abelschen Kategorien. *Mathematische Annalen*, 148:1–30, 1962.
- [124] Y. J. Ringard. Mustard watches: an integrated approach to time and food. *Équipe de logique mathématique, prépublications*, 1990. Can be found at <http://girard.perso.math.cnrs.fr/mustard/article.html>, Accessed: 24.10.2018.


- [125] A. Sabry, B. Valiron, and J. K. Vizzotto. From symmetric pattern-matching to quantum control. In *International Conference on Foundations of Software Science and Computation Structures*, pages 348–364. Springer, 2018.
- [126] M. Schordan, D. Jefferson, P. Barnes, T. Oppelstrup, and D. Quinlan. Reverse code generation for parallel discrete event simulation. In *Reversible Computing*, volume 9138 of *Lecture Notes in Computer Science*, pages 95–110. Springer, 2015.
- [127] U. P. Schultz, M. Bordignon, and K. Støy. Robust and reversible execution of self-reconfiguration sequences. *Robotica*, 29(1):35–37, 2011.
- [128] D. Scott. Continuous lattices. In *Toposes, algebraic geometry and logic*, pages 97–136. Springer, 1972.
- [129] P. Selinger. Dagger compact closed categories and completely positive maps. In *Quantum Programming Languages*, volume 170 of *Electronic Notes in Theoretical Computer Science*, pages 139–163, 2007.
- [130] P. Selinger. Idempotents in dagger categories. *Electronic Notes in Theoretical Computer Science*, 210:107–122, 2008.
- [131] P. Selinger. A survey of graphical languages for monoidal categories. In B. Coecke, editor, *New Structures for Physics*, number 813 in *Lecture Notes in Physics*, pages 289–356. Springer, 2009.
- [132] P. Selinger. “The definition of ‘evil’ ” Email to the categories mailing list. <http://permalink.gmane.org/gmane.science.mathematics.categories/5459>, 2010. Accessed: 24.10.2018.
- [133] P. Selinger and B. Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 16(3):527–552, 2006.
- [134] Z. Semadeni and H. Zidenberg. Inductive and inverse limits in the category of Banach spaces. *Bulletin de l’Académie polonaise des sciences. Série des sciences mathématiques, astronomiques, et physiques.*, 13(8):579—583, 1965.
- [135] M. B. Smyth and G. D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM Journal on Computing*, 11(4):761–783, 1982.

- [136] B. Stoddart and R. Lynas. A virtual machine for supporting reversible probabilistic guarded command languages. In *Reversible Computing*, volume 253 of *Electronic Notes in Theoretical Computer Science*, pages 33–56. Elsevier, 2010.
- [137] R. Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2(2):149–168, 1972.
- [138] R. Street. Fibrations and Yoneda’s lemma in a 2-category. In *Category seminar*, pages 104–133. Springer, 1974.
- [139] R. Street. Absolute colimits in enriched categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 24(4):377–379, 1983.
- [140] R. Street. Frobenius monads and pseudomonoids. *Journal of Mathematical Physics*, 45(10):3930–3948, 2004.
- [141] R. Street and R. Walters. Yoneda structures on 2-categories. *Journal of Algebra*, 50(2):350–379, 1978.
- [142] T. Toffoli. Reversible computing. In *International Colloquium on Automata, Languages and Programming*, pages 632–644, 1980.
- [143] Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. <http://homotopytypetheory.org/book>, Institute for Advanced Study, 2013.
- [144] D. Upper. The unsuccessful self-treatment of a case of “writer’s block”. *Journal of Applied Behavior Analysis*, 7(3):497–497, 1974.
- [145] D. H. Van Osdol. Sheaves in regular categories. In *Exact Categories and Categories of Sheaves*, pages 223–239. Springer, 1971.
- [146] J. Vicary. A categorical framework for the quantum harmonic oscillator. *International Journal of Theoretical Physics*, 47(12):3408–3447, 2008.
- [147] J. Vicary. Categorical formulation of quantum algebras. *Communications in Mathematical Physics*, 304(3):765–796, 2011.
- [148] J. Vicary. Completeness of \dagger -categories and the complex numbers. *Journal of Mathematical Physics*, 52(8):082104, 2011.

- [149] M. Wand. Fixed-point constructions in order-enriched categories. *Theoretical Computer Science*, 8(1):13–30, 1979.
- [150] C. E. Watts. Intrinsic characterizations of some additive functors. *Proceedings of the American Mathematical Society*, 11(1):5–8, 1960.
- [151] H. Wolff. Monads and monoids on symmetric monoidal closed categories. *Archiv der Mathematik*, 24(1):113–120, 1973.
- [152] R. J. Wood. Abstract proarrows I. *Cahiers de topologie et géométrie différentielle catégoriques*, 23(3):279–290, 1982.
- [153] R. J. Wood. Proarrows II. *Cahiers de topologie et géométrie différentielle catégoriques*, 26(2):135–168, 1985.
- [154] T. Yokoyama, H. B. Axelsen, and R. Glück. Towards a reversible functional language. In A. De Vos and R. Wille, editors, *Reversible Computation 2011*, volume 7165 of *LNCS*, pages 14–29. Springer, 2012.
- [155] T. Yokoyama and R. Glück. A reversible programming language and its invertible self-interpreter. In *Partial Evaluation and Semantics-Based Program Manipulation. Proceedings*, pages 144–153. ACM Press, 2007.

Index of symbols

- A, B, C, \dots , objects of a category, 7
Ban_{≤1}, Banach spaces and
 non-expansive maps, 89, 94
C, D, ..., (dagger) categories, 7
C_≡, the cofree dagger category on **C**,
 13, 35, 74
[C, D], dagger functors **C** → **D**, 12, 17,
 52
 $\text{dlim}^\Omega D$, the dagger limit of (D, Ω) , 30
DStoch, doubly stochastic matrices, 10
FEM(T), the category of
 FEM-algebras of T , 113, 114,
 128
 f, g, h, \dots , morphisms, 7
 f^\dagger , the dagger of f , 9
 F, G, H, \dots , functors, 7
FHilb, f.d. Hilbert spaces and linear
 maps, 9, 17, 47
FHilb_{≤1}, f.d. Hilbert spaces and
 non-expansive maps, 9
FinRel, finite sets and relations, 10, 34
FPinj, finite sets and partial injections,
 10
Hilb, Hilbert spaces and bounded
 linear maps, 9, 17, 47, 89
Hilb_{≤1}, Hilbert spaces and
 non-expansive maps, 9, 89, 94
 $\text{inv}(a)$, the dagger of an arrow a , 142
[J, C]_† functors and *adjointable*
 transformations, 15, 51
 $\mathcal{K}\ell(T)$ the Kleisli category of a monad,
 110, 114
 $l_A: L \rightarrow D(A)$, one leg of a cone
 $\Delta L \rightarrow D$ or a generic
 morphism denoting the whole
 cone, 7, 30
 ℓ^1 , a functor **Pfn** → **Ban**_{≤1}, 90, 101
 ℓ^2 , a functor **Pinj** → **Hilb**_{≤1}, 89
 \mathbb{N} , natural numbers, often as a
 one-object category, 7
 ω , the category induced by the order
 on \mathbb{N} , 7, 80–106
Pfn, sets and partial functions, 89, 94
Pinj, sets and partial injections, 10, 11,
 34, 89, 94
Rel, sets and relations, 10, 17, 34
Span(**C**), spans of **C**, 10
Span(**FinSet**), spans of finite sets, 34
 st , strength natural transformation, 119
 $\text{Sym}^n X$ the n -th symmetric tensor
 power of X , 33, 92, 98
 (T, μ, η) , a (dagger Frobenius) monad,

- 108
- X, Y, Z, \dots , objects of a category, 7
- ZigZag(C)**, the free dagger category on **C**, 13
- \dashrightarrow , dagger epi, 11
- \dashrightarrow , partial isometry, 11
- \dashrightarrow , unitary, 11
- \dashrightarrow , dagger mono, 11
-  a morphism f in the graphical calculus, 17
- \comult , comultiplication of a comonoid, 18
- μlt , multiplication of a monoid, 18

Index of concepts

- adjointable bifunctor, 68
- adjointable natural transformation, 15, 40
- ambilimit, 73
- antisymmetrized tensor power, 33, 92, 98
- arrow, 141
- biproduct, 75
- canonical fixed point, 83, 84, 94
- comonoid, 18
- cosiometry, 11
- dagger 2-adjunction, 115
- dagger 2-category, 12
- dagger adjoint, 25
- dagger adjunction, 25
 - in a dagger 2-category, 115
- dagger biproduct, 31
- dagger category, 9
 - cofree, 13, 35, 74
 - connected, 33
 - dagger skeletal, 23
 - free, 13
 - (symmetric) monoidal, 16
 - ω -continuous rig, 92, 94
 - rig, 89
 - sheathed, 126
 - unitary, 11
- dagger epimorphism, 11
- dagger equalizer, 31
- dagger equivalence, 21
- dagger fixed point, 83, 84, 94
- dagger Frobenius monad, 108
 - in a dagger 2-category, 115
- dagger functor, 12
 - dagger monadic, 130
- dagger idempotent, 11
- dagger intersection, 31
- dagger kernel, 31
- dagger limit, 30
 - (finitely) based, 45
 - dagger shaped, 32
- dagger monomorphism, 11
- dagger product, 31
- dagger skeleton, 23
- dagger subobject, 11
- disjointness tensor, 139
- embedding-projection chain, 91–93
- embedding-projection pair, 80, 91
- F -algebra, 82
- Fock space, 98
- Frobenius coequalizer, 131

- Frobenius law
 - of a monad, 108
 - of a monoid, 18, 107
 - of an algebra, 110
- graphical calculus, 17
- inverse category, 11
 - monoidal, 139
- inverse products, 139
- involutive monoidal category, 151
- isometry, 11
- monoid, 18
 - dagger Frobenius, 18
 - involutive, 152
- O**-category, 80
- O**-colimit, 80
- partial isometry, 11
- partial isomorphism, 140
- polar decomposition, 55
- positive morphism, 11
- projection, 11
 - splitting of, 31
- restriction category, 140
- rig category, 88
 - ω -continuous, 92, 94
- rig-polynomial functor, 92, 94
- self-adjoint morphism, 11
- symmetrized tensor power, 33, 92, 98
- trace, 99
- unitary morphism, 10