Logical Aspects of Quantum Computation

Daniel Marsden

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary Term 2015

To my family

Acknowledgements

I would first like to thank both my supervisors, Andreas Döring and Samson Abramsky for their advice and guidance through the development of this thesis. I also would like to show my appreciation to the many people in the Quantum Group at Oxford that have helped to provide a friendly and stimulating environment in which to study. I am grateful to Ralf Hinze for enlightening discussions about category theory and string diagrams that helped clarify my appreciation of the essentials of these topics.

The love and support of my family has been vital for this work even being possible, and I am profoundly grateful. Mum, Jez and Andy have been a constant source of kindness, support and humour as far back as I can remember, and have helped me at every point in my progress. Finally, and most importantly, I must thank Nuala, Florin and Kiko for countless wonderful moments and memories.

Abstract

A fundamental component of theoretical computer science is the application of logic. Logic provides the formalisms by which we can model and reason about computational questions, and novel computational features provide new directions for the development of logic. From this perspective, the unusual features of quantum computation present both challenges and opportunities for computer science. Our existing logical techniques must be extended and adapted to appropriately model quantum phenomena, stimulating many new theoretical developments. At the same time, tools developed with quantum applications in mind often prove effective in other areas of logic and computer science. In this thesis we explore logical aspects of this fruitful source of ideas, with category theory as our unifying framework.

Inspired by the success of diagrammatic techniques in quantum foundations, we begin by demonstrating the effectiveness of string diagrams for practical calculations in category theory. We proceed by example, developing graphical formulations of the definitions and proofs of many topics in elementary category theory, such as adjunctions, monads, distributive laws, representable functors and limits and colimits. We contend that these tools are particularly suitable for calculations in the field of coalgebra, and continue to demonstrate the use of string diagrams in the remainder of the thesis.

Our coalgebraic studies commence in chapter 3, in which we present an elementary formulation of a representation result for the unitary transformations, following work developed in a fibrational setting in [Abramsky, 2010]. That paper raises the question of what a suitable "fibred coalgebraic logic" would be. This question is the starting point for our work in chapter 5, in which we introduce a parameterized, duality based framework for coalgebraic logic. We show sufficient conditions under which dual adjunctions and equivalences can be lifted to fibrations of (co)algebras.

We also prove that the semantics of these logics satisfy certain "institution conditions" providing harmony between syntactic and semantic transformations.

We conclude by studying the impact of parameterization on another logical aspect of coalgebras, in which certain fibrations of predicates can be seen as generalized invariants. Our focus is on the lifting of coalgebra structure along a fibration from the base category to an associated total category of predicates. We show that given a suitable parameterized generalization of the usual liftings of signature functors, this induces a "fibration of fibrations" capturing the relationship between the two different axes of variation.

Contents

Chapter 1

Introduction

A major pillar of theoretical computer science is the application of logic. Type theories, logics and models are developed to reason about computational features such as state, polymorphism, concurrency and autonomous agents. Similarly, progress in logic provides new tools and models that can be applied to computational scenarios. The interaction between computer science and logic has been a fruitful one, as computer science produces new challenges, this stimulates new developments and applications in the field of logic. In this relationship, category theory has proved an effective unifying tool, providing an appropriately high level of abstraction, emphasising interaction and transformation and exposing underlying logical structure.

From this perspective, the field of quantum computation presents a wonderful opportunity for theoretical computer scientists. Questions from quantum computation and quantum foundations have encouraged a great deal of innovation in the development and application of modern logic, going all the way back to [Birkhoff and von Neumann, 1936]. Recent explorations of logical aspects of quantum theory, many adopting categorical methods, can be found in [Döring, 2012, 2011, Heunen et al., 2009, 2011, Jacobs, 2012a, Baltag and Smets, 2006, Abramsky and Hardy, 2012]. These techniques have then been exploited in other areas of computer science such as relational databases [Abramsky, 2013] and computational linguistics [Coecke et al., 2010].

We observe a bidirectional flow of developments:

- Logical techniques from computer science are applied to the field of quantum computation. The unusual aspects of quantum computation often require innovation and the development of new logical methods.

- Techniques developed with quantum applications in mind can be adapted and expanded to applications in the broader field of computer science.

The aim of this thesis is to explore logical developments in both these directions, using category theory as our underlying technical tool.

1.1 Calculational Category Theory

The calculational style of mathematics is an approach to formal proofs, originating in the computer science community. The scheme is typically characterized by a goal directed systematic manipulation of equations, similar in flavour to high school arithmetic. The merits of this style of reasoning are discussed in [van Gasteren, 1990, Dijkstra and Scholten, 1990] and discrete mathematics is formulated in a calculational style in [Gries and Schneider, 1993].

String diagrams are a graphical formalism for working in category theory, providing a convenient tool for manipulating morphisms within bicategories [Bénabou, 1967]. For some background see [Street, 1995]. Techniques for applying string diagrams in spirit similar to this paper can be found in [Curien, 2008]. Recent work in quantum computation and quantum foundations has successfully applied diagrammatic calculi based on string diagrams to model quantum systems in monoidal categories with additional structure. See for example [Abramsky and Coecke, 2008, Coecke and Duncan, 2011, Stay and Vicary, 2013, Vicary, 2012, Baez and Stay, 2011] and the comprehensive survey [Selinger, 2011] for more details. Further applications of string diagrams for monoidal categories in more logical settings can be found in [Melliès, 2006, 2012].

Category theory was developed in a calculational style in [Fokkinga, 1992a,b] and [Fokkinga and Meertens, 1994]. In these papers, we are presented with a choice between the traditional commuting diagram style of reasoning about category theory, and a calculational approach based on formal manipulation of the corresponding equations. In chapter 2 we propose string diagrams as an alternative strategy to both of these. These graphical techniques naturally support calculational reasoning, but continue to carry the important type information that is abandoned in the move to conventional symbolic equations. Additionally, string diagram notation “does a lot of the work for free”, an important aspect in choosing notation, as advocated in [Backhouse, 1989]. Issues of associativity, functoriality and naturality are handled silently by the notation, allowing attention to be focused on the essential aspects of the proof. In order to support our claim that string diagrams are a practical tool in this setting, we proceed by example, developing a large part of elementary category theory using graphical definitions and proofs throughout. In sections 2.3 and 2.4 we address areas in which string diagrams may be familiar, for example adjunctions and monads. Here, investigating various kinds of distributive laws leads to particularly compelling examples of the effectiveness of our approach. In order to demonstrate the breadth of applicability of string diagrams, in sections 2.5, 2.6 and 2.7 we also proceed into territory in which they may expected to be less comfortable, for example representable functors, limits and bifunctors.

It is a contention of this thesis that the graphical techniques of chapter 2 are particularly suitable for calculations in the field of coalgebra. Interactions between functors, natural transformations, (co)monads and various distributive laws are commonplace in coalgebraic applications, and hence these techniques should be better appreciated by that community. To that end, as space permits, we shall develop some of the proofs and definitions in the later chapters 3, 5 and 6 in a string diagrammatic style.

1.2 Representing the Unitaries Coalgebraically

Coalgebras [Rutten, 2000] provide models of systems with abstract states and dynamics connecting those states to available observations. They are natural candidates for modelling physical systems, and in particular quantum systems. In quantum mechanics, interactions can be divided into:

- Irreversible probabilistic measurements
- Reversible deterministic state evolution

If we consider measurements as observations upon a system, the dynamics of a suitable coalgebraic model should relate states to measurement behaviour. Optimistically, we may then hope to find the reversible state evolutions as automorphisms of our coalgebraic model. It is well known that coalgebra homomorphisms are functional bisimulations, that is, they *preserve observable behaviour.* Clearly this conflicts with our expectation that changes in state should be observable by measurements, and so this naive scheme cannot succeed.

In [Abramsky, 2010] a coalgebraic representation result for the unitary transformations that describe state evolution in quantum mechanics was developed. A novel fibrational setting was used to provide “enough contravariance” to successfully model these automorphisms, building upon earlier work of the same author using Chu spaces [Abramsky, 2009].

In chapter 3, we develop a more elementary version of this representation result, without the need for explicit use of fibrational machinery. This is done by weakening the notion of coalgebra homomorphism so that the usual diagrams only commute up to isomorphism. This weakening provides the additional flexibility we require in order to identify the unitary transformations as automorphisms of a suitable "quantum" coalgebra.

1.3 Fibred Coalgebraic Logic

By viewing coalgebras as generalized Kripke frames, the field of coalgebraic logic, initiated in [Moss, 1999], investigates modal logics with coalgebraic models. In chapter 4 we provide standard background on coalgebraic logic. As an example application we develop two expressive modal logics for quantum computation at the end of that chapter.

The paper [Abramsky, 2010], discussed in section 1.2, introduced a fibration of coalgebras over different signature functors, and raised the question of what a suitable "fibred coalgebraic logic" might look like. This question provides the starting point for the work in chapter 5. The central idea is that suitable functors:

$$F : T\text{-}\mathbf{Coalg} \to T'\text{-}\mathbf{Coalg} \tag{1.1}$$

should induce new modalities relating formulae for logics for the two different signature functors. After exploring several logically motivated examples, we introduce the notion of a *model transformation*, and show how these induce well behaved modalities relating to changes of the underlying coalgebraic model.

We then consider in detail the special case in which coalgebraic signature endofunctors on a category $\mathcal{X}$ are parameterized by a category $\mathcal{I}$ in the form of a bifunctor:

$$T : \mathcal{I} \times \mathcal{X} \to \mathcal{X} \tag{1.2}$$

We introduce corresponding parameterized logical signatures, described by a bifunctor:

$$L : \mathcal{I} \times \mathcal{A} \to \mathcal{A} \tag{1.3}$$

It is shown that compatibility between these two functors can be described by a parameterized form of distributive law, generalizing existing work in the single signature setting. It is then shown that under common assumptions about how the semantics of our logic is described, covariant model transformations can equivalently be replaced with contravariant formula transformations.

Fibrations have been used to investigate different aspects of coalgebraic logic, as discussed in the next section and chapter 6. A different notion of coalgebraic logic in a fibred setting can be found in [Klin, 2004].

1.4 Fibrations of Predicates

A second key logical aspect of coalgebra theory is the consideration of invariants [Jacobs, 2002], subsets of states closed under the transition structure. These invariants have a natural fibred structure, and this structure can be seen as being induced by lifting the coalgebraic signature functor to a category of predicates.

In chapter 6 we examine the impact of parameterization upon the mathematical structure of invariants. As parameterization leads to fibrations of categories of coalgebras, and each category of coalgebras leads to a fibration of invariants, we may expect to find parameterization leads to a “fibration of fibrations”. This is made technically precise and proven to be the case under very general assumptions in chapter 6.

1.5 Preliminaries and Notation

We assume a reasonable level of familiarity with elementary category theory, particularly the key notions of categories, functors, natural transformations, adjunctions and monads. The standard reference [MacLane, 1998] is more than sufficient for our purposes. Further background material will be provided in each of chapters 2, 3, 4, 5 and 6, addressing the specific technical theme of that chapter.

In order to fix notation, we quickly define some common categories that will be required in later sections:

- **Set** is the category of sets and functions between them
- **Stone** is the category of Stone spaces and continuous functions
- **Bool** is the category of Boolean algebras and their homomorphisms
- **MSLat** is the category of meet semilattices and their homomorphisms

Specific category names will be written in bold font, arbitrary categories will be written in calligraphic font, for example $\mathcal{C}$. Occasionally we will need to refer to 2-categories, in particular we write $\mathfrak{Cat}$ for the 2-category of (small) categories, functors and natural transformations, other 2-categories will be written similarly as they are introduced. For categories $\mathcal{C}$ and $\mathcal{D}$ we will write $[\mathcal{C}, \mathcal{D}]$ for the corresponding **functor category**.

We will also have need to refer to the following functors:

- $\mathcal{P}(-) : \mathbf{Set} \to \mathbf{Set}$ is the **covariant powerset functor**
- $2^{(-)} : \mathbf{Set}^{op} \to \mathbf{Bool}$ is the **contravariant powerset functor**
- $\mathcal{C}(-,-) : \mathcal{C}^{op} \times \mathcal{C} \to \mathbf{Set}$ is the **homset bifunctor** for a category $\mathcal{C}$

We will use the following notation throughout:

$A := B$	A is definitionally equal to B
$A = B$	A is propositionally equal to B
$A \cong B$	A is isomorphic to B
$A \simeq B$	A is equivalent to B

As we consider dualities in chapter 5, the following standard notation will be useful, for 2-category $\mathfrak{C}$:

$\mathfrak{C}^{op}$ is the 2-category with the 1-cells of $\mathfrak{C}$ reversed
$\mathfrak{C}^{co}$ is the 2-category with the 2-cells of $\mathfrak{C}$ reversed
$\mathfrak{C}^{coop}$ is the 2-category with both the 1-cells and the 2-cells of $\mathfrak{C}$ reversed

1.6 Relationship to Published Work

There are two papers under development based on the work in chapter 2. [Marsden, 2014] is an extended version of the material in that chapter, and [Hinze and Marsden, 2014] is an introductory account of string diagrammatic techniques aimed at the functional programming community, coauthored with Professor Ralf Hinze.

Chapter 3 is based on material published in the first half of [Marsden, 2013a], with the expressive logics for quantum systems of chapter 4 coming both from the later part of that paper, and also [Marsden, 2013b].

The paper [Marsden, 2013b] also contains very early ideas on which chapter 5 is based, although the material has changed a great deal since that publication. In particular model transformations, the material explicitly using fibrational machinery and the introduction of parameterized logical connections is new and has not appeared before in print.

The material on generalized fibrations of predicates in chapter 6 is entirely new, and has not been published in any form before.

Chapter 2

String Diagrams for Elementary Category Theory

2.1 Introduction

This chapter develops in some detail many aspects of basic category theory, with the aim of demonstrating the combined effectiveness of two key concepts:

1. The calculational reasoning approach to mathematics.
2. The use of string diagrams in category theory.

To illustrate the difference between traditional "diagram pasting", the calculational approach using symbolic equations as described in for example [Fokkinga, 1992a], and the string diagrammatic calculus, we will investigate a simple example.

Example 2.1.1 (Algebra Homomorphisms)**.** Let $\mathcal{C}$ be a category, and $T : \mathcal{C} \to \mathcal{C}$ be an endofunctor on $\mathcal{C}$. A T**-algebra** is a pair consisting of an object X of $\mathcal{C}$ and a $\mathcal{C}$-morphism $a : TX \to X$. The functor T is referred to as the **signature functor**. We leave the underlying object X implicit, and write T-algebras vertically as follows $\begin{pmatrix} TX \\ \downarrow a \\ X \end{pmatrix}$. A T**-algebra homomorphism** of type $\begin{pmatrix} TX \\ \downarrow a \\ X \end{pmatrix} \to \begin{pmatrix} TY \\ \downarrow a' \\ Y \end{pmatrix}$ is a $\mathcal{C}$-morphism $h : X \to Y$ such that the following diagram commutes:

$$\begin{tikzcd}
T(X) \arrow[r, "T(h)"] \arrow[d, "a"'] & T(Y) \arrow[d, "a'"] \\
X \arrow[r, "h"'] & Y
\end{tikzcd} \tag{2.1}$$

T-algebras and their homomorphisms form a category T-**Alg**. We will now show, using the three different proof styles discussed above, that for a category $\mathcal{D}$, a functor $F : \mathcal{C} \to \mathcal{D}$, an endofunctor $T' : \mathcal{D} \to \mathcal{D}$ and natural transformation $\alpha : T' \circ F \Rightarrow F \circ T$, if $h : \begin{pmatrix} TX \\ \downarrow a \\ X \end{pmatrix} \to \begin{pmatrix} TY \\ \downarrow a' \\ Y \end{pmatrix}$ is a T-algebra homomorphism then Fh is a T'-algebra homomorphism of type $\begin{pmatrix} T'FX \\ \downarrow Ta \circ \alpha_X \\ FX \end{pmatrix} \to \begin{pmatrix} T'FY \\ \downarrow Fa' \circ \alpha_Y \\ FY \end{pmatrix}$.

Firstly we will prove this in the traditional diagram pasting manner. We note that, by the naturality of α, the following diagram commutes:

$$\begin{tikzcd}
T'F(X) \arrow[r, "T'F(h)"] \arrow[d, "\alpha_X"'] & T'F(Y) \arrow[d, "\alpha_Y"] \\
FT(X) \arrow[r, "FT(h)"'] & FT(Y)
\end{tikzcd} \tag{2.2}$$

Also, as functor application preserves commuting diagrams, the following also commutes:

$$\begin{tikzcd}
FT(X) \arrow[r, "FT(h)"] \arrow[d, "F(a)"'] & FT(Y) \arrow[d, "F(a')"] \\
F(X) \arrow[r, "F(h)"'] & F(Y)
\end{tikzcd} \tag{2.3}$$

We can then paste diagrams (2.2) and (2.3) together, giving:

$$\begin{tikzcd}
T'F(X) \arrow[r, "T'F(h)"] \arrow[d, "\alpha_X"'] & T'F(Y) \arrow[d, "\alpha_Y"] \\
FT(X) \arrow[r, "FT(h)"] \arrow[d, "F(a)"'] & FT(Y) \arrow[d, "F(a')"] \\
F(X) \arrow[r, "F(h)"'] & F(Y)
\end{tikzcd} \tag{2.4}$$

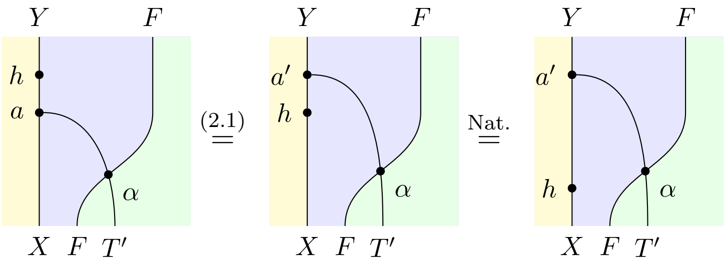
The claim then follows from the outer rectangle of diagram (2.4).

A proof of the same claim in a calculational style similar to [Fokkinga, 1992b] might proceed as follows:

$$\begin{aligned}
& F(h) \circ F(a) \circ \alpha_X \\
= \quad & \{ \text{ functoriality } \} \\
& F(h \circ a) \circ \alpha_X \\
= \quad & \{ \text{ } h \text{ is a } T\text{-algebra homomorphism } \} \\
& F(a' \circ T(h)) \circ \alpha_X \\
= \quad & \{ \text{ functoriality } \} \\
& F(a') \circ FT(h) \circ \alpha_X \\
= \quad & \{ \text{ naturality } \} \\
& F(a') \circ \alpha_Y \circ T'F(h)
\end{aligned}$$

and this proves the claim.

Finally, we will sketch how such a proof will look using string diagrams, deliberately mirroring the previous calculational style of proof, but now encoding the mathematical data graphically. The precise details of this graphical approach will be developed in later sections, but hopefully the example is sufficiently straightforward for the underlying ideas to be apparent:



(2.5)

Note the topological nature of the proof, the first step allows us to slide h through a as h is a homomorphism, the second equality follows as naturality manifests itself as sliding two morphisms past each other along the “wires” of the diagram.

Now we highlight some similarities and differences between the three styles of reasoning:

- Using diagram pasting, a proof may often be presented as a single large and potentially complex commuting diagram from which the required equalities can be read off. Such a diagram efficiently encodes a large number of equalities, and all the associated type information. Unfortunately, this style of presentation does not capture how the proof was developed and it can be unclear in what order the reasoning proceeded. Constructing such a diagram is also somewhat of an art form.

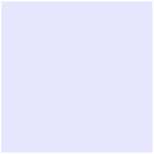
- The calculational approach presents a proof as a simple series of equalities and manipulations of equations. The sequence of steps and their motivation can be made very explicit, aiding both the development and later understanding of such a proof. The notation is also clearly more compact than the commuting diagrams, but at a significant cost as all the type information is lost in the conversion to equational form. This type information can help eliminate errors in reasoning, and often proofs in category theory are guided by the need to compose morphisms in a type correct manner.
- The string diagrammatic proof combines some of the benefits of both of the previous approaches. The presentation is clearly more verbose than the traditional symbolic equations, but this additional verbosity allows us to retain the valuable type information. Proofs consist of equalities and manipulations of equations in a calculational style, with the types helping to guide and constrain the steps that can be performed. Also, using string diagrams, some aspects of proofs are silently "handled by the notation". For example, the equational calculational proof above must explicitly invoke functoriality but in the graphical proof functoriality is implicit in the notation. Tool support could significantly reduce the cost of manipulating these more complex objects when developing a proof. This is done for a similar diagrammatic calculus, used in quantum computing, by the Quantomatic tool [Kissinger et al., 2014].

The remainder of this chapter aims to provide a self contained account of the use of string diagrams to reason calculationally about elementary category theory. Our aim is to demonstrate the breadth of applications rather than to analyze theoretical properties of string diagrammatic calculi, and so we develop a large number of examples, providing many diagrammatic proofs. We will assume some basic familiarity with categorical concepts such as categories, functors and natural transformations, and the elementary details of adjunctions, limits and colimits.

As our aim to describe "ordinary" category theory, we will work exclusively with the 2-category of all categories, rather than working axiomatically with bicategories carrying sufficient structure. Many results are folklore or well documented in the literature, although our heavy emphasis on providing a graphical formulation of these results suitable for calculation is certainly non-standard. In section 2.5 we provide a new perspective on representable functors as a source of graphical calculation rules. This perspective is then used to provide graphical rules for Kan extensions, limits and colimits in later sections. The calculational tools in this section will be used in later sections of the thesis to demonstrate the suitability of these techniques in the fields of coalgebra and coalgebraic logic.

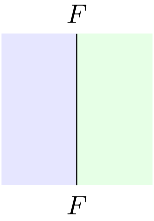
2.2 An Outline of the Graphical Approach

In the string diagram formalism, a category $\mathcal{C}$ is represented as a coloured region:



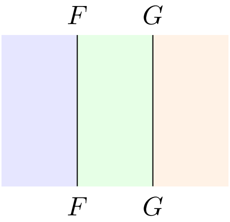
(2.6)

Different colours are used to denote different categories within a diagram. To avoid a great deal of repetitive detail, in what follows we will not in general specify which categories correspond to which colours. This information, if needed, should be apparent from the other visual components within our diagrams. A functor $F : \mathcal{C} \to \mathcal{D}$ is represented as an edge, commonly referred to as a wire, for example:



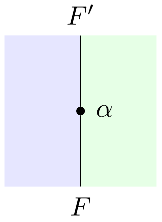
(2.7)

Identity functors are often omitted and simply drawn as a region. Given a second functor $G : \mathcal{D} \to \mathcal{E}$, the composite $G \circ F$ is represented as:



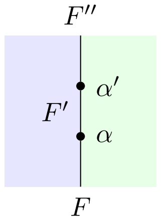
(2.8)

So *functors compose from left to right.* Note how by representing identity functors as regions, composing on the left or right with an identity functor "does nothing" to our diagram as we would expect. A natural transformation $\alpha : F \Rightarrow F'$ is represented as:



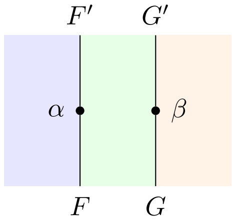
(2.9)

Given a second natural transformation $\alpha' : F' \Rightarrow F''$, the vertical composite $\alpha' \circ \alpha$ is written as follows:



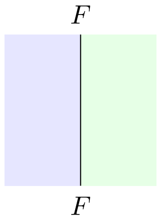
(2.10)

So *natural transformations compose vertically from bottom to top.* Now if we also have a natural transformation $\beta : G \Rightarrow G'$ we represent the horizontal composite $\beta * \alpha$ as:



(2.11)

So *natural transformations compose horizontally from left to right.* Identity natural transformations are omitted, so $1 : F \Rightarrow F$ is drawn as:



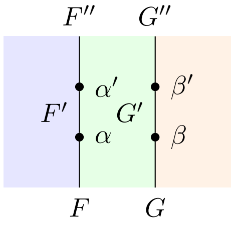
(2.12)

In this way, vertical composition with identity natural transformations "does nothing" to our diagram as we would expect. Given another natural transformation $\beta' : G' \Rightarrow$

G'', the **interchange law** holds:

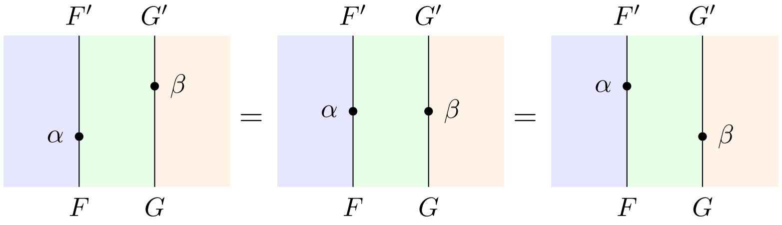
$$(\beta' \circ \beta) * (\alpha' \circ \alpha) = (\beta' * \alpha') \circ (\beta * \alpha) \tag{2.13}$$

and so the following composite is well defined, corresponding to a unique natural transformation:



(2.14)

By combining the rules for identities and the interchange law appropriately, we have the following **sliding equalities**:



(2.15)

For a similar graphical calculus in [Dubuc and Szyld, 2013], this is memorably summarized as:

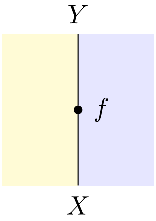
> This allows us to move cells up and down when there are no obstacles, as if they were elevators.

Much of what can be proved with the string diagram calculus flows from the fact we can slide natural transformations past each other like this, as we saw in example 2.1.1 in the introduction.

2.2.1 Some Elementary Techniques

Objects and Morphisms

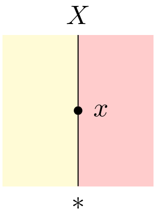
Often we will want to reason using categories, functors, natural transformations and "ordinary" objects and morphisms within categories, as we did in example 2.1.1. To achieve this, we consider the category 1 with one object and one (identity) morphism. Functors of type $1 \to \mathcal{C}$ can be identified with objects of the category $\mathcal{C}$. A morphism $f : X \to Y$ is then a natural transformation between two of these functors from the terminal category. If we identify the functors and the corresponding objects, then we can write f in the obvious way as:



(2.16)

Elements of Sets

When we work in the category **Set**, given a set X, we may wish to refer to an element $x \in X$. Extending the idea in section 2.2.1, such an element can be identified with a morphism $x : * \to X$, where $*$ is the one element set (the terminal object in **Set**). We then write this element as:



(2.17)

Naturality

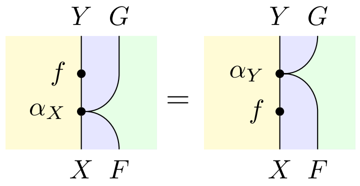
Often, given a family of morphisms:

$$(\alpha_X : F(X) \to G(X))_{X \in \mathsf{obj}(\mathcal{C})} \tag{2.18}$$

in some category $\mathcal{C}$, we wish to show that this family constitutes a natural transformation. In terms of commuting diagrams, this requires that for all $\mathcal{C}$ objects X, Y and morphisms $f : X \to Y$, the following diagram commutes:

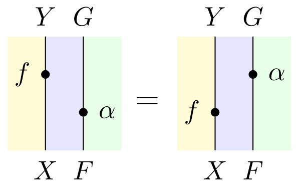
$$\begin{tikzcd}
FX \arrow[r, "\alpha_X"] \arrow[d, "Ff"'] & GX \arrow[d, "Gf"] \\
FY \arrow[r, "\alpha_Y"'] & GY
\end{tikzcd} \tag{2.19}$$

Rephrasing this condition in terms of string diagrams, we must equivalently show that for all $\mathcal{C}$ objects X, Y and morphisms $f : X \to Y$ that the following equation holds:



(2.20)

Notice the similarity between the above condition and the sliding equations. The "pinching" effect in the string diagrams of equation (2.20) reflects the fact we are reasoning about a natural transformation componentwise rather than as a totality. The naturality condition says that the components of the natural transformation and the morphism f "slide past each other". Once naturality is established, we can draw α and f on two parallel wires to emphasize this independence, as follows:



(2.21)

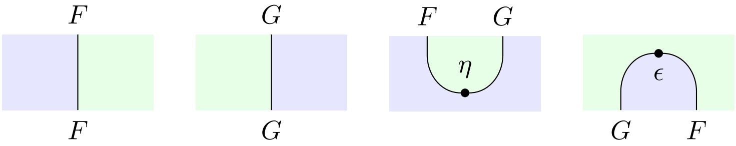
2.3 Adjunctions

We will now investigate some of the properties of one of the central concepts of category theory, adjunctions. As our focus is calculational proofs we will not discuss concrete examples of adjunctions in the text, instead, we will formulate some of the key concepts graphically, and then illustrate the use of graphical techniques in the proof of some standard results.

2.3.1 Calculational Properties of Adjunctions

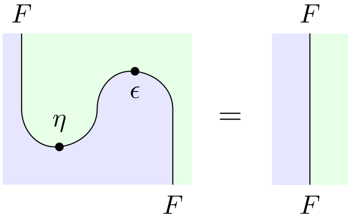
Adjunctions can be described in a myriad of different and equivalent ways. In this section we adopt an approach that particularly suits our diagrammatic presentation, the subsequent examples will allow us to recover many of the other aspects of adjunctions. A comprehensive discussion of the different formulations of the concept of an adjunction, using an algebraic calculational style, is given in [Fokkinga and Meertens, 1994].

Definition 2.3.1 (Adjunction)**.** An **adjunction** consists of functors and natural transformations:

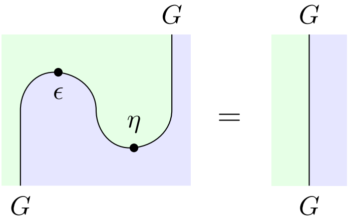


(2.22)

satisfying the following **snake equations**:



(2.23a)



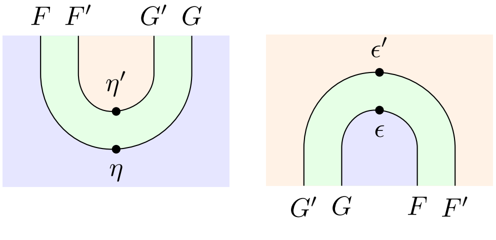
(2.23b)

In this case we say that F is a **left adjoint** for G, or G is a **right adjoint** for F, written $F \dashv G$. The natural transformations η and ϵ are referred to as the **unit** and **counit** of the adjunction.

Lemma 2.3.2 (Adjunctions Compose)**.** *Two adjunctions $F : \mathcal{C} \to \mathcal{D} \dashv G : \mathcal{D} \to \mathcal{C}$ and $F' : \mathcal{D} \to \mathcal{E} \dashv G' : \mathcal{E} \to \mathcal{D}$ compose to give an adjunction:*

$$F' \circ F \dashv G \circ G' \tag{2.24}$$

Proof. Take as unit and counit the composites:



(2.25)

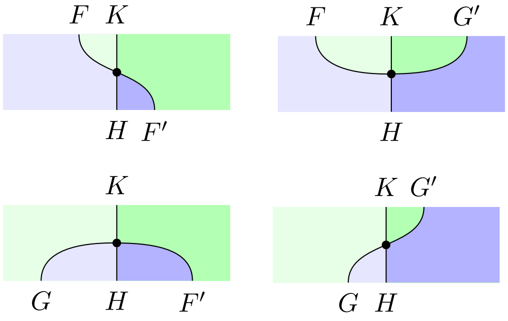
That these satisfy the adjunction axioms is then trivial to show. □

We now consider in some detail how the units and counits of a pair of adjunctions relate to certain natural transformations. A lot of structure can be derived from the fact that units and counits let us “bend wires”.

Definition 2.3.3 (Wire Bending)**.** We consider a situation with adjunctions $F \dashv G$ and $F' \dashv G'$ and functors H, K, with types as in the following (not necessarily commuting) diagram:

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "H"] \arrow[d, bend right=60, "F"'] \arrow[d, phantom, "\dashv"] & \mathcal{C}' \arrow[d, bend right=60, "F'"'] \arrow[d, phantom, "\dashv"] \\
\mathcal{D} \arrow[r, "K"] \arrow[u, bend right=60, "G"'] & \mathcal{D}' \arrow[u, bend right=60, "G'"']
\end{tikzcd} \tag{2.26}$$

There are then bijections between the sets of natural transformations of the four types shown below:

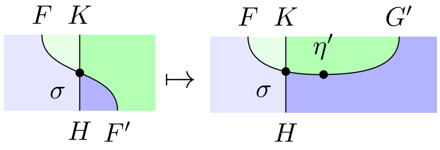


(2.27)

Graphically, these bijections are induced by "bending wires" using the adjunction axioms to move between the different types.

Starting in the top left hand corner, between each pair of types we define a function $(-)^{\triangleright}$, referred to as **move right**, in the clockwise direction, with an inverse $(-)^{\triangleleft}$, referred to as **move left**, in the counter clockwise direction. Hopefully this overloading of names will not cause confusion, given the abundance of type information in the string diagrams we are using. Firstly:

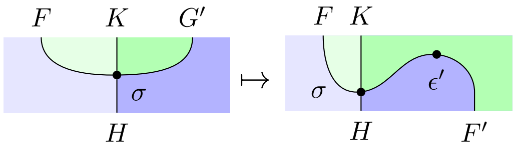
$$(-)^{\triangleright} : [\mathcal{C}, \mathcal{D}'](F'H, KF) \to [\mathcal{C}, \mathcal{C}'](H, G'KF)$$



(2.28)

with inverse:

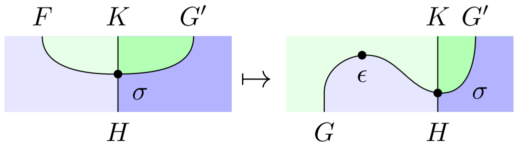
$$(-)^{\triangleleft} : [\mathcal{C},\mathcal{C}'](H,G'KF) \to [\mathcal{C},\mathcal{D}'](F'H,KF)$$



(2.29)

Secondly:

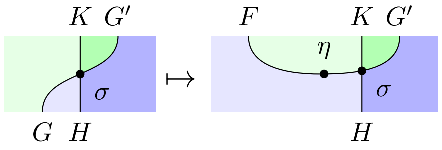
$$(-)^{\triangleright} : [\mathcal{C},\mathcal{C}'](H,G'KF) \to [\mathcal{D},\mathcal{C}'](HG,G'K)$$



(2.30)

with inverse:

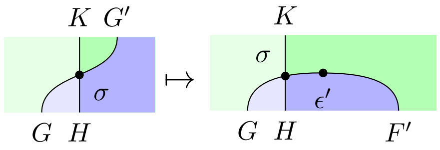
$$(-)^{\triangleleft} : [\mathcal{D},\mathcal{C}'](HG,G'K) \to [\mathcal{C},\mathcal{C}'](H,G'KF)$$



(2.31)

Thirdly:

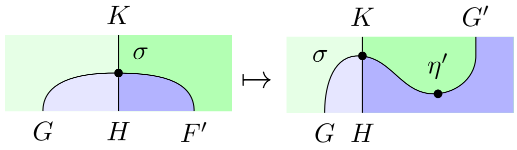
$$(-)^{\triangleright} : [\mathcal{D},\mathcal{C}'](HG,G'K) \to [\mathcal{D},\mathcal{D}'](F'HG,K)$$



(2.32)

with inverse:

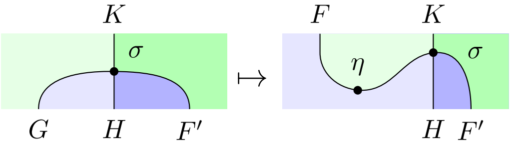
$$(-)^{\triangleleft} : [\mathcal{D},\mathcal{D}'](F'HG,K) \to [\mathcal{D},\mathcal{C}'](HG,G'K)$$



(2.33)

Finally:

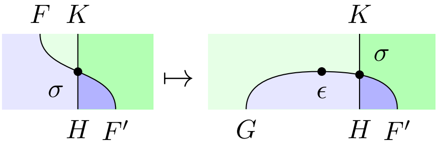
$$(-)^{\triangleright} : [\mathcal{D}, \mathcal{D}'](F'HG, K) \to [\mathcal{C}, \mathcal{D}'](F'H, KF)$$



(2.34)

with inverse:

$$(-)^{\triangleleft} : [\mathcal{C}, \mathcal{D}'](F'H, KF) \to [\mathcal{D}, \mathcal{D}'](F'HG, K)$$



(2.35)

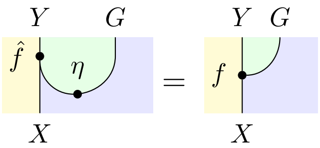
That each of these pairs of maps witness a bijection is easy to check by applying the adjunction axioms. In fact, each of these families of maps is natural in both H and K. This is easy, if lengthy, to check graphically, and omitted to conserve space.

Lemma 2.3.4. *Each of the composites* $(-)^{\triangleleft\triangleleft\triangleleft\triangleleft}$ *and* $(-)^{\triangleright\triangleright\triangleright\triangleright}$ *as described in definition 2.3.3 are equal to the identity.*

Proof. Straightforward from expanding definitions and exploiting the adjunction axioms. ☐

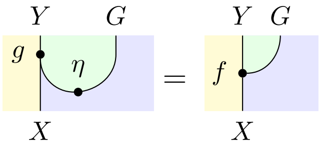
We will now exploit these wire bending ideas to provide a graphical proof of a standard result about adjunctions.

Lemma 2.3.5. *For every morphism* $f : X \to G(Y)$ *there exists a unique morphism* $\hat{f} : F(X) \to Y$ *such that:*

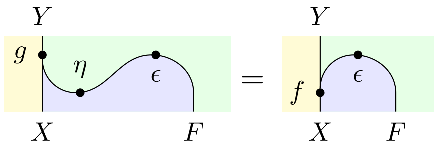


(2.36)

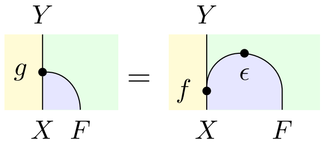
Proof. Both existence and uniqueness follow from the following equivalences:



$\Leftrightarrow$ { wire bending is a bijection }



$\Leftrightarrow$ { adjunction axiom (2.23a) }

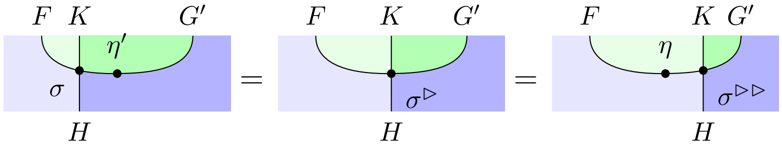


□

Remark 2.3.6. Although we proved lemma 2.3.5 directly, we could simply have observed that equation (2.36) is a special case of the first bijection in definition 2.3.3, with the leftmost adjunction in the diagram being the trivial adjunction between the identity functors.

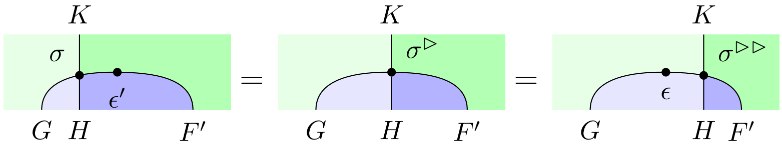
We now see why the suggestive names "move left" and "move right" were chosen for the mapping in definition 2.3.3. They allow us to "slide" a natural transformation back and forth along the bends given by the units and counits of adjunctions, as described by the next lemma.

Lemma 2.3.7 (Adjunction Sliding). *We have the following equalities allowing us to "slide" a natural transformation along a pair of adjunctions:*



(2.37)

and

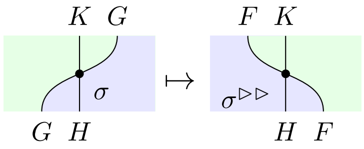


(2.38)

Proof. The equalities either follow immediately from the definitions, or require a single application of one of the adjunction axioms. □

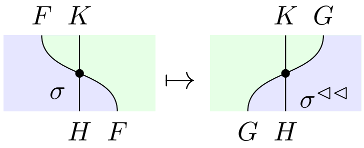
Remark 2.3.8. Combining lemma 2.3.7 with the fact that the various pairs of functions $(-)^{\triangleleft}$ and $(-)^{\triangleright}$ witness a bijection will allow us to derive several similar sliding identities, involving various combinations of both $(-)^{\triangleleft}$ and $(-)^{\triangleright}$.

Definition 2.3.9 (Mates Under an Adjunction)**.** Given an adjunction $F : \mathcal{C} \to \mathcal{D} \dashv G : \mathcal{D} \to \mathcal{C}$ and endofunctors $H : \mathcal{C} \to \mathcal{C}, K : \mathcal{D} \to \mathcal{D}$, there is a bijection between the natural transformations of types $HG \Rightarrow GK$ and $FH \Rightarrow KF$. This is easily seen using the operations in definition 2.3.3, in the special case where both adjunctions are the same, with:



(2.39)

and



(2.40)

Natural transformations related by these bijections are referred to as **mates under the adjunction** $F \dashv G$ [Kelly and Street, 1974].

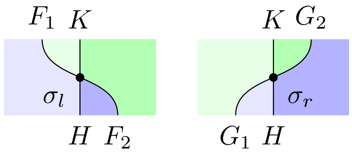
2.3.2 Adjoint Squares

We will now apply our graphical approach to adjunctions in an investigation of some properties of adjoint squares.

Definition 2.3.10 (Adjoint Square)**.** An **adjoint square** σ is a pair of adjunctions and two functors H, K arranged as in the following (not necessarily commuting) diagram:

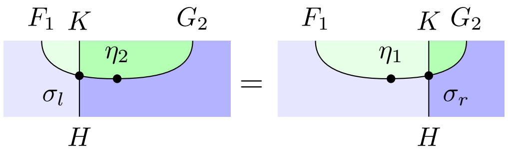
$$\begin{tikzcd}
\mathcal{C}_1 \arrow[rr, "H"] \arrow[d, bend right=60, "F_1"'] \arrow[d, phantom, "\dashv"] & & \mathcal{C}_2 \arrow[d, bend right=60, "F_2"'] \arrow[d, phantom, "\dashv"] \\
\mathcal{D}_1 \arrow[rr, "K"] \arrow[u, bend right=60, "G_1"'] & & \mathcal{D}_2 \arrow[u, bend right=60, "G_2"']
\end{tikzcd} \tag{2.41}$$

and also natural transformations:

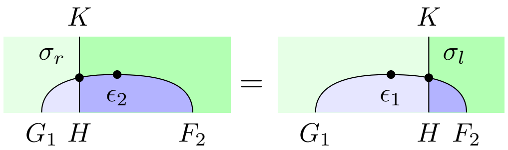


(2.42)

satisfying the following sliding equalities:



(2.43a)

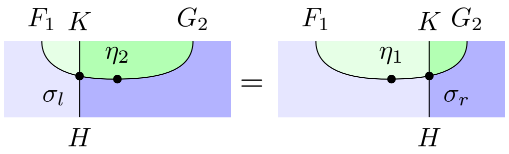


(2.43b)

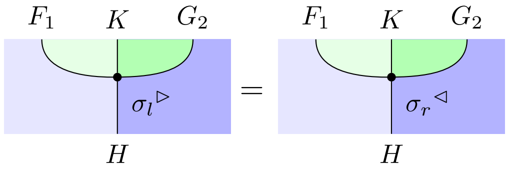
Such a pair of natural transformations are said to be **conjugate**. See [MacLane, 1998] exercise IV.7.4.

Lemma 2.3.11. *For notation as in definition 2.3.10, the conditions in equations* (2.43a) *and* (2.43b) *are equivalent. Furthermore,* σ_l *and* σ_r *determine each other uniquely.*

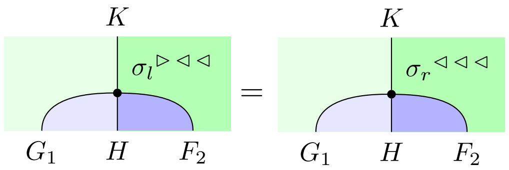
Proof. That the conditions are equivalent follows from:



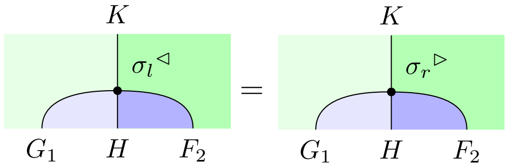
⇔ { definitions }



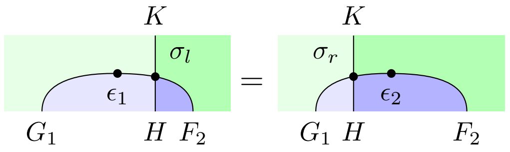
⇔ { applying the $(-)^{\lhd}$ wire bending bijection twice }



$\Leftrightarrow$ { $(-)^{\triangleleft}$ and $(-)^{\triangleright}$ are inverses, and $(-)^{\triangleleft\triangleleft\triangleleft\triangleleft}$ is the identity }



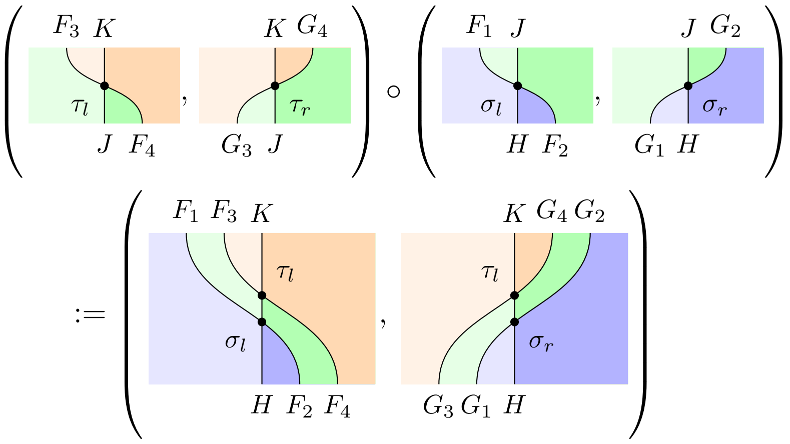
$\Leftrightarrow$ { definitions }



That σ_l and τ_r determine each other uniquely is then immediate as they are related by the bijections from definition 2.3.3. ☐

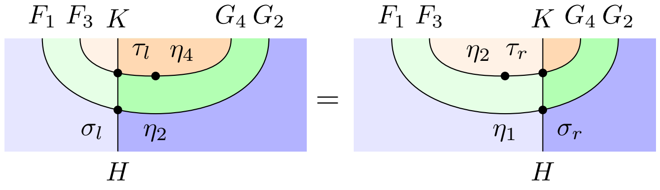
Proposition 2.3.12. *Adjoint squares compose vertically and horizontally.*

Proof. By lemma 2.3.11, we will only need to check one of the two equations (2.43a) and (2.43b) holds in order to ensure our composition operations preserve conjugate pairs. The vertical composition of adjunctions will be assumed to be as in lemma 2.3.2. We will denote the vertical composition operation as $\circ$, defined as follows:



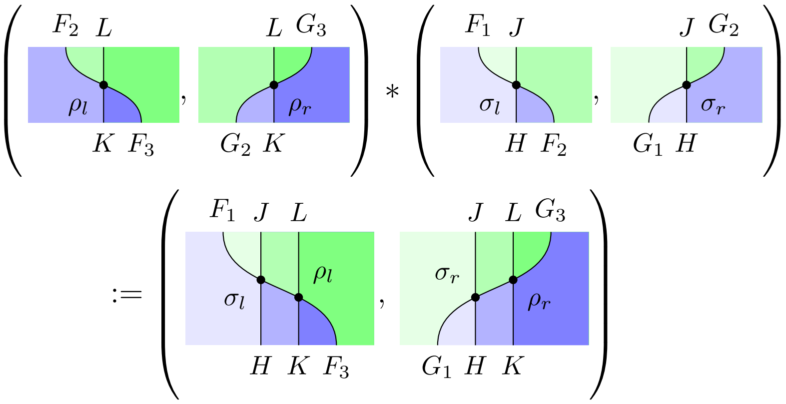
(2.44)

Confirming that the resulting pair of natural transformations are conjugate is trivial as the following equality holds because the component natural transformations are conjugate:



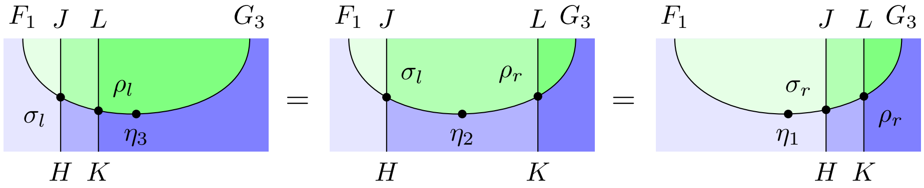
(2.45)

We will denote the horizontal composition operation $*$, defined as follows:



(2.46)

That the resulting pair of natural transformations are conjugate can be seen from the following equalities, given by applying the adjoint square equations (2.43a) and (2.43b) for both the component adjoint squares:



(2.47)

□

Remark 2.3.13. In proposition 2.3.12 and lemma 2.3.2 we see the usefulness of the explicit type information in the string diagram notation. The need to match colours correctly when forming the desired composites make the required constructions easy to identify.

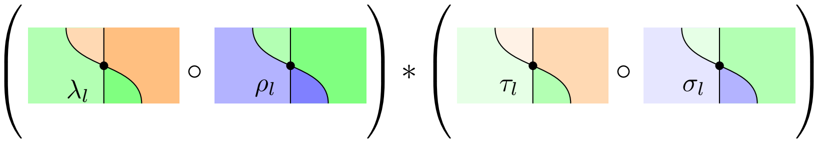
It is straightforward to check that the two forms of composition both give adjoint squares the structure of a category, with an appropriate choice of identities. We also have an interchange law that holds between these two types of composition.

Lemma 2.3.14 (Interchange Law). *With the composition operations defined in the proof of proposition 2.3.12, let $\lambda, \rho, \tau, \sigma$ be adjoint squares such that the required composites are well defined, then the following equality holds:*

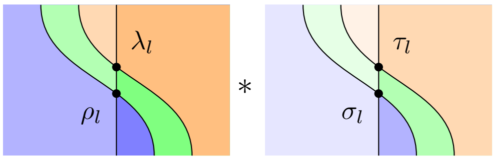
$$(\lambda \circ \rho) * (\tau \circ \sigma) = (\lambda * \tau) \circ (\rho * \sigma) \tag{2.48}$$

Proof. By proposition 2.3.12 the composition operations preserve conjugate pairs, and from lemma 2.3.11 the members of a conjugate pair determine each other uniquely.

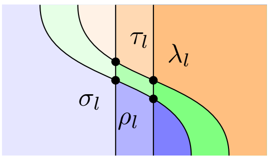
We therefore only need to consider one of the component natural transformations of the adjoint squares. We then have the following sequence of equalities directly from the definitions (with a slight abuse of notation we omit the second component of the adjoint squares):



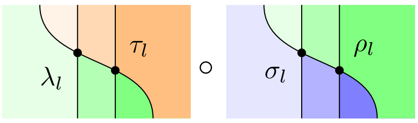
= { definition of vertical composition of adjoint squares }



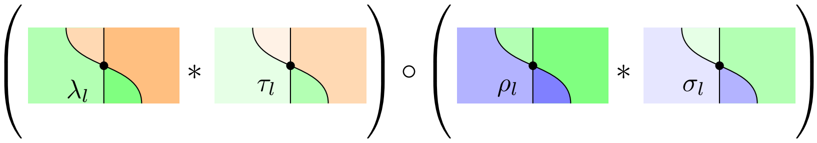
= { definition of horizontal composition of adjoint squares }



= { definition of vertical composition of adjoint squares }



= { definition of horizontal composition of adjoint squares }



□

Remark 2.3.15. In fact, the adjoint squares form a double category [Ehresmann, 1963], commonly used as a device for organizing information about adjunctions, see [Palmquist, 1971] and [Gray, 1974].

2.4 Monads

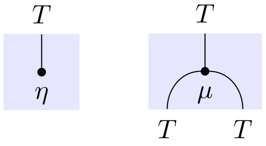
Monads, also historically referred to as triples and standard constructions, are another important notion in category theory. They are of particular interest to computer scientists, with important connections to the semantics of programming languages, see for example [Moggi, 1991] and [Wadler, 1995]. Monads are of course described in [MacLane, 1998], but there is much that is not covered. Standard sources for monad theory are [Barr and Wells, 2005] and [Manes, 1976], a more modern presentation is given in the book in preparation [Jacobs, 2012b]. We do not attempt to give a comprehensive treatment here. Rather, we aim to formulate some basic aspects of monads in graphical terms, and then provide examples of the string diagrammatic approach by proving some standard results.

The string diagram notation is particularly effective when working with monads, clarifying the presentation on many of the mathematical structures under consideration.

2.4.1 Calculational Properties of Monads

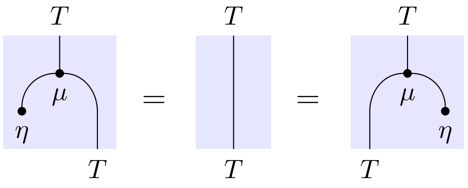
We now introduce graphical descriptions of monads and some associated mathematical structures.

Definition 2.4.1 (Monad)**.** Let $\mathcal{C}$ be some category. A **monad** on $\mathcal{C}$ consists of an endofunctor T on $\mathcal{C}$, a **unit** natural transformation η and **multiplication** natural transformation μ:



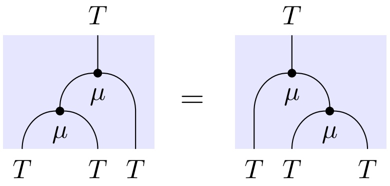
(2.49)

The following **unit axioms** are required to hold:



(2.50a)

Also μ is required to satisfy the following **associativity axiom**:

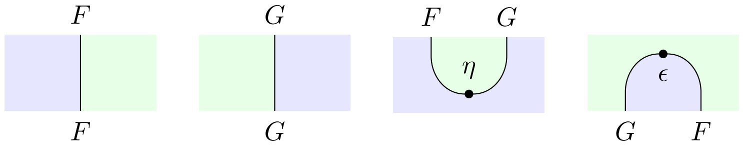


(2.50b)

The string diagram notation makes clear that a monad can be seen as a monoid in the functor category $[\mathcal{C}, \mathcal{C}]$, as suggested by the use of the names "unit" and "multiplication".

There is a close relationship between monads and adjunctions, in fact every adjunction induces a monad.

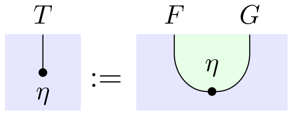
Lemma 2.4.2. *Every adjunction:*



(2.51)

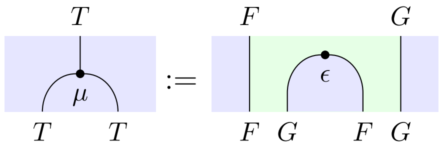
induces a monad.

Proof. Take the endofunctor as $T := G \circ F$, and the unit of the monad is also η:



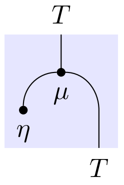
(2.52)

The multiplication μ of the monad is then defined as follows:

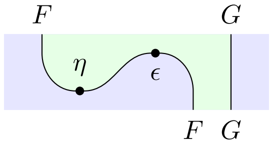


(2.53)

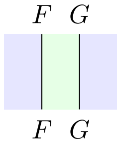
We must then check the required axioms hold. For the unit axioms we apply the snake equations:



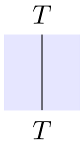
= { definitions (2.52) and (2.53) }



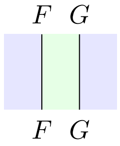
= { snake equation (2.23a) }



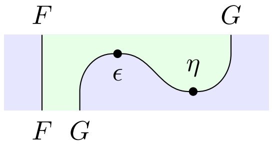
= { definition }



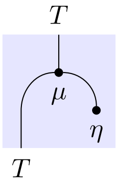
= { definition }



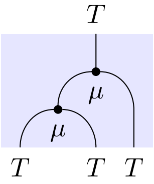
= { snake equation (2.23b) }



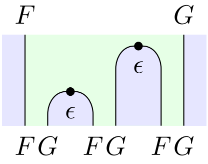
= { definitions (2.52) and (2.53) }



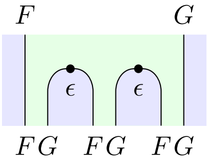
The associativity axiom follows directly from naturality:



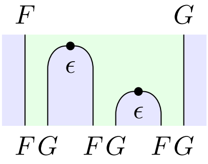
$= \{$ definition (2.53) $\}$



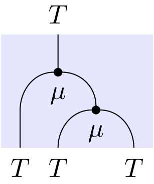
$= \{$ naturality $\}$



$= \{$ naturality $\}$



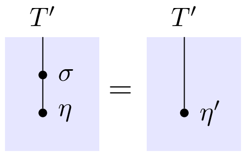
$= \{$ definition (2.53) $\}$



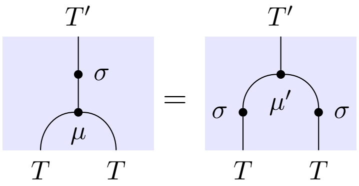
□

We now introduce a suitable notion of morphism for monads on the same category. We will generalize this situation in section 2.4.2.

Definition 2.4.3 (Monad Morphism)**.** For a fixed category $\mathcal{C}$, a **monad morphism** $(\mathcal{C}, T, \eta, \mu) \to (\mathcal{C}, T', \eta', \mu')$ is a natural transformation $\sigma : T \Rightarrow T'$ such that the following equalities hold:



(2.54a)

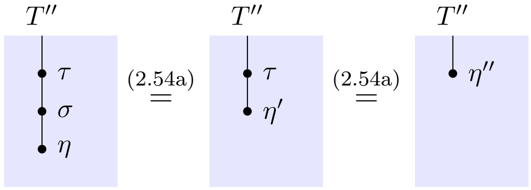


(2.54b)

Again the string notation is instructive. If we consider the two monads as monoids in $[\mathcal{C},\mathcal{C}]$, then σ is clearly a monoid homomorphism, commuting appropriately with the unit and multiplication.

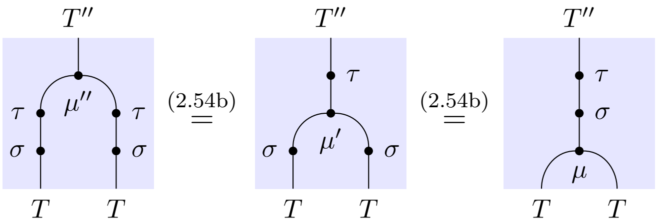
Lemma 2.4.4. *Let $(\mathcal{C}, T, \eta, \mu)$, $(\mathcal{C}, T', \eta', \mu')$ and $(\mathcal{C}, T'', \eta'', \mu'')$ be monads. Also let $\sigma : T \Rightarrow T'$ and $\tau : T' \Rightarrow T''$ be monad morphisms. Then $\tau \circ \sigma$ is a monad morphism. Also for any monad $(\mathcal{C}, T, \eta, \mu)$, the identity natural transformation on T is a monad morphism.*

Proof. For the unit we have:



(2.55)

Similarly for the multiplication:



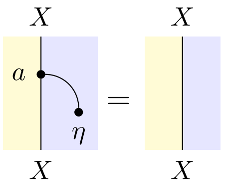
(2.56)

The second part is trivial. □

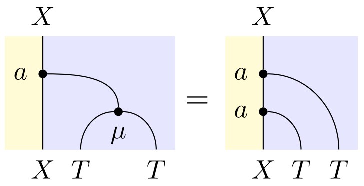
We could have seen the above lemma immediately as *monoid homomorphisms compose*. Clearly for a category $\mathcal{C}$, the monads on $\mathcal{C}$ and the monad morphisms form a category. We now investigate two important constructions for a given monad.

Definition 2.4.5 (Eilenberg-Moore Category)**.** The **Eilenberg-Moore category** for a monad $(\mathcal{C}, T, \eta, \mu)$, denoted $\mathcal{EM}(T)$, is defined as follows:

- **Objects**: Pairs consisting of an object X of $\mathcal{C}$ and a morphism $a : T(X) \to X$ satisfying the following equalities:



(2.57a)



(2.57b)

- **Morphisms**: A morphism $(X, a) \rightarrow (Y, b)$ is an algebra homomorphism, as defined in example 2.1.1. Composition of morphisms is as in $\mathcal{C}$.

The graphical notation shows that if we consider the monad as a monoid in $[\mathcal{C}, \mathcal{C}]$, we can then view the objects in the Eilenberg-Moore category as monoid actions on objects in $\mathcal{C}$. In fact, X is a constant functor, and we can generalize the above definition to arbitrary functors. We will not require this extra generality in our setting, details can be found in [Kelly and Street, 1974].

In lemma 2.4.2 we saw that every adjunction induces a monad. We now see that every monad arises in this way.

Proposition 2.4.6 (Eilenberg and Moore [1965])**.** *For a monad $(\mathcal{C}, T, \eta, \mu)$ there are functors:*

$$U : \mathcal{EM}(T) \rightarrow \mathcal{C} \quad \textit{and} \quad J : \mathcal{C} \rightarrow \mathcal{EM}(T) \tag{2.58}$$

with $J \dashv U$ and the monad induced as in lemma 2.4.2 is equal to T.

Proof. We define the obvious forgetful functor:

$$U : \mathcal{EM}(T) \rightarrow \mathcal{C} \tag{2.59}$$

$$(X, a : T(X) \rightarrow X) \mapsto X \tag{2.60}$$

$$h : X \rightarrow Y \mapsto h \tag{2.61}$$

and a putative left adjoint:

$$J : \mathcal{C} \rightarrow \mathcal{EM}(T) \tag{2.62}$$

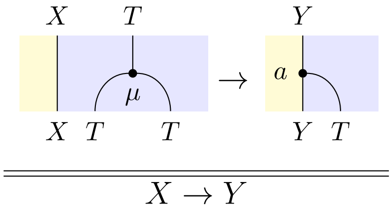
$$X \mapsto \mu_X \tag{2.63}$$

$$h : X \rightarrow Y \mapsto T(h) \tag{2.64}$$

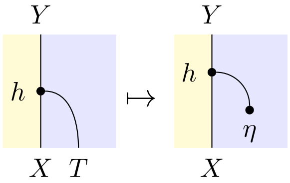
That these are valid functors is easy to check. Now we require a bijection:

$$\frac{J(X) \rightarrow (Y, a : T(Y) \rightarrow Y)}{X \rightarrow U(Y, a : T(Y) \rightarrow Y)}$$

Expanding definitions this becomes:

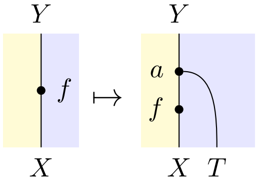


We define a map in the downward direction as follows:



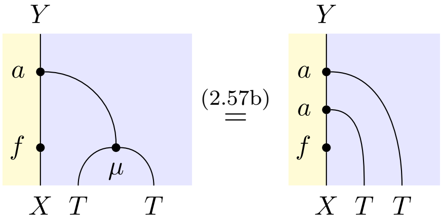
(2.65)

In the upward direction we define the map:



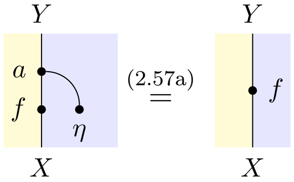
(2.66)

That this results in a valid algebra homomorphism follows immediately as a is an Eilenberg-Moore algebra and so:



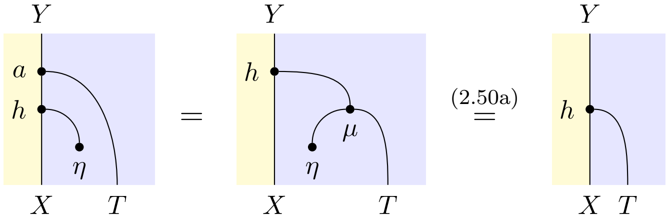
(2.67)

We now check these maps constitute a bijection. Firstly, as a is an Eilenberg-Moore algebra homomorphism we immediately have:



(2.68)

In the opposite direction:



(2.69)

The first equality above follows by the assumption h a homomorphism from μ_X to a.

We omit checking the naturality of these mappings to save space. That T is the monad induced by this adjunction follows from the construction in lemma 2.4.2. □

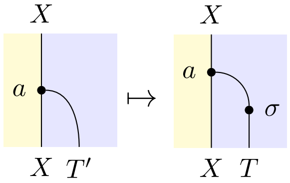
Proposition 2.4.7. *Every monad morphism:*

$$\sigma : (\mathcal{C}, T, \eta, \mu) \to (\mathcal{C}, T', \eta', \mu') \tag{2.70}$$

induces a functor:

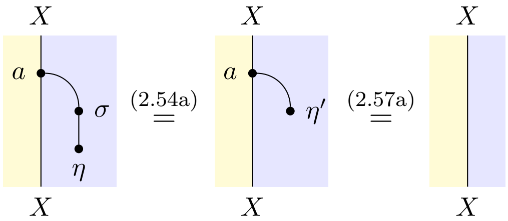
$$\mathcal{EM}(\sigma) : \mathcal{EM}(T') \to \mathcal{EM}(T) \tag{2.71}$$

Proof. On objects, a suitable functor acts as follows:



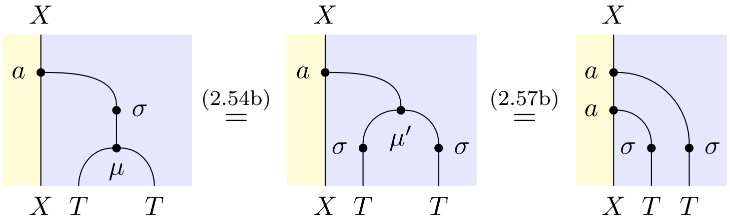
(2.72)

On morphisms, the functor acts as the identity. We must show the resulting algebra satisfies the unit and multiplication axioms. The unit axiom follows from the equalities:



(2.73)

That the multiplication axiom is satisfied is shown as follows:

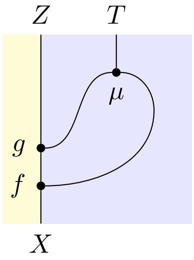


(2.74)

That this is functorial is easy to show by applying a special case of example 2.1.1. ☐

Definition 2.4.8 (Kleisli Category)**.** For a monad $(\mathcal{C}, T, \eta, \mu)$ the **Kleisli category** $\mathcal{K}\ell(T)$ is defined as follows:

- **Objects**: Objects of $\mathcal{C}$
- **Morphisms**: A morphism of type $f : X \to Y$ in $\mathcal{K}\ell(T)$ is a $\mathcal{C}$-morphism $f : X \to T(Y)$. Given another morphism $g : Y \to Z$ in $\mathcal{K}\ell(T)$, their (Kleisli) composite, written $g \bullet f$ is given by the following composite in $\mathcal{C}$:



(2.75)

The identity morphism on X in $\mathcal{K}\ell(T)$ is given by η_X, that this is a valid identity morphism follows immediately from the monad axioms.

We now see that the Kleisli category can be used to give an alternative construction of an adjunction inducing a particular monad, as was done with the Eilenberg-Moore category in proposition 2.4.6.

Proposition 2.4.9 (Kleisli [1965])**.** *For a monad $(\mathcal{C}, T, \eta, \mu)$ there are functors:*

$$V : \mathcal{K}\ell(T) \to \mathcal{C} \quad \textit{and} \quad H : \mathcal{C} \to \mathcal{K}\ell(T) \tag{2.76}$$

with $H \dashv V$ and the monad induced as in lemma 2.4.2 is equal to T.

Proof. We define the obvious forgetful functor:

$$V : \mathcal{K}\ell(T) \to \mathcal{C} \tag{2.77}$$

$$X \mapsto T(X) \tag{2.78}$$

$$f : X \to T(X) \mapsto \mu_Y \circ T(f) \tag{2.79}$$

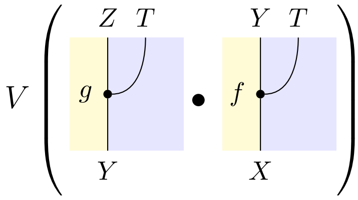
and a putative left adjoint:

$$H : \mathcal{C} \to \mathcal{K}\ell(T) \tag{2.80}$$

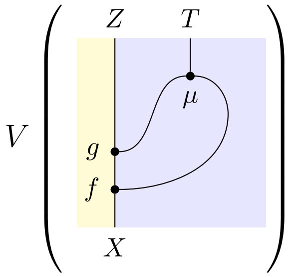
$$X \mapsto X \tag{2.81}$$

$$f : X \to Y \mapsto \eta_Y \circ f \tag{2.82}$$

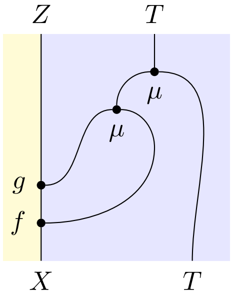
That V and H preserve identities is easy to check. For functoriality of composition for V, we have the following equalities:



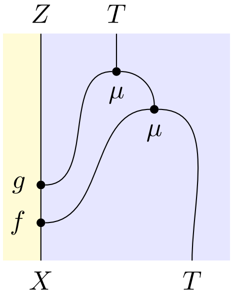
= { definition of Kleisli Composition }



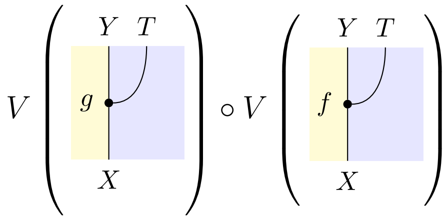
= { definition of V }



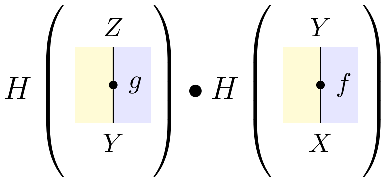
$=$ { monad associativity axiom (2.50b) }



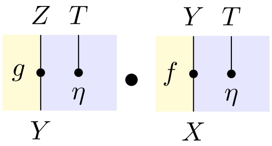
$=$ { definition of V }



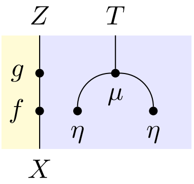
For functoriality of composition for H we have:



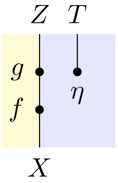
$=$ { definition of H }



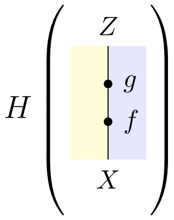
$=$ { definition of Kleisli composition }



$=$ { monad unit axiom (2.50a) }



$$= \quad \{ \text{ definition of } H \ \}$$



Now we require a bijection:

$$\frac{H(X) \to Y}{\overline{X \to V(Y)}}$$

Expanding definitions, and noting that a Kleisli morphism $X \to Y$ is a $\mathcal{C}$-morphism $X \to T(Y)$ by definition, the required bijection reduces to:

$$\frac{X \to T(Y)}{\overline{X \to T(Y)}}$$

Naturality of this bijection under the two forms of composition is easy to check. That we recover the original monad from this adjunction follows from the construction in lemma 2.4.2. ☐

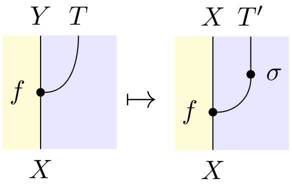
Proposition 2.4.10. *Every monad morphism:*

$$\sigma : (\mathcal{C}, T, \eta, \mu) \to (\mathcal{C}, T', \eta', \mu') \tag{2.83}$$

induces a functor:

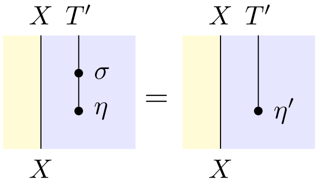
$$\mathcal{K}\ell(\sigma) : \mathcal{K}\ell(T) \to \mathcal{K}\ell(T') \tag{2.84}$$

Proof. On objects, the functor acts as the identity. The action on morphisms is defined in $\mathcal{C}$ as follows:



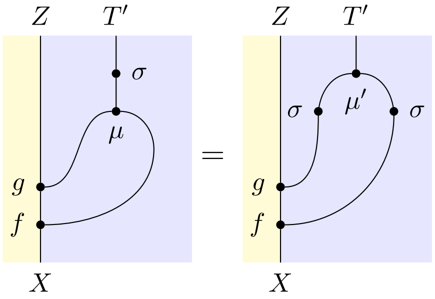
(2.85)

That this preserves identities is immediate from equation (2.54a) as:



(2.86)

Similarly, composition follows from equation (2.54b) as:



(2.87)

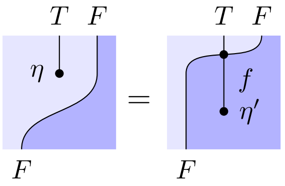
□

2.4.2 Categories of Monads

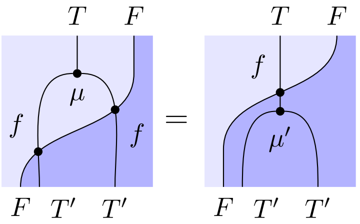
We now consider some appropriate types of morphisms between monads on different base categories.

Definition 2.4.11. The category **Monad** is defined as having:

- **Objects**: Monads
- **Morphisms**: A morphism of type $(\mathcal{C}, T, \eta, \mu) \to (\mathcal{C}', T', \eta', \mu')$ is a pair consisting of a functor $F : \mathcal{C} \to \mathcal{C}'$ and a natural transformation $f : T'F \Rightarrow FT$ satisfying the following equations:



(2.88a)



(2.88b)

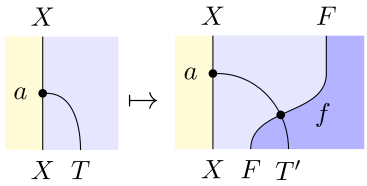
Proposition 2.4.12. *Every* **Monad**-*morphism:*

$$(\mathcal{C}, T, \eta, \mu) \to (\mathcal{C}', T', \eta', \mu') \tag{2.89}$$

induces a functor:

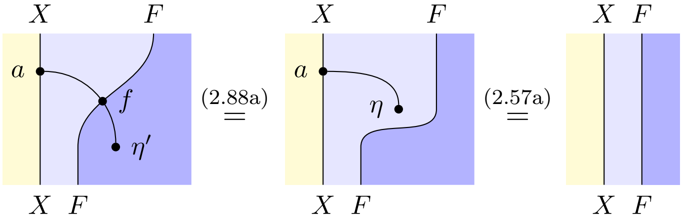
$$\mathcal{EM}(T) \to \mathcal{EM}(T') \tag{2.90}$$

Proof. For **Monad**-morphism (F, f) the action on objects of our functor is given by:



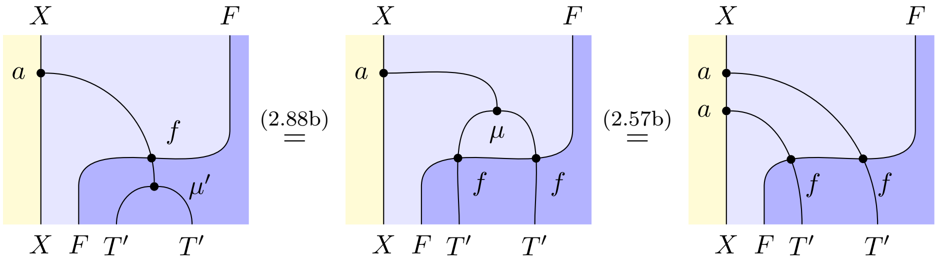
(2.91)

That this extends to a functor $\mathcal{EM}(T) \to T'$-**Alg** follows from example 2.1.1. We must then confirm that the resulting algebra is in $\mathcal{EM}(T')$. For the unit axiom we have:



(2.92)

For the multiplication axiom:



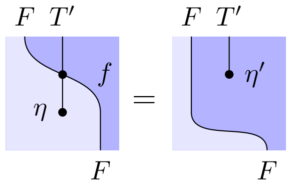
(2.93)

□

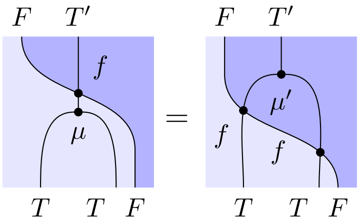
Remark 2.4.13. Every monad morphism (definition 2.4.3) is an endomorphism in **Monad** with the functor part the identity. Proposition 2.4.7 is then a special case of proposition 2.4.12.

Definition 2.4.14. The category **Monad*** is defined as having:

- **Objects**: Monads
- **Morphism**: A morphism of type $(\mathcal{C}, T, \eta, \mu) \to (\mathcal{C}', T', \eta', \mu')$ is a pair consisting of a functor $F : \mathcal{C} \to \mathcal{C}'$ and a natural transformation $f : FT \Rightarrow T'F$ satisfying the following equations:



(2.94a)



(2.94b)

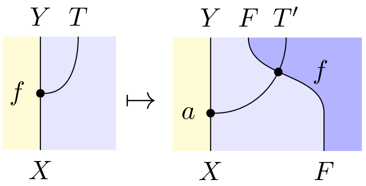
Proposition 2.4.15. *Every* **Monad****-morphism:*

$$(\mathcal{C}, T, \eta, \mu) \to (\mathcal{C}', T', \eta', \mu') \tag{2.95}$$

induces a functor:

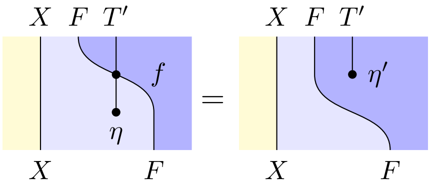
$$\mathcal{K}\ell(T) \to \mathcal{K}\ell(T') \tag{2.96}$$

Proof. On objects, the induced functor maps X to $F(X)$. On morphisms the action is:



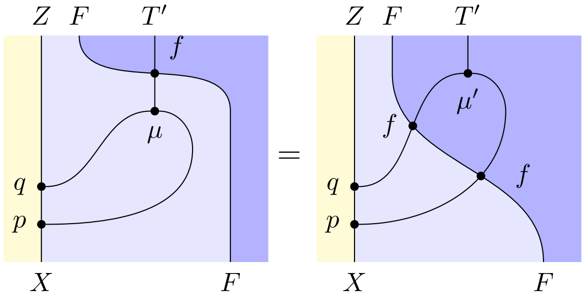
(2.97)

That this preserves identities follows immediately from the **Monad*** axiom (2.94a) as:



(2.98)

Functoriality of composition also follows immediately from the **Monad*** axiom (2.94b) as we have:



(2.99)

□

Remark 2.4.16. Every monad morphism (definition 2.4.3) is an endomorphism in

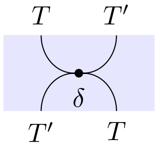
Monad* with the functor part the identity. Proposition 2.4.10 is then a special case of proposition 2.4.15.

Remark 2.4.17. The categories **Monad** and **Monad*** can be described more generally for an arbitrary 2-category, and in fact can be given the structure of 2-categories themselves, although we will not require this additional structure for our examples. For more details, see [Street, 1972] and [Street and Lack, 2002] and also the excellent exposition in the early parts of Power and Watanabe [2002].

2.4.3 Distributive Laws

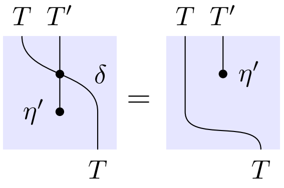
In lemma 2.3.2 we saw that adjunctions compose in a straightforward manner. Monads do not in general compose, but they can be composed in the presence of a suitable mediating natural transformation, referred to as a distributive law.

Definition 2.4.18 (Distributive Law)**.** For monads $(\mathcal{C}, T, \eta, \mu)$ and $(\mathcal{C}, T', \eta', \mu')$ a **distributive law** [Beck, 1969] is a natural transformation:

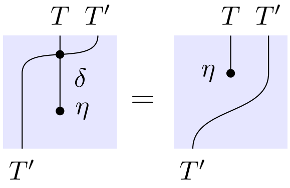


(2.100)

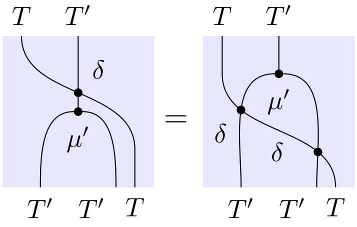
satisfying the following equations:



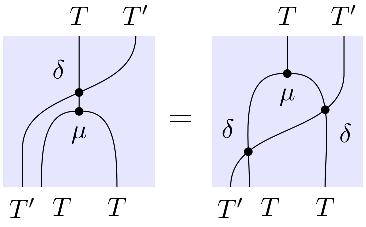
(2.101a)



(2.101b)

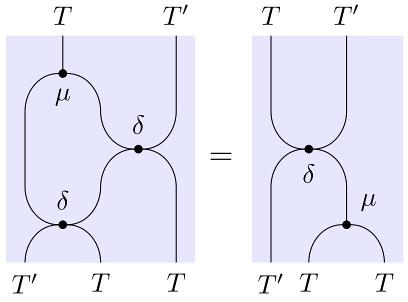


(2.101c)



(2.101d)

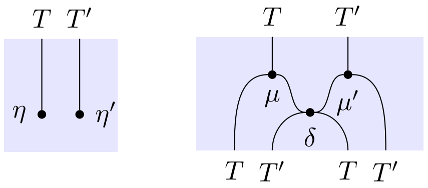
Remark 2.4.19 (Artistic Values). The axioms presented for a distributive law in definition 2.4.18 perhaps best illustrate the importance of how we choose to draw our string diagrams. In the axioms of equations (2.101a), (2.101b), (2.101c) and (2.101d) we choose to draw the natural transformation δ in different ways. The different choices of how δ is drawn allow us to emphasise the nature of the transformations as "sliding" various η and ϵ natural transformations across an appropriate line in the diagram. For example, we could instead have persisted with the neutral depiction of diagram (2.100), and drawn axiom (2.101d) as:



(2.102)

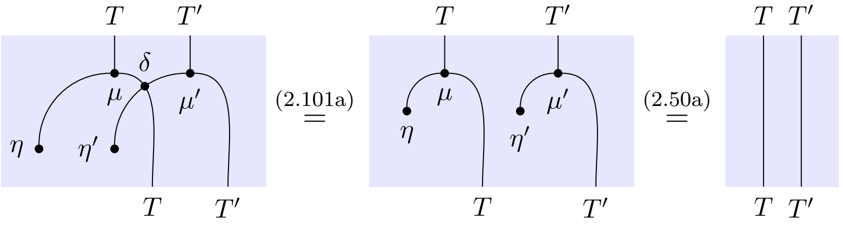
These diagrams describe exactly the same relationship, but the visual intuition for the nature of the axiom is completely lost.

Proposition 2.4.20. *Given monads* $(\mathcal{C}, T, \eta, \mu)$ *and* $(\mathcal{C}, T', \eta', \mu')$ *and a distributive law as in definition 2.4.18, then* $T' \circ T$ *gives a monad on* $\mathcal{C}$ *with unit and multiplication:*



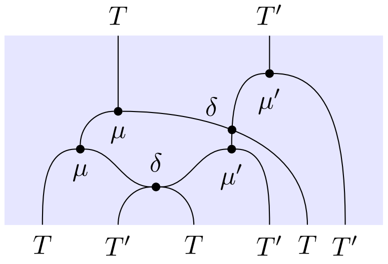
(2.103)

Proof. For the first of the unit axioms, we have the following equalities:

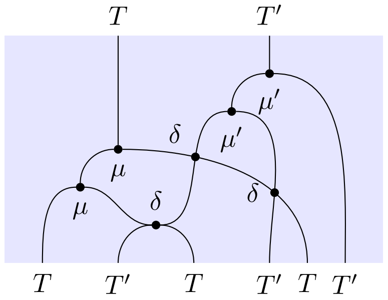


(2.104)

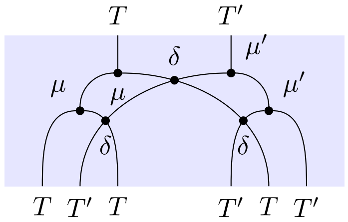
The second unit axiom follows dually using axiom (2.101b). The proof of associativity is a more interesting exercise in manipulating string diagrams. At each stage we attempt to draw our string diagrams so as to emphasise the forthcoming proof step, exploiting the topological flexibility of the notation.



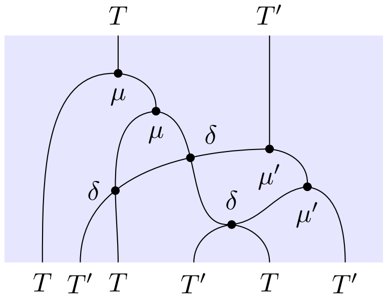
$=$ { distributive law axiom (2.101c) }



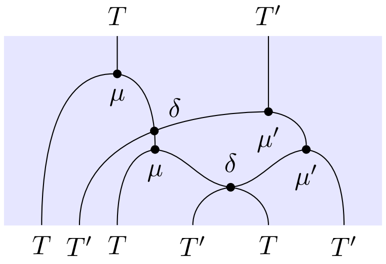
$=$ { monad associativity axiom (2.50b) }



$=$ { monad associativity axiom (2.50b) }



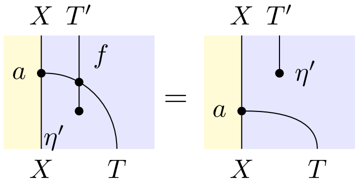
$= \quad \{ \text{ distributive law axiom (2.101c) } \}$



We note how the proof makes essential use of the symmetry of the monad and distributive law multiplication axioms. ☐

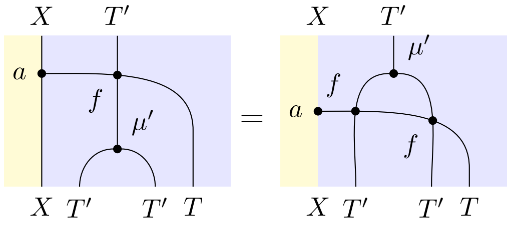
Proposition 2.4.21. *Given monads $(\mathcal{C}, T, \eta, \mu)$ and $(\mathcal{C}, T', \eta', \mu')$ and a distributive law as in definition 2.4.18, then T' induces a monad on $\mathcal{EM}(T)$.*

Proof. We first note that a distributive law is a **Monad**-morphism, and so T' lifts to an endofunctor on $\mathcal{EM}(T)$ by proposition 2.4.12. We take η' and μ' as the unit and multiplication. That η gives a natural transformation in $\mathcal{EM}(T)$ follow from:



(2.105)

Similarly that μ' is a natural transformation in $\mathcal{EM}(T)$ follows from:



(2.106)

That η' and μ' satisfy the monad axioms in then obvious as morphisms in $\mathcal{EM}(T)$ compose exactly as in $\mathcal{C}$. ☐

2.5 Representable Functors

We now investigate the important concept of representable functors. Many aspects of category theory can be phrased as requiring that a certain functor is representable.

Examined from the string diagrammatic perspective, representability provides a standard framework for generating calculation rules in a uniform manner. These calculation rules can then be specialized to provide graphical rules for limits, colimits, adjunctions, and left and right Kan extensions, unifying how these concepts can be approached graphically. Our use of representability bears some resemblance to the use of initiality to provide calculation rules in [Fokkinga, 1992b].

2.5.1 Covariant Representable Functors

Definition 2.5.1 (Covariant Representable Functor)**.** A covariant functor:

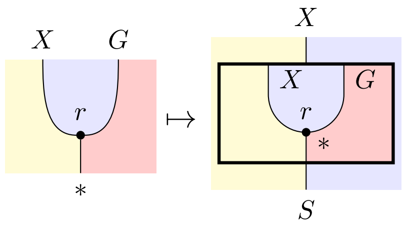
$$G : \mathcal{C} \to \mathbf{Set} \tag{2.107}$$

is said to be **representable** if there is an object S of $\mathcal{C}$ such that there is a natural isomorphism:

$$G \cong \mathcal{C}(S, -) \tag{2.108}$$

The elements of a set will be identified with morphisms from the one element set, as discussed in section 2.2.1. We introduce a box notation to represent the mappings of the natural isomorphism, similar to the notation for monoidal functors introduced in [Cockett and Seely, 1999, Blute et al., 2002] and described in detail in [Melliès, 2006]. For an object X in $\mathcal{C}$ the mapping from left to right in (2.108) can be drawn graphically as:

$$G(X) \to \mathcal{C}(S, X) \tag{2.109}$$

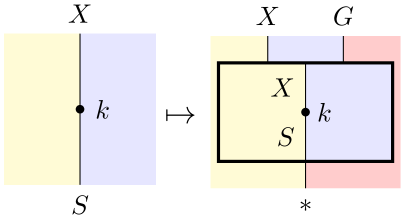


(2.110)

We can think of this isomorphism as taking an element $r \in G(X)$ and producing a morphism of type $S \to X$ named after it. The box notation is then just a particularly large vertex, with the material inside the box simply indicating the element of $G(X)$ naming the morphism.

Similarly, we can draw the mapping from right to left in (2.108) as:

$$\mathcal{C}(S, X) \to G(X) \tag{2.111}$$

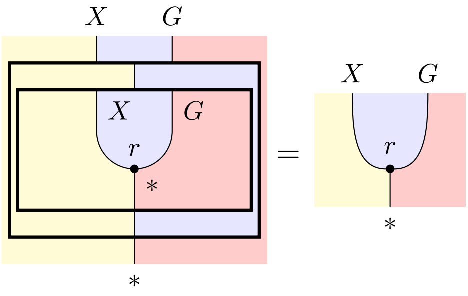


(2.112)

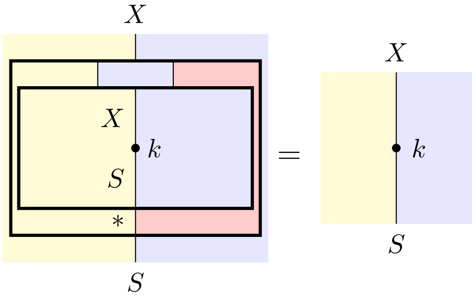
Again, we can consider this direction of the isomorphism as taking a morphism of type $S \to X$ and producing an element of $G(X)$ named after it. As before, from this perspective the box notation is a special case of the vertices we have seen in string diagrams in earlier sections. The contents of the box conveniently show the morphism from which the element was constructed.

As the box notation is an instance of the usual string diagram notation for natural transformations, boxes will behave exactly like any other natural transformation in their containing diagram. In particular, no additional mathematics has been required to introduce the box notation. We can manipulate the contents of a box as if it were an independent string diagram, as equal contents of a box will name the same natural transformation in the larger diagram. Similarly, we can manipulate the contents of a diagram outside a box as we would normally. The fact that the mappings given by representability are natural isomorphisms will provide us with equations between various diagrams involving boxes. These equations then allow us to move things in and out of the boxes in useful ways, as we now discuss.

As these representability mappings are isomorphisms, the following two conditions hold:



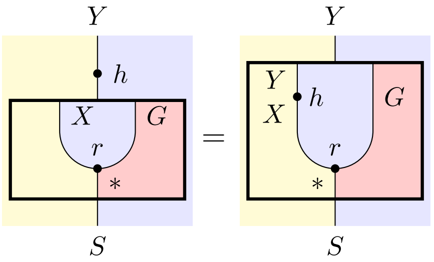
(2.113a)



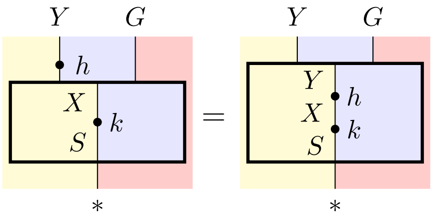
(2.113b)

Informally, *the box of a box is the identity.*

The important aspect of representability is that the isomorphism is natural. This can be seen as requiring that the following two calculation rules hold, allowing us to "push and pop" morphisms in and out of the box notation.



(2.114a)



(2.114b)

The next lemma shows that to prove representability it is sufficient to show a family of bijections satisfy just one of the two equations above.

Lemma 2.5.2. *For a given family of isomorphisms of hom sets:*

$$\theta_X : F(X) \cong \mathcal{C}(S, X) \tag{2.115}$$

the conditions in equations (2.114a) *and* (2.114b) *are equivalent.*

Proof. It is easy to show that if every component of a natural transformation has an inverse, then the inverses form a natural transformation. The claim then follows immediately. □

Remark 2.5.3. Given a representable functor:

$$G \cong \mathcal{C}(S, -) \tag{2.116}$$

generally we will know more about the structure of the left hand side than just that it is a functor with codomain **Set**. It will be common for G to involve a hom bifunctor $\mathcal{C}(-,-)$ in some way. This additional structure can then be used to put the above equations into a more convenient form than reasoning in terms of elements of an abstract set. This will be seen in the examples in section 2.5.3.

Remark 2.5.4 (Contravariant Representable Functors). Representable functors of the form $\mathcal{C}(-, R) : \mathcal{C}^{op} \to \mathbf{Set}$ will also be of interest. These dual properties to those discussed above, and we will deal with contravariant representable functors later where necessary. We avoid a full explicit discussion for space reasons.

2.5.2 Universality from Representables

Much of category theory is concerned with the important notion of a universal property. We now show how the representability of a functor leads to a particular universal property. When more structure is known about the representable functor, we can recover well known universal properties such as those of adjunctions, limits, colimits and Kan extensions, as will be discussed in later sections and examples.

Definition 2.5.5 (Counit)**.** For a representable functor:

$$G \cong \mathcal{C}(R, -) \tag{2.117}$$

the **counit** is the element of $G(R)$ that is the image of 1_R under the natural isomorphism.

Proposition 2.5.6. *For a representable functor:*

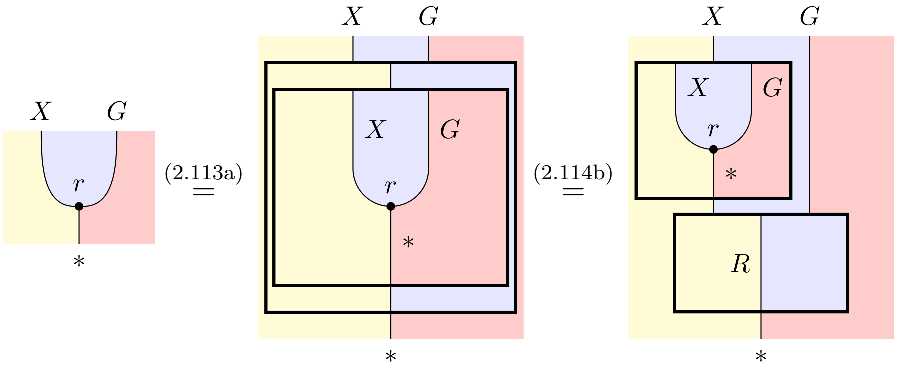
$$G \cong \mathcal{C}(R, -) \tag{2.118}$$

and X an object of $\mathcal{C}$, for every element $r \in G(X)$, there exists a unique f such that r can be written in the form:

$$r = G(f)(c) \tag{2.119}$$

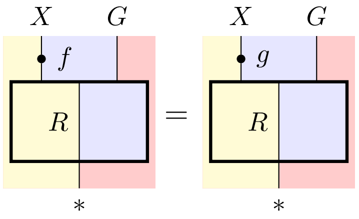
where c is the counit of the representable functor.

Proof. For an arbitrary $r \in F(X)$ we have the following sequence of equalities:

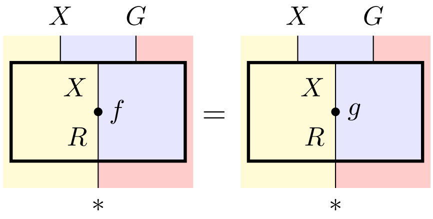


(2.120)

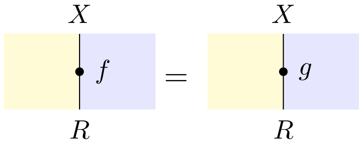
Now assume $r = G(f)(c)$ and $r = G(g)(c)$, then:



⇔ { push / pop equation (2.114b) }



⇔ { isomorphism }

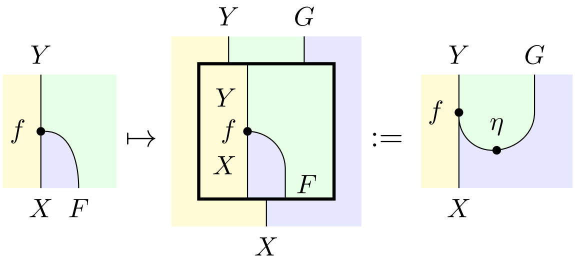


□

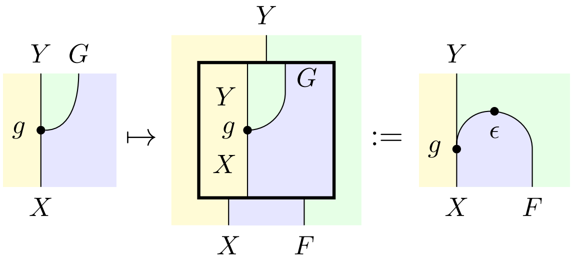
2.5.3 Examples of Representable Functors

In this section we will consider two important applications of the representability based approach of section 2.5, adjunctions and Kan extensions. Another significant source of examples are limits and colimits, these will be discussed in detail later in section 2.6.

Example 2.5.7 (Adjunctions)**.** For $F \dashv G$, for each X and Y we have mappings:

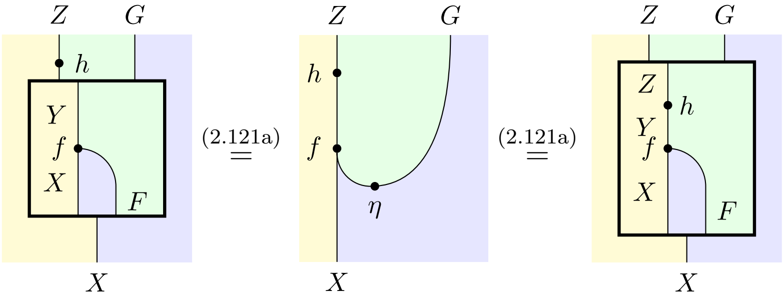


(2.121a)



(2.121b)

It is immediate from the adjunction axioms that these maps are mutually inverse. Naturality of the first mapping follows easily from the following equalities:



(2.122)

Naturality of the second map then follows from lemma 2.5.2. We remark that the usual bijection, natural in X and Y induced by an adjunction:

$$\frac{F(X) \to Y}{\overline{\overline{X \to G(Y)}}}$$

can also be seen as an immediate corollary of the naturality of the move left and move right wire bending mappings of definition 2.3.3.

Example 2.5.8 (Kan Extensions)**.** A novel graphical notation for Kan extensions was introduced in [Hinze, 2012]. The paper then develops many proofs using both a traditional symbolic style and a string diagram based graphical approach, illustrating the compactness and efficiency of the latter.

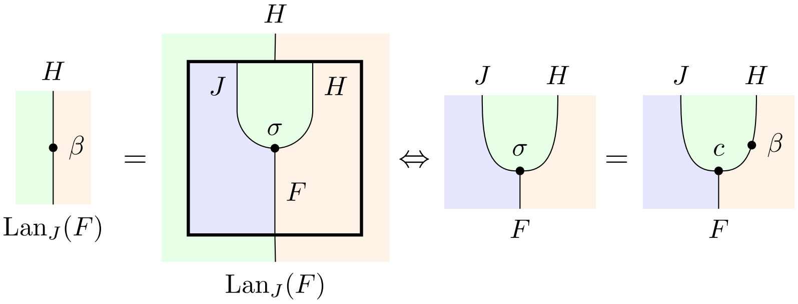
We will examine Kan extensions as an example application of the representability based results we have developed in earlier sections. Given functors $J : \mathcal{B} \to \mathcal{A}$ and $F : \mathcal{B} \to \mathcal{C}$, F is said to have a **left Kan extension** along J if the functor $[\mathcal{B}, \mathcal{C}](F, (-) \circ J)$ is representable. Then we have:

$$[\mathcal{B}, \mathcal{C}](F, (-) \circ J) \cong [\mathcal{A}, \mathcal{C}](\mathrm{Lan}_J(F), -) \tag{2.123}$$

Dually, F has a **right Kan extension** along J if we have the following natural isomorphism:

$$[\mathcal{B}, \mathcal{C}]((-) \circ J, F) \cong [\mathcal{A}, \mathcal{C}](-, \mathrm{Ran}_J(F)) \tag{2.124}$$

If we denote the counit as c, in the case of left Kan extensions, the universality results of section 2.5.2 specialize to the following axiom, giving the usual unique factorization property of left Kan extensions:



(2.125)

We recover the calculation laws of [Hinze, 2012] as follows:

- The **computation law** follows immediately from the universal property above
- The **reflection law** follows directly from the definition of the counit
- The **fusion law** is an instance of the "push / pop" identity in equation (2.114a)

2.6 Limits and colimits

Limits and colimits are key notions in category theory. They can be approached from the perspective of representability introduced in section 2.5. We will start with the general setting, and then provide specialized results and notation for some common cases.

2.6.1 Arbitrary Limits and Colimits

We first define a standard functor used when reasoning about limits and colimits.

Definition 2.6.1 (Diagonal Functor)**.** Let $\mathcal{C}$, $\mathcal{D}$ be categories. We define the **diagonal functor** $\Delta : \mathcal{C} \to [\mathcal{D}, \mathcal{C}]$ as taking an object to the corresponding constant functor, and a morphism f to the natural transformation with all components equal to f.

Now we define limits and colimits in terms of representability of appropriate functors.

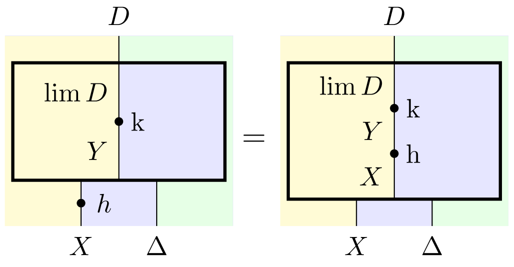
Definition 2.6.2. Let $\mathcal{C}$, $\mathcal{D}$ be categories, and the functor $D : \mathcal{D} \to \mathcal{C}$ a diagram in $\mathcal{C}$. The limit of D exists if the functor $[\mathcal{D}, \mathcal{C}](\Delta(-), D)$ is representable, i.e.

$$[\mathcal{D}, \mathcal{C}](\Delta(-), D) \cong \mathcal{C}(-, \lim D) \tag{2.126}$$

The limiting cone then corresponds to the identity morphism on $\lim D$. Dually, the colimit of D exists if the functor $[\mathcal{D}, \mathcal{C}](D, \Delta(-))$ is representable, i.e.

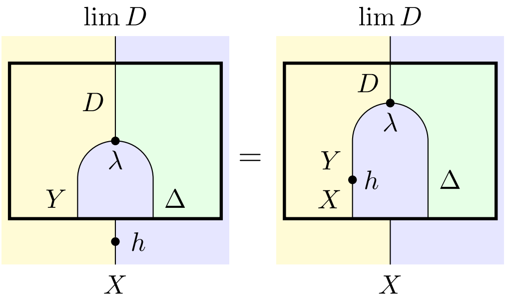
$$[\mathcal{D}, \mathcal{C}](D, \Delta(-)) \cong \mathcal{C}(\operatorname{colim} D, -) \tag{2.127}$$

As the existence of limits and colimits corresponds to representability of certain functors, we can use the calculation rules in section 2.5. Our additional knowledge of the structure of the functor on the left hand side allows us to phrase the calculation laws in a more convenient form. For limits we have:



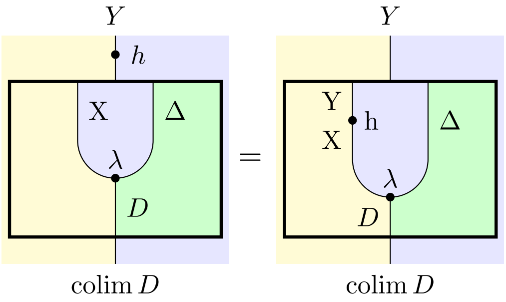
(2.128)

and:

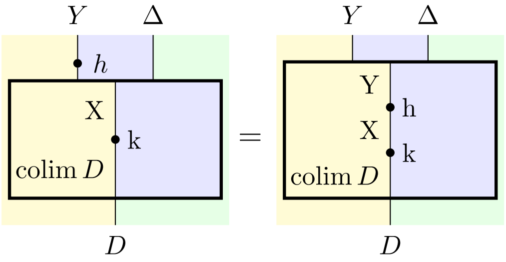


(2.129)

For colimits the following equations hold:



(2.130)



(2.131)

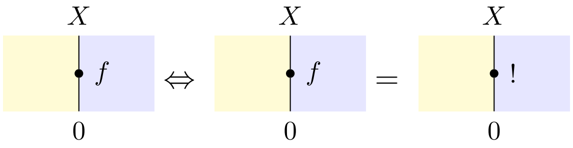
We will specialize the universality result described in section 2.5.2 when we discuss some specific limits and colimits in the following section.

2.6.2 Specific Limits and Colimits

We now examine a few specific limits and colimits. In these concrete cases we can specialize the results in section 2.6.1 and provide more convenient notation and calculation rules.

Initial Objects

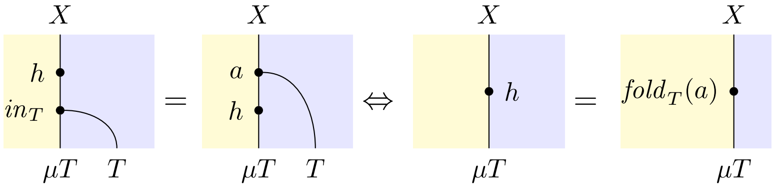
We will use 0 to denote an initial object, and $! : 0 \to X$ to denote the unique morphism from the initial object to an arbitrary object X. The universal property for initial objects can be expressed by the following relationship, for each object X in $\mathcal{C}$:



(2.132)

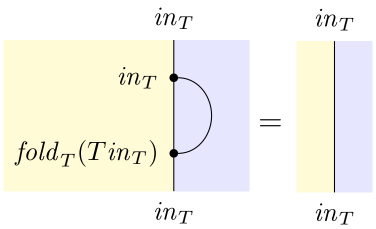
We will now further specialize this rule for initial algebras, and as an example calculation apply the rule to provide a graphical proof of Lambek's lemma. Initial algebras, and their dual notion terminal coalgebras, are of interest in modelling of datatypes [Goguen et al., 1975, 1977, Hagino, 1987a,b, Malcolm, 1990a,b].

Example 2.6.3 (Initial Algebras and Lambek's Lemma)**.** For endofunctor $T : \mathcal{C} \to \mathcal{C}$, if T has an initial algebra $\left(\begin{array}{c} \mathrm{T}\ \mu T \\ \downarrow in_T \\ \mu T \end{array}\right)$ then we can rewrite the initiality condition in a more useful form. We also adopt some standard notation in this context and denote the unique morphism from the initial algebra to an algebra a as $fold_T(a)$.

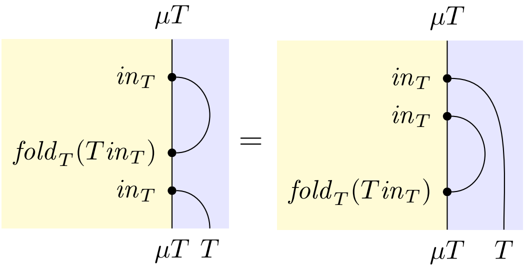


(2.133)

It is immediately obvious that $fold_T(in_T) = 1_{\mu T}$. We can then prove Lambek's lemma, that the initial algebra is an isomorphism. We have:



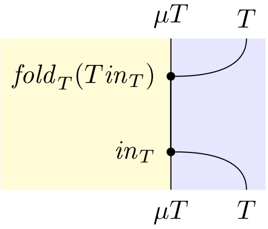
$\Leftrightarrow$ { initial algebra law and $fold_T(in_T) = 1_{\mu T}$ }



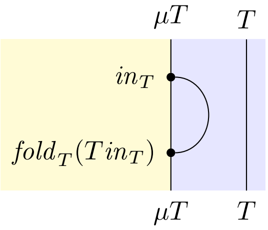
$\Leftrightarrow$ { $fold_T(T\,in_T)$ is an algebra homomorphism }

true

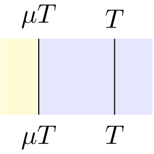
For the other direction we have:



$=$ { $fold_T(T\,in_T)$ is an algebra homomorphism }

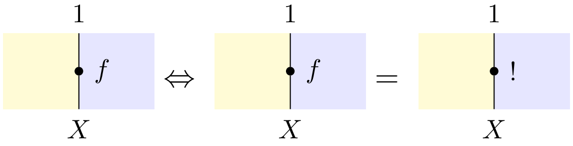


$=$ { previous part }



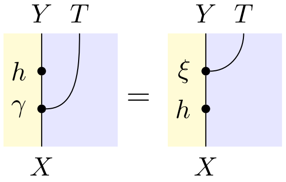
Terminal Objects

Terminal objects are the dual notion to the initial objects of section 2.6.2. The terminal object will be denoted 1, and $! : X \to 1$ will denote the unique morphism from an object X to the terminal object. The universal property of the terminal object can then be written, for each object X in $\mathcal{C}$:



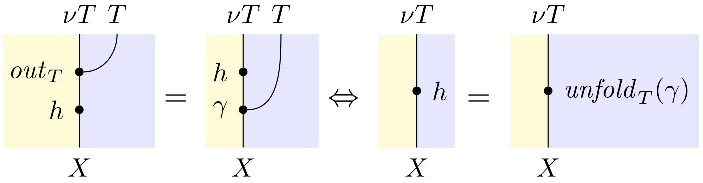
(2.134)

Example 2.6.4 (Terminal Coalgebras and the Fusion Law)**.** As we will be interested in coalgebras in chapters 3, 4, 5 and 6, we now introduce a standard coalgebraic example. For an endofunctor $T : \mathcal{C} \to \mathcal{C}$ a **coalgebra** is a pair consisting of an object X and a morphism $X \xrightarrow{\gamma} T(X)$, which, following the notation for algebras introduced in example 2.1.1, we shall denote vertically as $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$. The functor T is referred to as the **signature functor**. A T**-coalgebra morphism** of type $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix} \to \begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix}$ is a $\mathcal{C}$-morphism $h : X \to Y$ satisfying:



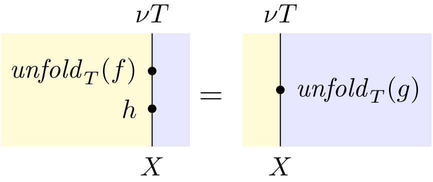
(2.135)

For fixed endofunctor $T : \mathcal{C} \to \mathcal{C}$ the category of coalgebras and coalgebra morphisms will be denoted T-**Coalg**. Dually to the situation with algebras in example 2.6.3, if an endofunctor T has a terminal coalgebra $\begin{pmatrix} \nu T \\ \downarrow out_T \\ \mathrm{T}\ \nu T \end{pmatrix}$, we can rewrite the terminal object condition for coalgebras in a more convenient form. We adopt the standard notation of $unfold_T(\gamma)$ for the unique morphism from a coalgebra γ to the terminal coalgebra, giving:

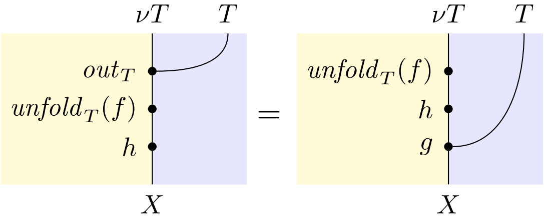


(2.136)

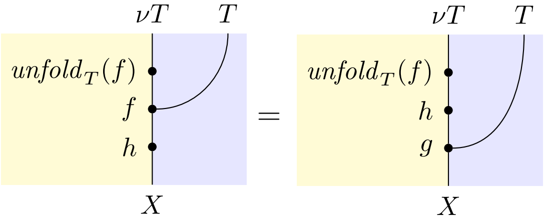
The **strong fusion law** is an important property of terminal coalgebras. It can be deduced from the following chain of equivalences:



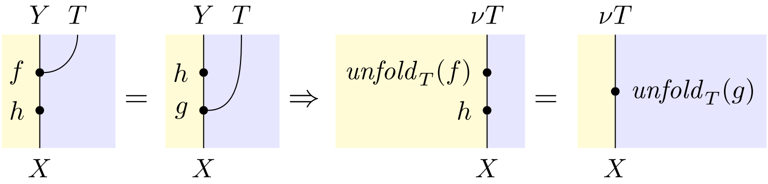
⇔ { universal property of terminal coalgebra }



⇔ { *unfold*$_T(f)$ is a coalgebra homomorphism }



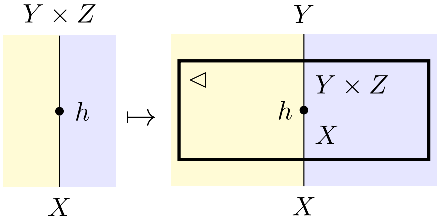
The intuitive explanation of the fusion law is that if the final equality is satisfied, then we can "fuse" the morphism h with *unfold*$_T(f)$ to give a single unfold morphism *unfold*$_T(g)$. As the precondition for fusion is rather cumbersome, it is often easier to apply the **weak fusion law**, which follows as an immediate corollary via Leibniz:



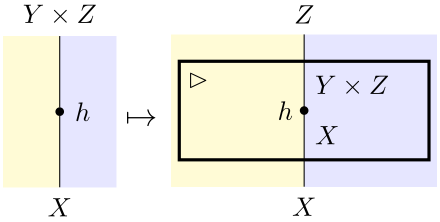
(2.137)

Binary Products and Coproducts

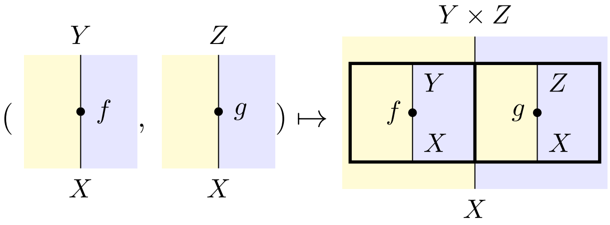
The binary product of objects Y and Z will be written in the usual way as $Y \times Z$. The representability condition gives a bijective correspondence between morphisms $X \to Y \times Z$ and pairs of morphisms $X \to Y$ and $X \to Z$. To aid calculations we will introduce three different maps:



(2.138)

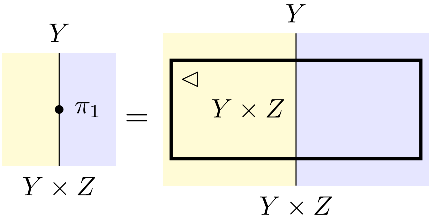


(2.139)

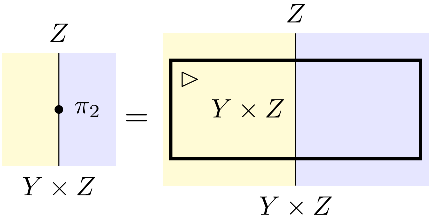


(2.140)

We then define the usual notation for the two components of the unit:

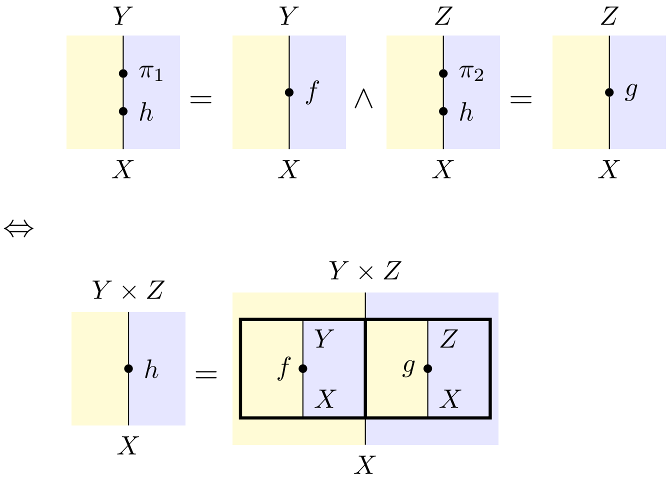


(2.141)



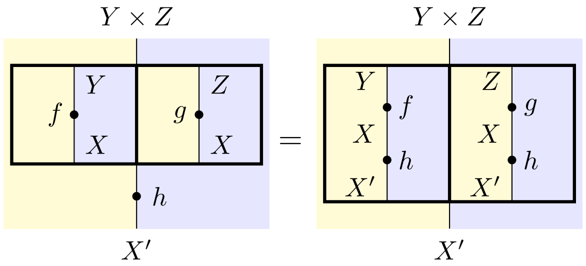
(2.142)

Using the material in section 2.5.2 we can then write the universal property of products as:

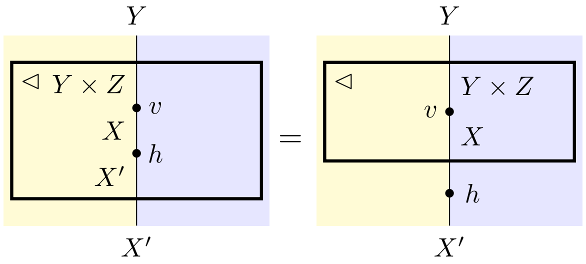


(2.143)

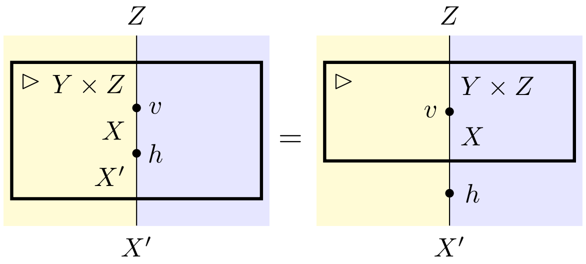
The "push / pop" equations (2.114a) and (2.114b) then lead to the following three equalities:



(2.144)

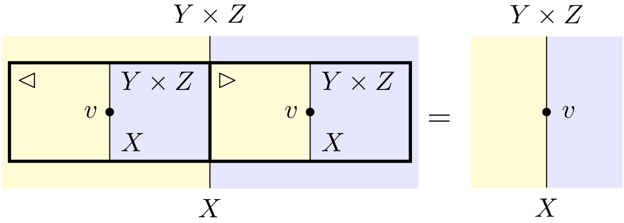


(2.145)

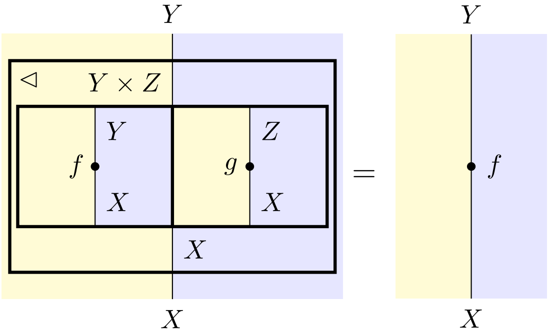


(2.146)

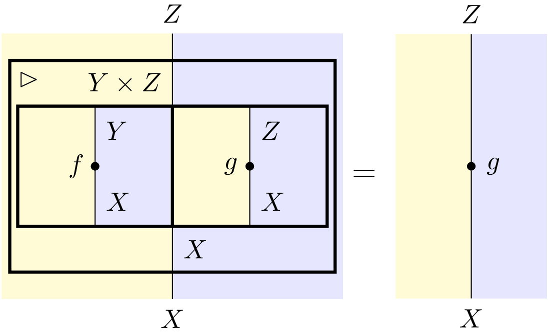
That these maps witness a bijection leads to the following three of equalities:



(2.147)



(2.148)



(2.149)

Those familiar with more standard introductions to category theory will hopefully recognise many standard exercises in the properties of binary products are given in graphical form by the various equations above.

We can develop the dual notion of binary coproducts in a symmetrical manner, we avoid a detailed exposition here keep the length of this chapter manageable.

Remark 2.6.5. Throughout this section there has been no attempt to present a minimal set of equations. Many of the equalities above are interderivable, but we have aimed to give explicit statements of many useful properties, and to exploit results about of representable functors to prove standard identities from general principles.

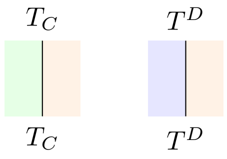
2.7 Bifunctors via Sections

In earlier sections, for example the account of binary products in section 2.6.2, we have implicitly been dealing with bifunctors. We now describe a general strategy for handling bifunctors using string diagrams. Our approach can be seen as using “sections” in the terminology of functional programming. Bifunctors are slightly awkward in the string diagrammatic framework as ideally we would use horizontal juxtaposition to describe the parameters to the bifunctor, in the style used in the calculi for monoidal categories. Unfortunately we have already used this graphical freedom to describe functor composition; if we wish to continue working with 2-dimensional diagrams we must make some compromises.

Consider a bifunctor:

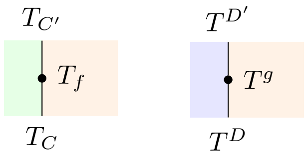
$$T : \mathcal{C} \times \mathcal{D} \to \mathcal{E} \tag{2.150}$$

By fixing either the first or second parameter, each $\mathcal{C}$-object C and $\mathcal{D}$-object D induce (unary) functors:



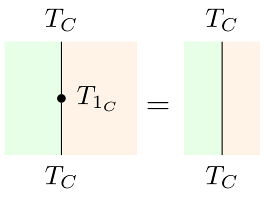
(2.151)

Also, a $\mathcal{C}$-morphism $f : C \to C'$ and $\mathcal{D}$-morphism $g : D \to D'$ induce natural transformations:

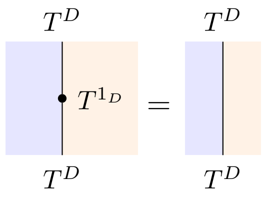


(2.152)

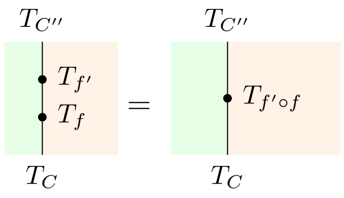
These satisfy the following obvious functoriality equations:



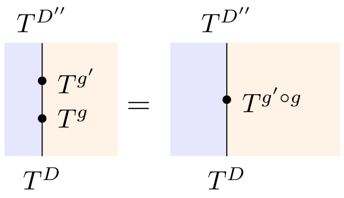
(2.153a)



(2.153b)

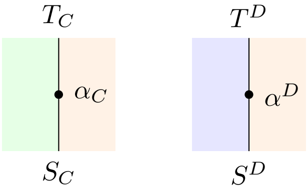


(2.153c)



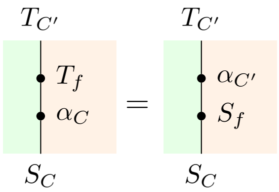
(2.153d)

For bifunctors $S, T : \mathcal{C} \times \mathcal{D} \to \mathcal{E}$ we can consider a natural transformation $\alpha : S \Rightarrow T$, typically $\alpha_{C,D}$ is then said to be "natural in both C and D". Again we can fix either the first or second parameter, giving for each $\mathcal{C}$-object C and each $\mathcal{D}$-object D families of natural transformations:

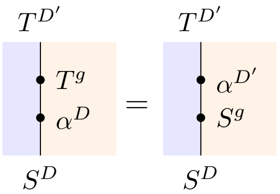


(2.154)

Naturality in the fixed parameters must then be handled explicitly as the satisfaction of the following equations:



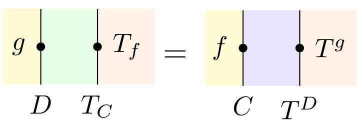
(2.155a)



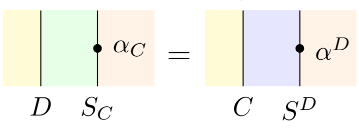
(2.155b)

We can then work with transformations natural in two parameters by choosing one parameter to fix. Naturality in the other parameter then behaves in the usual way for string diagrams, and naturality in the fixed parameter is captured equationally as the commutativity conditions of equations (2.155a) and (2.155b). Some of the calculations are now effectively being performed in the subscripts and superscripts in these diagrams, but we do retain a topological feel to our reasoning. An example of the application of this graphical approach to naturality in two parameters appears in the proof of theorem 5.6.11 in chapter 5.

We can also relate the two choices of notation via the following equations:

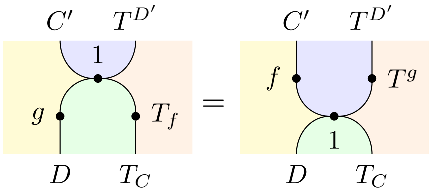


(2.156)



(2.157)

The relationships in equations (2.156) and (2.157) illustrate a new phenomenon. We have equalities in which the functors at the top and bottom of the diagrams on each side are apparently different, but the composites are actually equal. For example in this case by definition we have equalities $T_X Y = T(X, Y) = T^Y X$. These types of equations between composite functors are an occasion on which it can be useful to explicitly insert identity natural transformations in order to witness the equalities. For example, we can rewrite equation (2.156) in a topologically more instructive form as:



(2.158)

These **witnessing identity morphisms** then appear as a trivial form of distributive law, smoothing diagrammatic calculations by allowing us to switch between two equal composite functors.

Chapter 3

Quantum Symmetries Coalgebraically

3.1 Introduction

When describing the behaviour of a quantum system, interactions with the system can be divided into two types:

- Measurements
- Reversible evolution of the system state

Measurement outcomes are probabilistic, and their properties are governed by the important Born rule of quantum mechanics. The reversible evolution of the system state is clearly related to group actions on the underlying mathematical model. This suggests when we model the measurement behaviour of a quantum system coalgebraically we should hope to find the appropriate symmetries as automorphisms of our "quantum" coalgebra.

In categories of coalgebras, morphisms are functional bisimulations, in particular, for a strongly extensional coalgebra in which bisimilarity reduces to equality, there can only be one automorphism. This leads to a tension between the nature of coalgebra homomorphisms and the desire to model symmetries effectively.

To resolve this tension, Abramsky [2010] introduced a novel fibrational structure, in which a representation of the physical symmetries of a quantum system could be found as automorphisms of a particular coalgebra. A similar construction was used in Kurz and Pattinson [2000a] to model parameterization of coalgebras, and logical aspects were developed in Kurz and Pattinson [2000b]. This construction will form the starting point for the investigation of what a "fibred coalgebraic logic" would mean in chapter 5.

In this chapter we aim for a more elementary formulation of a coalgebraic representation of the unitaries. We emphasize the importance of automorphisms of the signature functor as a uniform method of reversibly adapting the dynamics of a given coalgebra. A suitable choice of group of automorphisms can then be chosen to weaken the notion of coalgebra morphism in order provide the flexibility to model symmetries successfully.

We assume an understanding of elementary coalgebra at the level of [Rutten, 2000, Jacobs and Rutten, 2011, Jacobs, 2012b]. We also assume some basic background in quantum computation, for example at the level of [Mermin, 2007] or [Nielsen and Chuang, 2010]. We will continue to use string diagrams in our calculations where appropriate. Applications of string diagrams in algebraic and coalgebraic calculations were introduced in examples 2.1.1, 2.6.3 and 2.6.4 of chapter 2.

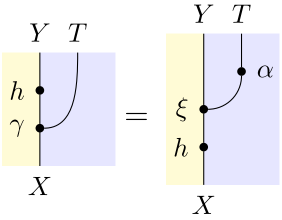
3.2 Weakening the Notion of Morphism for Coalgebras

We now introduce a generalization of the notion of morphism between coalgebras, as defined in example 2.6.4. We will require that equality (2.135) only holds "up to isomorphism", in a manner that will be made precise shortly. Our general programme is then to extend coalgebraic notions such as bisimulation to this new setting in an appropriate manner. These generalized notions will be given the prefix "pseudo" to distinguish them from the usual terminology. To avoid an excessively technical presentation, we shall often restrict our attention to coalgebras over the category **Set**.

Definition 3.2.1. For category $\mathcal{C}$, endofunctor $T : \mathcal{C} \to \mathcal{C}$ and G a subgroup of of the automorphisms of T (natural isomorphims of type $T \Rightarrow T$), define the category T-**PseudoCoalg**(G) as follows:

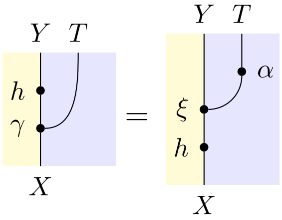
- Objects: T-coalgebras
- Morphisms: A morphism $\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \to \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix}$ is a $\mathcal{C}$ morphism $h : X \to Y$ such

that there exists an $\alpha \in G$ making the following equality hold:

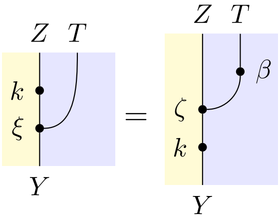


(3.1)

Remark 3.2.2. That composition in T-**PseudoCoalg**(G) is well defined in definition 3.2.1 is straightforward to verify. If we assume two morphisms satisfying:

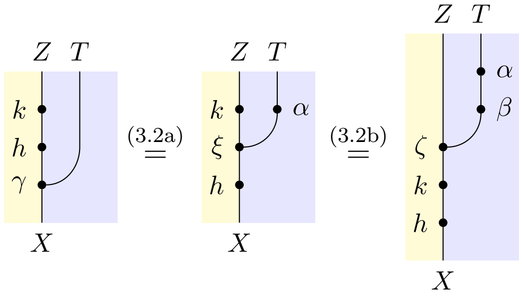


(3.2a)



(3.2b)

We then have equations:

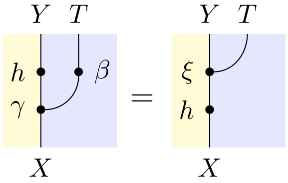


(3.3)

As G is a subgroup of the automorphisms of T, it is closed under composition and so the composite $k \circ h$ is a morphism.

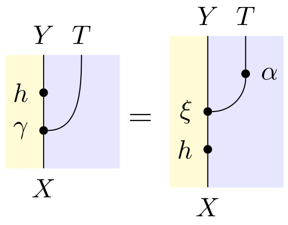
The following simple lemma gives some symmetry to the condition for being a morphism, we will use either condition as convenient in subsequent sections.

Lemma 3.2.3. *For the assumptions in definition 3.2.1, there exists $\alpha \in G$ such that equation* (3.1) *holds if and only if there exists $\beta \in G$ such that:*

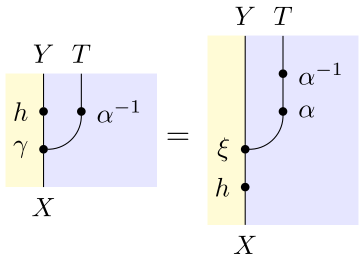


(3.4)

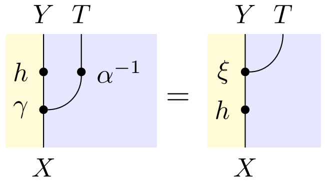
Proof. As G is a group, every $\alpha \in G$ has a inverse $\alpha^{-1} \in G$, and we have the following equivalences:



$\Leftrightarrow$ { bijection }



$\Leftrightarrow$ { inverses }



□

We recall the definition of bisimulation in **Set** coalgebra theory:

Definition 3.2.4 (Bisimulation)**.** Let $T : \mathbf{Set} \to \mathbf{Set}$ be an endofunctor, and $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$ and $\begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix}$ two T-coalgebras. A **bisimulation** between γ and ξ is a relation $R \subseteq X \times Y$ such that there exists a coalgebra $\begin{pmatrix} R \\ \downarrow\rho \\ TR \end{pmatrix}$ making the following diagram commutes:

$$\begin{tikzcd}
X \arrow[d, "\gamma"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\rho"'] & Y \arrow[d, "\xi"] \\
TX & TR \arrow[l, "T\pi_1"] \arrow[r, "T\pi_2"'] & TY
\end{tikzcd} \tag{3.5}$$

where π_1, π_2 are the obvious projections. Two elements $x \in X$ and $y \in Y$ are said to be **bisimilar** if they are related by a bisimulation.

Remark 3.2.5. The coalgebraic definition of bisimilarity generalizes earlier notions of bisimilarity [Park, 1981] studied in concurrency, automata theory and modal logic. Intuitively, two states are bisimilar if they exhibit the same observable behaviour. The graph of any coalgebra homomorphism is a bisimulation, and so homomorphisms are sometimes described as functional bisimulations. So coalgebra homomorphisms "preserve behaviour", motivating our need to consider a broader class of morphisms.

Definition 3.2.6. In T-**PseudoCoalg**(G):

- A morphism where the usual coalgebra homomorphism condition holds strictly will be referred to as a **functional bisimulation**
- A morphism of the form $1_X : \begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \to \begin{pmatrix} X \\ \downarrow \alpha_X \circ \gamma \\ TX \end{pmatrix}$ will be referred to as an **adaptation**

Lemma 3.2.7. *Let $\mathcal{C}$ be a category, $T : \mathcal{C} \to \mathcal{C}$ an endofunctor, and G a subgroup of the automorphisms of T:*

1. *Every morphism in T-**PseudoCoalg**(G) factors as an adaption followed by a functional bisimulation*
2. *Every morphism in T-**PseudoCoalg**(G) factors as a functional bisimulation followed by an adaptation*

Proof. Consider T-**PseudoCoalg**(G) morphism $\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix}$ There then exists $\alpha \in G$ such that h is a coalgebra homomorphism $\begin{pmatrix} X \\ \downarrow \alpha_X \circ \gamma \\ TX \end{pmatrix} \to \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix}$. For part 1 we take the factorization:

$$\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \xrightarrow{1} \begin{pmatrix} X \\ \downarrow \alpha_X \circ \gamma \\ TX \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix} \tag{3.6}$$

Also there must exists $\beta \in G$ such that h is a coalgebra homomorphism $\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \to$

$\left(\begin{array}{c} Y \\ \downarrow \beta_Y \circ \xi \\ TY \end{array}\right)$. For part 2 we take the factorization:

$$\left(\begin{array}{c} X \\ \downarrow \gamma \\ TX \end{array}\right) \xrightarrow{h} \left(\begin{array}{c} Y \\ \downarrow \beta_Y \circ \xi \\ TY \end{array}\right) \xrightarrow{1} \left(\begin{array}{c} Y \\ \downarrow \xi \\ TY \end{array}\right) \tag{3.7}$$

□

Definition 3.2.8 (Pseudo-bisimulation)**.** Let T : **Set** $\to$ **Set** be an endofunctor and G a subgroup of the automorphisms of T. A **pseudo-bisimulation** between coalgebras $\left(\begin{array}{c} X \\ \downarrow \gamma \\ TX \end{array}\right)$ and $\left(\begin{array}{c} Y \\ \downarrow \xi \\ TY \end{array}\right)$ is a relation $R \subseteq X \times Y$ such that the span given by the projection morphisms in **Set** lifts to a span:

$$\left(\begin{array}{c} X \\ \downarrow \gamma \\ TX \end{array}\right) \xleftarrow{\pi_1} \left(\begin{array}{c} R \\ \downarrow \sigma \\ TR \end{array}\right) \xrightarrow{\pi_2} \left(\begin{array}{c} Y \\ \downarrow \xi \\ TY \end{array}\right) \tag{3.8}$$

in T-**PseudoCoalg**(G), for some structure map σ.

We now outline some simple properties of pseudo-bisimulations, in analogy to [Rutten, 2000]. The proofs either follow from the corresponding property of bisimulations, or from a slight modification of the associated proof.

Lemma 3.2.9. *For an endofunctor T :* **Set** $\to$ **Set** *and a subgroup G of the automorphisms of T, if $h : (X, \gamma_1) \to (Y, \gamma_2)$ is a morphism in T-***PseudoCoalg***(G), then the graph of h is a pseudo-bisimulation.*

Proof. Assume $h : (X, \gamma_1) \to (Y, \gamma_2)$ is a morphism in T-**PseudoCoalg**(G), then there exists $\alpha \in G$ such that we have coalgebra morphism $h : (X, \alpha_X \circ \gamma_1) \to (Y, \gamma_2)$, and so the graph of h gives a bisimulation between these coalgebras, and so a pseudo-bisimulation between (X, γ_1) and (Y, γ_2). □

Lemma 3.2.10. *For an endofunctor T :* **Set** $\to$ **Set** *and a subgroup G of the automorphisms of T:*

1. *Every bisimulation is a pseudo-bisimulation*

2. *The converse of a pseudo-bisimulation is a pseudo-bisimulation*

3. *For morphisms $f : (X, \gamma_1) \to (Y, \gamma_2)$ and $g : (X, \gamma_1) \to (Z, \gamma_3)$, $\mathrm{im}(\langle f, g\rangle)$ is a pseudo-bisimulation between (Y, γ_2) and (Z, γ_3)*

4. If T preserves weak pullbacks then pseudo-bisimulations compose as relations

5. If T preserves weak pullbacks then the kernel of a morphism is a pseudo-bisimulation

Proof. Parts 1 and 2 are obvious.

For part 3 there must exist $\alpha, \beta \in G$ such that f is a coalgebra morphism $f : (X, \gamma_1) \to (Y, \alpha_Y \circ \gamma_2)$ and g is a coalgebra morphism $g : (X, \gamma_1) \to (Z, \beta_Z \circ \gamma_3)$. Therefore $\mathrm{im}(\langle f, g\rangle)$ is a bisimulation between $(Y, \alpha_Y \circ \gamma_2)$ and $(Z, \beta_Z \circ \gamma_3)$, which is then a pseudo-bisimulation between (Y, γ_2) and (Z, γ_3).

For part 4 assume (R_1, γ_{R_1}) is a pseudo-bisimulation between (X, γ_1) and (Y, γ_2), then there exist $\alpha, \alpha' \in G$ such that the following diagram commutes:

$$\begin{tikzcd}
X \arrow[d, "\gamma_1"'] & R_1 \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[dd, "\gamma_{R_1}"] & Y \arrow[d, "\gamma_2"] \\
TX \arrow[d, "\alpha_X"'] & & TY \arrow[d, "\alpha'_Y"] \\
TX & TR_1 \arrow[l, "T\pi_1"] \arrow[r, "T\pi_2"'] & TY
\end{tikzcd} \tag{3.9}$$

Also assume (R_2, γ_{R_2}) is a pseudo-bisimulation between (Y, γ_2) and (Z, γ_3), then there exist $\alpha'', \alpha''' \in G$ such that the following diagram commutes:

$$\begin{tikzcd}
Y \arrow[d, "\gamma_2"'] & R_2 \arrow[l, "\pi'_1"'] \arrow[r, "\pi'_2"] \arrow[dd, "\gamma_{R_2}"] & Z \arrow[d, "\gamma_3"] \\
TY \arrow[d, "\alpha''_Y"'] & & TZ \arrow[d, "\alpha'''_Z"] \\
TY \arrow[d, "(\alpha' \circ \alpha''^{-1})_Y"'] & TR_2 \arrow[l, "T\pi'_1"] \arrow[r, "T\pi'_2"] \arrow[d, "(\alpha' \circ \alpha''^{-1})_{R_2}"] & TZ \arrow[d, "(\alpha' \circ \alpha''^{-1})_Z"] \\
TY & TR_2 \arrow[l, "T\pi'_1"] \arrow[r, "T\pi'_2"'] & TZ
\end{tikzcd} \tag{3.10}$$

Therefore R_1 is a bisimulation between $(X, \alpha_X \circ \gamma_1)$ and $(Y, \alpha'_Y \circ \gamma_2)$, and R_2 is a bisimulation between $(Y, \alpha'_Y \circ \gamma_2)$ and $(Z, (\alpha' \circ \alpha''^{-1} \circ \alpha''')_X \circ \gamma_3)$. We then have a bisimulation between $(X, \alpha_X \circ \gamma_1)$ and $(Z, (\alpha' \circ \alpha''^{-1} \circ \alpha''')_Z \circ \gamma_3)$, and so a pseudo-bisimulation between (X, γ_1) and (Z, γ_3).

Part 5 then follows from parts 2, 4 and lemma 3.2.9, as the kernel of a function can be formed as the relational composition of its graph and its converse. □

We now note a significant difference between conventional bisimulations and pseudo-bisimulations. It is essentially caused by the need to choose compatible natural automorphisms of the signature functor, when combining pseudo-bisimulations under union. This is most easily seen by noting we can simplify the weakening in the definition of pseudo-bisimulation to an asymmetric condition:

Lemma 3.2.11. *For coalgebras* $\left(\begin{array}{c} X \\ \downarrow \gamma \\ TX \end{array}\right)$ *and* $\left(\begin{array}{c} Y \\ \downarrow \xi \\ TY \end{array}\right)$ *with elements* $x \in X$ *and* $y \in Y$*, there exists a pseudo bisimulation between* x *and* y *if and only if there exists a relation* $R \subseteq X \times Y$ *such that the span given by the projections in* **Set** *lifts to a span in* T-**PseudoCoalg**(G)*:*

$$\left(\begin{array}{c} X \\ \downarrow \gamma \\ TX \end{array}\right) \xleftarrow{\pi_1} \left(\begin{array}{c} R \\ \downarrow \sigma \\ TR \end{array}\right) \xrightarrow{\pi_2} \left(\begin{array}{c} Y \\ \downarrow \xi \\ TY \end{array}\right) \tag{3.11}$$

with π_1 *a functional bisimulation as in definition 3.2.6.*

Proof. For the non trivial direction, if the projection span lifts to an arbitrary span in T-**PseudoCoalg**(G), we have a span of the following form in T-**Coalg**:

$$\left(\begin{array}{c} X \\ \downarrow \alpha_X \circ \gamma \\ TX \end{array}\right) \xleftarrow{\pi_1} \left(\begin{array}{c} R \\ \downarrow \rho \\ TR \end{array}\right) \xrightarrow{\pi_2} \left(\begin{array}{c} Y \\ \downarrow \beta_Y \circ \xi \\ TY \end{array}\right) \tag{3.12}$$

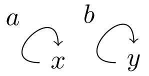
As α has an inverse, we then have a span in T-**Coalg**:

$$\left(\begin{array}{c} X \\ \downarrow \gamma \\ TX \end{array}\right) \xleftarrow{\pi_1} \left(\begin{array}{c} R \\ \downarrow \alpha_R^{-1} \circ \rho \\ TR \end{array}\right) \xrightarrow{\pi_2} \left(\begin{array}{c} Y \\ \downarrow \alpha^{-1} \circ \beta_Y \circ \xi \\ TY \end{array}\right) \tag{3.13}$$

This then gives a span of the required form in T-**PseudoCoalg**(G), if we take $\alpha_R^{-1} \circ \rho$ as the dynamics on R. ☐

Corollary 3.2.12. *In general pseudo-bisimulations are not closed under even finite unions.*

Proof. We consider the $\mathcal{P}(-)^{\{a,b\}}$-coalgebra with diagram:



(3.14)

The identity relation gives a pseudo bisimulation by lemma 3.2.10. The relation $\{(x, y), (y, x)\}$ is also a pseudo bisimulation if we consider the automorphism of the signature functor given by the non trivial permutation of the label set. The union of these relations is not a pseudo bisimulation as there is no automorphism of the signature functor that gives both states the same transition structure. □

3.3 Representing the Unitaries Coalgebraically

Definition 3.3.1. Let $\mathcal{H}$ be some Hilbert space. $\mathcal{L}(\mathcal{H})$ will denote the lattice of projection operators on $\mathcal{H}$. Following [Abramsky, 2010] we define a signature functor for modelling quantum systems as follows:

$$Q : \mathbf{Set} \to \mathbf{Set} \tag{3.15}$$

$$Q := (1 + (0,1] \times 1_{\mathbf{Set}})^{\mathcal{L}(\mathcal{H})} \tag{3.16}$$

The idea of this signature is that $\mathcal{L}(\mathcal{H})$ describes propositions we can probabilistically test on our system. In general, quantum mechanical measurements may modify the system state, and this influences how we model our tests. If a test outcome is impossible, this will be described by an outcome in the one element set. Otherwise, we give the probability of a successful outcome and the state of the system after the measurement. We note that Q is a polynomial functor, and so will preserve weak pullbacks. For a finite dimensional Hilbert space with dimension n we will write Q_n when we wish to make the dimension explicit.

We will require the following definition and technical lemma:

Definition 3.3.2. A coalgebra is said to be **strongly extensional** if bisimilarity of two states implies their equality.

Lemma 3.3.3. *Let $T : \mathbf{Set} \to \mathbf{Set}$ be an endofunctor, and $\alpha : T \Rightarrow T$ a split monomorphism. If $\left(\begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right)$ is strongly extensional, then $\left(\begin{array}{c} X \\ \downarrow\alpha_X \circ \gamma \\ TX \end{array}\right)$ is also strongly extensional.*

Proof. Let $\alpha^* : T\text{-}\mathbf{Coalg} \to T\text{-}\mathbf{Coalg}$ be the functor induced by α. We aim to show bisimilarity of states of $\left(\begin{array}{c} X \\ \downarrow\alpha_X \circ \gamma \\ TX \end{array}\right)$ implies their equality.

Assume two elements $x, x' \in X$ are bisimilar as elements of $\left(\begin{array}{c} X \\ \downarrow\alpha_X \circ \gamma \\ TX \end{array}\right)$. There is then a bisimulation R, giving a span:

$$\alpha^* \left(\begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right) \xleftarrow{\pi_1} \left(\begin{array}{c} R \\ \downarrow\rho \\ TR \end{array}\right) \xrightarrow{\pi_2} \alpha^* \left(\begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right) \tag{3.17}$$

Let $\beta : T \Rightarrow T$ be the retraction of α. We apply the induced functor β^* to the span in (3.17) giving a span:

$$\left(\begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right) \xleftarrow{\pi_1} \beta^* \left(\begin{array}{c} R \\ \downarrow\rho \\ TR \end{array}\right) \xrightarrow{\pi_2} \left(\begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right) \tag{3.18}$$

Therefore x and x' are bisimilar as elements of $\left(\begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right)$. By the assumed strong extensionality, we then have:

$$x = x' \tag{3.19}$$

□

In quantum mechanics, the pure states of a quantum system are described by rays (lines through the origin) in Hilbert space. This leads us to consider a coalgebra over a quotient of the vectors in Hilbert space.

Definition 3.3.4 (The Quantum Coalgebra)**.** For a fixed Hilbert space $\mathcal{H}$, define $\mathcal{H}_0$ as the set of all vectors in $\mathcal{H}$ with the origin removed. We define the equivalence relation on $\mathcal{H}_0$:

$$|\varphi\rangle \sim |\psi\rangle \text{ iff } \exists c \in \mathbb{C}.\, |\varphi\rangle = c|\psi\rangle \tag{3.20}$$

We then define **projective Hilbert space** $Pr(\mathcal{H})$ as the quotient $\mathcal{H}_0/\sim$. Two equivalence classes $[|\varphi\rangle]$ and $[|\psi\rangle]$ are said to be **orthogonal** if $|\varphi\rangle$ and $|\psi\rangle$ are orthogonal in Hilbert space. In this case we write:

$$[|\varphi\rangle] \perp [|\psi\rangle] \tag{3.21}$$

The **quantum coalgebra** is then defined as the Q-coalgebra $(Pr(\mathcal{H}), \gamma_q)$ describing the measurement behaviour governed by the Born rule of quantum mechanics as follows:

$$\gamma_q([|\varphi\rangle])(\hat{P}) := \begin{cases} (\frac{\langle\varphi|\hat{P}|\varphi\rangle}{\langle\varphi|\varphi\rangle}, [\hat{P}|\varphi\rangle]) & \text{if } \langle\varphi \mid \hat{P} \mid \varphi\rangle \neq 0 \\ \star & \text{otherwise} \end{cases} \tag{3.22}$$

This coalgebra is strongly extensional [Abramsky, 2010].

Having modelled measurement behaviour in the dynamics of a suitable coalgebra, we now move to state evolution of our quantum system. This is described mathematically by the unitary and (the possibly slightly less well known) antiunitary transformations of our Hilbert space. In fact, physical predictions are unaffected by multiplication by an arbitrary phase, so we shall be interested in groups with this phase quotiented out.

Definition 3.3.5. For Hilbert space $\mathcal{H}$ we will refer to the group of unitary and antiunitary operators as the **semiunitary group**. Define the **projective semiunitary group** Pr_S as the quotient of the semiunitary group identifying elements that are equal up to a phase. Similarly we define the **projective unitary group** Pr_U as the same quotient restricted to the unitaries.

Definition 3.3.6. For an object A in some category $\mathcal{C}$, the group of automorphisms of A will be denoted $\mathrm{auto}(A)$.

Lemma 3.3.7. *For Hilbert space $\mathcal{H}$, every semiunitary $\hat{U}$ induces an automorphism a_U of $\mathcal{L}(\mathcal{H})$. If two semiunitaries only differ by a phase, they induce the same automorphism. Also $a_1 = 1$ and $a_{U \circ V} = a_U \circ a_V$.*

Proof. Take the function: $\hat{P} \mapsto \hat{U}\hat{P}\hat{U}^\dagger$ with inverse $\hat{P} \mapsto \hat{U}^\dagger \hat{P}\hat{U}$ □

Corollary 3.3.8. *For Hilbert space $\mathcal{H}$, we have a group homomorphism $\alpha^{(-)} : Pr_S \to \mathrm{auto}(Q)$.*

Proof. For $[U] \in Pr_S$ take α^U as the natural isomorphism given by precomposing a_U, i.e.

$$\alpha^U_X(f)(\hat{P}) = f(\hat{U}\hat{P}\hat{U}^\dagger) \tag{3.23}$$

□

Definition 3.3.9. We will denote the image of Pr_S under $\alpha^{(-)}$ in $\mathrm{auto}(Q)$ as A_S , and the image of Pr_U under $\alpha^{(-)}$ as A_U .

We state the following version of Wigner's theorem, in the form given in [Abramsky, 2009], as it will used required for the subsequent lemma. Further details of modern projective geometry can be found in [Stubbe and van Steirteghem, 2007].

Theorem 3.3.10 (Wigner's Theorem)**.** *For Hilbert spaces $\mathcal{H}$, $\mathcal{H}'$ with $\mathcal{H}$ of dimensional at least 3, let $f : Pr(\mathcal{H}) \to Pr(\mathcal{H}')$ be a total map of projective geometries. If f satisfies:*

$$[|\varphi\rangle] \perp [|\psi\rangle] \Rightarrow f([|\varphi\rangle]) \perp f([|\psi\rangle]) \tag{3.24}$$

then there is a semilinear map $V : \mathcal{H} \to \mathcal{H}'$, with associated homomorphism $\sigma : \mathbb{C} \to \mathbb{C}$, such that:

$$f([|\varphi\rangle]) = [V|\varphi\rangle] \tag{3.25}$$

and

$$\langle V\varphi \mid V\psi\rangle = \sigma(\langle\varphi \mid \psi\rangle) \tag{3.26}$$

Further, V is unique up to a phase, and σ is either the identity or complex conjugation, so V is either linear or antilinear.

Lemma 3.3.11. *For Hilbert space $\mathcal{H}$, we have group homomorphism:*

$$h^{(-)} : Pr_S \to \mathrm{auto}(Pr(\mathcal{H})) \tag{3.27}$$

$$[|\psi\rangle] \mapsto [U|\psi\rangle] \tag{3.28}$$

Furthermore, if $\mathcal{H}$ is of dimension greater than 2, $h^{(-)}$ is injective.

Proof. That this is a well defined group homomorphism is straightforward to check. The last part follows from Wigner's theorem as stated in theorem 3.3.10, as the maps h^U induced by unitaries and antiunitaries are total maps of projective geometries [Stubbe and van Steirteghem, 2007], and they preserve orthogonality. □

These induced automorphisms of $Pr(\mathcal{H})$ lift to Q-coalgebra homomorphisms.

Proposition 3.3.12. *For $[U] \in Pr_S$, with definitions as in lemma 3.3.11 and corollary 3.3.8 α^U is a bijection, and the following diagram commutes.*

$$\begin{tikzcd}
Pr(\mathcal{H}) \arrow[r, "h^U"] \arrow[dd, "\gamma_q"'] & Pr(\mathcal{H}) \arrow[d, "\gamma_q"] \\
 & Q(Pr(\mathcal{H})) \arrow[d, "\alpha^U"] \\
Q(Pr(\mathcal{H})) \arrow[r, "Q(h^U)"'] & Q(Pr(\mathcal{H}))
\end{tikzcd} \tag{3.29}$$

Proof. A straightforward diagram chase expanding the definitions as needed. □

Remark 3.3.13. The automorphism described in proposition 3.3.12 can be seen as encoding the physicsts notion of *covariance*, if both the state and physical quantities are evolved by the same unitary our physical predictions remain as before, as captured by bisimilarity in this setting. We can also view these pseudo-coalgebra automorphisms as capturing Schrödinger evolution of the quantum system, in which states change, but physical quantities (as encoded by the coalgebra) remain fixed.

We are now in a position to give our representation results for the physically significant symmetries of a quantum system. We see that by varying the group of automorphisms of the signature functor, we can "fine tune" the automorphisms of our coalgebraic model. Our weaker notion of morphism makes the coalgebra morphisms in proposition 3.3.12 into automorphisms in the pseudo setting.

Theorem 3.3.14. *For a Hilbert space $\mathcal{H}$ with dimension greater than 2:*

- *Pr_S is fully and faithfully represented in T-$\mathbf{PseudoCoalg}(A_S)$ as the automorphisms of γ_q*
- *Pr_U is fully and faithfully represented in T-$\mathbf{PseudoCoalg}(A_U)$ as the automorphisms of γ_q*

Proof. That we have an injective group homomorphism $Pr_S \to \text{auto}(\gamma_q)$ in T-$\mathbf{PseudoCoalg}(A_S)$ follows from lemma 3.3.11 and proposition 3.3.12.

For fullness, we note that an automorphism k of γ_q must be such that the following diagram commutes, for some $U \in Pr_S$:

$$\begin{tikzcd}
Pr(\mathcal{H}) \arrow[r, "k"] \arrow[dd, "\gamma_q"'] & Pr(\mathcal{H}) \arrow[d, "\gamma_q"] \\
 & Q(Pr(\mathcal{H})) \arrow[d, "\alpha^U"] \\
Q(Pr(\mathcal{H})) \arrow[r, "Q(k)"'] & Q(Pr(\mathcal{H}))
\end{tikzcd} \tag{3.30}$$

By lemma 3.3.3, $\alpha^U \circ \gamma_q$ is strongly extensional, and so there can be at most one such k. As $k = h^U$ makes the diagram commute, it follows that every automorphism of γ_q is induced by such at U.

The case for the restriction to unitaries follows similarly. □

In fact, we can slightly sharpen the previous result:

Corollary 3.3.15. *There exists:*

- *A full and faithful functor $Pr_S \to Q$-$\mathbf{PseudoCoalg}(A_S)$*
- *A full and faithful functor $Pr_U \to Q$-$\mathbf{PseudoCoalg}(A_U)$*

Proof. We consider the case for Pr_S explicitly. The action of our functor on objects is given by $h^{(-)}$. It remains to check that every endomorphism of γ_q is an automorphism. This follows from noting in the proof of fullness in theorem 3.3.14 using strong extensionality, we did not exploit that the putative morphism k was required to be an automorphism.

Again the case for unitaries follows similarly. □

Chapter 4

Coalgebraic Logic

4.1 Introduction

In chapter 5 we will investigate the generalization of coalgebraic logic to a parameterized or fibred setting. In order to establish the background and terminology for that chapter, we now introduce two standard formulations of coalgebraic logic. Most of the material in this chapter is well known in the literature. To continue our quantum theme, we show logics and corresponding expressivity results for quantum systems in section 4.6. These logics serve as a concrete application of the material in this chapter, and set the foundations for later discussions.

4.2 Coalgebraic Logic

The Kripke frames used in the semantics of conventional modal logics are binary relations, or equivalently coalgebras of the form $X \xrightarrow{\gamma} \mathcal{P}X$. The key idea of coalgebraic modal logic is to view T-coalgebras for arbitrary endofunctor T as generalized Kripke frames, upon which we can interpret suitable modal logics. The seminal work in this area is [Moss, 1999], in which the structure of the logic is derived from the coalgebraic signature functor. This "Moss style" logic does not resemble the modal logics used in practice in mathematical logic and computer science, and so we shall not discuss it directly. We will instead consider two different approaches to coalgebraic logic:

1. A flexible and relatively concrete scheme, based on the use of natural transformations referred to as *predicate liftings* described in in [Pattinson, 2003, Schröder, 2008]. Background on this style of coalgebraic logic will be provided in section 4.3.

2. A more abstract formulation based on dual adjunctions and suitable natural transformations referred to as *logical connections*, connecting coalgebras to the algebraic semantics for a corresponding logic. This material follows ideas in [Abramsky, 2011, Jacobs and Sokolova, 2010, Kurz and Rosický, 2012, Klin, 2007, Bonsangue and Kurz, 2005, Kurz and Velebil, 2013]. Background on this style of coalgebraic logic will be provided in section 4.4.

The relationship between predicate liftings and logical connections will be discussed in section 4.5.

We now introduce another notion of equivalence of states, emphasized in work on coalgebraic logic:

Definition 4.2.1 (Behavioural Equivalence)**.** Let T : **Set** $\to$ **Set** be an endofunctor, and $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$ and $\begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix}$ T-coalgebras. Elements $x \in X$ and $y \in Y$ are **behaviourally equivalent** [Kurz, 2000] if there exists a cospan:

$$\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix} \xrightarrow{f} \begin{pmatrix} E \\ \downarrow\gamma_E \\ TE \end{pmatrix} \xleftarrow{g} \begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix} \tag{4.1}$$

in T-**Coalg** such that $f\,x = g\,y$. We write $x \sim y$ for x is behaviourally equivalent to y.

Bisimilarity and behavioural equivalence often coincide, in particular for endofunctors that preserve weak pullbacks. When the two notions do diverge, behavioural equivalence has been argued to be the more natural notion, see for example the discussion in [Pattinson, 2004].

4.3 Predicate Liftings

Definition 4.3.1 (Syntax)**.** A **modal signature** is a family of **modality symbols** $(\heartsuit)_{i\in I}$ each with a specified cardinality giving its **arity**. For a given modal signature Λ, set of proposition variables Prop, and regular cardinal κ we specify formulae for the language $\mathcal{L}^{\kappa}_{\mathrm{Prop}}(\Lambda)$ by the following grammar:

$$\begin{aligned}
\varphi &:= \neg\varphi && && (4.2)\\
&\mid \bigwedge_{i\in I}\varphi_i && \mathrm{card}(I) < \kappa && (4.3)\\
&\mid \heartsuit(\varphi_a)_{a\in\alpha} && \heartsuit \in \Lambda \text{ and } \heartsuit \text{ has arity } \alpha && (4.4)\\
&\mid p && p \in \mathrm{Prop} && (4.5)
\end{aligned}$$

Definition 4.3.2 (Pattinson Predicate Lifting)**.** Let $T : \mathbf{Set} \to \mathbf{Set}$ be a **Set** endofunctor. A (unary) **predicate lifting** [Pattinson, 2003] for T is a natural transformation:

$$[\![\Box]\!] : U \circ 2^{(-)} \Rightarrow U \circ 2^{T^{op}(-)} \tag{4.6}$$

More generally for a set I, an I-ary predicate lifting is a natural transformation:

$$[\![\heartsuit]\!] : \prod_{i\in I}(U \circ 2^{(-)}) \Rightarrow U \circ 2^{T^{op}(-)} \tag{4.7}$$

where U is the forgetful functor $\mathbf{Bool} \to \mathbf{Set}$.

Definition 4.3.3 (Valuation)**.** For a fixed set of proposition variables Prop and T-coalgebra $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$ a **valuation** on γ is a morphism $\mathrm{v} : \mathrm{Prop} \to U \circ 2^X$, where again U is the forgetful functor $\mathbf{Bool} \to \mathbf{Set}$.

Before proceeding to define the semantics of logics based on predicate lifting, we introduce two basic notions relating logics for coalgebras to the underlying sets of states.

Definition 4.3.4. A coalgebraic logic is said to be:

- **Adequate** if behaviourally equivalent elements satisfy the same formulae.
- **Expressive** if elements satisfying the same formulae are behaviourally equivalent.

Remark 4.3.5. The notions of behavioural equivalence, adequacy and expressivity can be generalized greatly from the **Set** coalgebra based definitions we have given, but they will be sufficient for our examples such as those in section 4.6.

Definition 4.3.6 (Semantics)**.** For a T-coalgebra (X, γ), valuation v and specified predicate liftings, the semantics of coalgebraic modal logic is specified inductively as follows:

$$\llbracket \neg\varphi \rrbracket_{(X,\gamma),\mathrm{v}} := X \setminus \llbracket \varphi \rrbracket_{(X,\gamma),\mathrm{v}} \tag{4.8}$$

$$\llbracket \bigwedge_{i\in I} \varphi_i \rrbracket_{(X,\gamma),\mathrm{v}} := \bigcap_{i\in I} \llbracket \varphi_i \rrbracket_{(X,\gamma),\mathrm{v}} \tag{4.9}$$

$$\llbracket \heartsuit(\varphi_a)_{a\in A} \rrbracket_{(X,\gamma),\mathrm{v}} := \gamma^{-1} \circ \llbracket \heartsuit \rrbracket_X \left(\llbracket \varphi_a \rrbracket_{X,\gamma,\mathrm{v}} \right)_{a\in A} \tag{4.10}$$

$$\llbracket p \rrbracket_{(X,\gamma),\mathrm{v}} := \mathrm{v}\, p \tag{4.11}$$

With these semantics, the language $\mathcal{L}^{\kappa}_{\emptyset}(\Lambda)$ of definition 4.3.1 is adequate. As is standard, we define:

$$x \models_{\gamma,\mathrm{v}} \varphi \quad := \quad x \in \llbracket \varphi \rrbracket_{\gamma,\mathrm{v}} \tag{4.12}$$

We also relate some special classes of modalities and predicate liftings, following [Schröder, 2008].

Definition 4.3.7. A unary modality $\Box$ is said to be:

- **Monotone** if:

$$\Box(\varphi \wedge \psi) \rightarrow \Box\varphi \tag{4.13}$$

- **α-normal**, for a regular cardinal α, if for $\mathrm{card}(I) < \alpha$:

$$\bigwedge_{i\in I} \Box\varphi_i \leftrightarrow \Box \bigwedge_{i\in I} \varphi_i \tag{4.14}$$

A unary predicate lifting $\llbracket \Box \rrbracket$ is said to be:

- **Monotone** if:

$$U \subseteq V \subseteq X \Rightarrow \llbracket \Box \rrbracket_X(U) \subseteq \llbracket \Box \rrbracket_X(V) \tag{4.15}$$

- **Continuous** if:

$$\bigcap_{i\in I} \llbracket \Box \rrbracket_X(U_i) = \llbracket \Box \rrbracket_X \left(\bigcap_{i\in I} U_i \right) \tag{4.16}$$

We then have the following facts:

- A modality with semantics given by a monotone predicate lifting is monotone
- A modality with semantics given by a continuous predicate lifting is α-normal for all regular cardinals α

We will also require a technical property of signature functors in our later discussions:

Definition 4.3.8. A functor of type **Set** $\to$ **Set** is said to be **κ-accessible**, for regular cardinal κ, if it preserves κ-directed colimits. A functor is said to be **accessible** if it is κ-accessible for some κ.

Remark 4.3.9. Accessible functors can be defined in much greater generality [Makkai and Paré, 1989, Adámek and Rosický, 1994], but we restrict our attention to the **Set** endofunctors that we will require in later sections.

Remark 4.3.10. ω-accessible functors are often referred to as **finitary functors**.

We have not described κ-directed colimits and their preservation in detail. Instead, we will exploit the following standard properties of accessible **Set** endofunctors:

Proposition 4.3.11 ([Adámek and Rosický, 1994], [Adámek et al.])**.** *Accessible* **Set** *endofunctors have the following properties:*

- *Constant functors are finitary*
- *For a set A, with cardinality less than λ, the functor $(-)^A$ is λ-accessible*
- *The powerset functor is* not *λ-accessible for any λ*
- *The* **λ-bounded powerset functor***, taking a set to its subsets of cardinality less than λ, is λ-accessible*

λ-accessible **Set** *endofunctors are closed under:*

- *Products*
- *Coproducts*
- *Subfunctors*
- *Quotient Functors*
- *Composition*

4.4 Logical Connections

For a category $\mathcal{C}$ and endofunctor $T : \mathcal{C} \to \mathcal{C}$ there is an obvious isomorphism:

$$T\text{-}\mathbf{Alg} \cong T^{op}\text{-}\mathbf{Coalg}^{op} \tag{4.17}$$

This basic observation suggests a starting point for constructing the algebraic semantics for logics corresponding to T-coalgebras. A straightforward generalization of the above situation is to introduce a dual equivalence such as Stone duality, and consider functors:

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "\mathrm{Pt}"] \arrow[r, phantom, "\simeq"] & \mathcal{X}^{op} \arrow[l, bend left, "\Omega"] \arrow[loop right, "T^{op}"] \end{tikzcd} \tag{4.18}$$

If we wish to define a suitable endofunctor for a corresponding algebraic semantics, the obvious choice is to define:

$$L := \Omega \circ T^{op} \circ \mathrm{Pt} \tag{4.19}$$

We then have an equivalence:

$$L\text{-}\mathbf{Alg} \simeq T\text{-}\mathbf{Coalg}^{op} \tag{4.20}$$

and a natural isomorphism:

$$L \circ \Omega \overset{\cong}{\Rightarrow} \Omega \circ T^{op} \tag{4.21}$$

A logical connection generalizes this situation in two directions:

1. We allow arbitrary endofunctors $L : \mathcal{A} \to \mathcal{A}$ such that there exists a natural transformation of type $L \circ \Omega \Rightarrow T^{op} \circ \Omega$, generalizing the isomorphism in (4.21). This provides flexibility to tailor the form of the logic independently of the structure of the coalgebraic signature functor.

2. We consider arbitrary adjunctions rather than simply dual equivalences. This can ease the task of finding concrete presentations for the abstract logics under consideration, and broadens the scope of applicability of the theory, see [Kurz and Rosický, 2012, Klin, 2007] for further discussion.

Definition 4.4.1 (Logical Connection)**.** Given a dual adjunction:

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "\mathrm{Pt}"] \arrow[r, phantom, "\perp"] & \mathcal{X}^{op} \arrow[l, bend left, "\Omega"] \end{tikzcd} \tag{4.22}$$

a **logical connection** consists of the following data:

- A coalgebraic signature endofunctor $T : \mathcal{X} \to \mathcal{X}$
- A logical signature endofunctor $L : \mathcal{A} \to \mathcal{A}$
- A natural transformation:

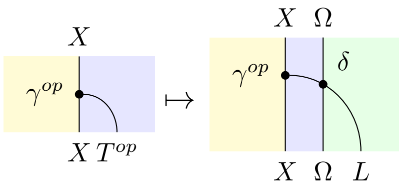
$$\delta : L \circ \Omega \Rightarrow \Omega \circ T^{op} \tag{4.23}$$

Where no confusion is likely, we shall refer to δ as a logical connection, leaving the additional data implicit.

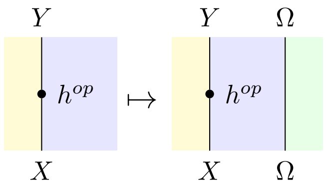
Remark 4.4.2. Typically in definition 4.4.1, the category $\mathcal{X}$ will be some "spatial" category such as **Set** or **Stone**. The category $\mathcal{A}$ will be an "algebraic" category, carrying structure providing models of some form of propositional logic, for example **Bool** or **MSLat**.

Definition 4.4.3 (Algebraization Functor)**.** A logical connection δ as described in definition 4.4.1 induces an **algebraization functor**, given graphically as follows:

$$\mathbf{Alg}^{\delta} : T\text{-}\mathbf{Coalg} \to L\text{-}\mathbf{Alg} \tag{4.24}$$



(4.25)



(4.26)

From the string diagram we see that δ serves as a type of distributive law, allowing the construction of an L-algebra from a T-coalgebra by lifting the functor Ω to the (co)algebraic level.

Now that we can construct an algebra from a coalgebra, we can consider semantics for our logic on the algebraic side of our adjunction. In the presence of initial algebras, we can describe the semantics directly as follows:

Definition 4.4.4 (Initial Algebra Logical Semantics)**.** Assuming that L has an initial algebra $\begin{pmatrix} L\,\mathrm{Form}_L \\ \downarrow in_L \\ \mathrm{Form}_L \end{pmatrix}$ we can regard Form_L as representing (equivalence classes of) formulae in our logic. We define the **semantics morphism** for a given coalgebra $\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix}$ as the unique L-**Alg** morphism:

$$\begin{pmatrix} L\,\mathrm{Form}_L \\ \downarrow in_L \\ \mathrm{Form}_L \end{pmatrix} \xrightarrow{[\![-]\!]_{(X,\gamma)}} \mathbf{Alg}^{\delta} \begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \tag{4.27}$$

More generally we can define a stratified form of the semantics, by considering the initial sequence of our logical signature functor:

Definition 4.4.5 (Initial Sequence Logical Semantics)**.** We define the semantics of our logic inductively by considering the initial sequence [Adámek, 1974, Adámek and Koubek, 1979, Adámek and Trnková, 1990]:

$$\begin{tikzcd}
0 \arrow[r, "!"] \arrow[d, "{[\![-]\!]^0_\gamma}"'] & L0 \arrow[r, "L!"] \arrow[d, "{L[\![-]\!]^0_\gamma}"'] & L^2 0 \arrow[r, "L^2!"] \arrow[d, "{L^2[\![-]\!]^0_\gamma}"'] & \cdots \\
\Omega X & L\,\Omega(X) \arrow[l, "\mathbf{Alg}^{\delta}(\gamma)"] & L^2\,\Omega(X) \arrow[l, "L\mathbf{Alg}^{\delta}(\gamma)"] & \cdots \arrow[l, "L^2\mathbf{Alg}^{\delta}(\gamma)"]
\end{tikzcd} \tag{4.28}$$

For natural number n and T-coalgebra $\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix}$, the **depth** n **semantics morphism**,$[\![-]\!]^n_{(X,\gamma)}$, is defined inductively as follow:

$$[\![-]\!]^0_{(X,\gamma)} := 0 \xrightarrow{!} \Omega(X) \tag{4.29}$$

$$[\![-]\!]^{n+1}_{(X,\gamma)} := \mathbf{Alg}^{\delta}(\gamma) \circ L[\![-]\!]^n_{(X,\gamma)} \tag{4.30}$$

Both these semantics can be extended to support proposition variables, by moving to free algebras over the proposition variables, or the sequence used in construction of free algebras. We avoid giving details at this point, as we will introduce them in sections 5.7.1.2 and 5.7.1.4 when we generalize to the parameterized setting.

4.5 Relating Predicate Liftings and Logical Connections

We consider the adjunction:

$$\begin{tikzcd}\mathbf{Bool} \arrow[r, bend left, "F_u"] \arrow[r, phantom, "\perp"] & \mathbf{Set}^{op} \arrow[l, bend left, "2^{(-)}"]\end{tikzcd} \tag{4.31}$$

where F_u is the functor that takes a Boolean algebra to its set of ultrafilters and homomorphisms to their inverse image maps. Let $F \dashv U$ be the free / forgetful adjunction for Boolean algebras. Given a unary predicate lifting:

$$[\![\Box]\!] : U \circ 2^{(-)} \Rightarrow U \circ 2^{T^{op}(-)} \tag{4.32}$$

Such a natural transformation corresponds directly under the adjunction $F \dashv U$ to a logical connection:

$$\delta : F \circ U \circ 2^{(-)} \Rightarrow 2^{T^{op}(-)} \tag{4.33}$$

More generally, given a family of predicate liftings $([\![\heartsuit_j]\!])_{j \in J}$ with corresponding arities I_j, noting that the functor category $[\mathbf{Set}^{op}, \mathbf{Set}]$ has coproducts (in fact all small limits and colimits), given pointwise, we can construct the natural transformation:

$$[[\![\heartsuit_j]\!] \mid j \in J] : \left(\coprod_{j \in J} \prod_{i \in I_j} U \right) \circ 2^{(-)} \Rightarrow U \circ 2^{T^{op}(-)} \tag{4.34}$$

and as before, under the adjunction this corresponds to a logical connection of type:

$$F \circ \left(\coprod_{j \in J} \prod_{i \in I_j} U \right) \circ 2^{(-)} \Rightarrow U \circ 2^{T^{op}(-)} \tag{4.35}$$

Using these logical connections we recover the same semantics as those given in definition 4.3.6 for the original predicate liftings. We consider both schemes as they capture different aspects of coalgebraic logic:

- Predicate liftings are easy to reason about, with a simple inductively defined semantics. When constructing practical logics they are a natural choice.
- Logical connections explicitly capture the duality theoretic nature of coalgebraic logic. They allow us to manipulate logics as totalities rather than families of predicate liftings, and are often the right choice for theoretical work.

4.6 Expressive Logics for Quantum Systems

In this section we consider coalgebraic modal logics for quantum systems, based on two different signature functors. We first examine the Boolean test based signature functor of [Abramsky, 2010], used in the representation result of chapter 3. We then introduce a new signature functor, based on finite probability distributions over measurement outcomes, that may be more practical for realistic problems. In both cases we describe a suitable basic set of predicate liftings, and show that the resulting logic is expressive. The technical results in this section apply expressivity criteria described in [Schröder, 2008].

4.6.1 A Boolean Test Based Signature

We first consider modal logics for the signature functor Q of definition 3.3.1, using the predicate lifting approach. We fix a background finite dimensional Hilbert space $\mathcal{H}$ throughout this section. Firstly we note a technical property of Q:

Lemma 4.6.1. *For Hilbert space $\mathcal{H}$, the functor Q given in definition 3.3.1 is κ-accessible where κ is the successor cardinal of* $\mathrm{card}(\mathcal{L}(\mathcal{H}))$*.*

Proof. We apply some properties of accessible functors, using proposition 4.3.11. The constant functors 1 and $(0,1]$ are finitary. As λ-accessible functors are closed under products and coproducts, $1 + (0,1] \times (-)$ is finitary. As $\mathcal{L}(\mathcal{H})$ has cardinality less than κ, the functor $(-)^{\mathcal{L}(\mathcal{H})}$ is κ-accessible. Finally, as κ-accessible functors are closed under composition, the claim follows. $\square$

We now define a family of predicate liftings:

Definition 4.6.2. For $\hat{P} \in \mathcal{L}(\mathcal{H})$ and rational $0 < q \leq 1$ we define unary predicate lifting:

$$[\![\Box_{\hat{P},q}]\!]_X(U) := \{f \mid \exists u \in U, r > q. f\hat{P} = (r,u)\} \tag{4.36}$$

To check naturality, for arbitrary $\varphi : X \to Y$, we consider the commutativity of the following diagram:

$$\begin{tikzcd}
2^Y \arrow[r, "\llbracket \Box_{\hat{Q},q} \rrbracket_X"] & 2^{QX} \\
2^X \arrow[u, "\varphi^{-1}"] \arrow[r, "\llbracket \Box_{\hat{Q},q} \rrbracket_Y"'] & 2^{QY} \arrow[u, "(Q\varphi)^{-1}"']
\end{tikzcd} \tag{4.37}$$

That this commutes can be seen as follows:

$$\begin{aligned}
& f \in (Q\varphi)^{-1} \circ \llbracket \Box_{\hat{Q},q} \rrbracket_Y(V) \\
\Leftrightarrow \quad & \{ \text{ definition of } (Q\varphi)^{-1} \ \} \\
& \exists g. g \in \llbracket \Box_{\hat{Q},q} \rrbracket_Y(V) \wedge g = (1 + (0,1] \times \varphi) \circ f \\
\Leftrightarrow \quad & \{ \text{ definition of } \llbracket \Box_{\hat{P},q} \rrbracket \text{, inverse images } \} \\
& f\hat{P} \in \varphi^{-1}(V) \\
\Leftrightarrow \quad & \{ \text{ definition of } \llbracket \Box_{\hat{P},q} \rrbracket \ \} \\
& f \in \llbracket \Box_{\hat{P},q} \rrbracket_X \circ \varphi^{-1}(V)
\end{aligned}$$

Theorem 4.6.3. *The predicate liftings of definition 4.6.2:*

1. *Are monotone*
2. *Constitute a separating set of liftings*

Proof. We first check monotonicity as described in definition 4.3.7. Assume $U \subseteq V \subseteq X$ and $f \in \llbracket \Box_{\hat{P},q} \rrbracket_X(U)$. There then exists $r > q$ and $u \in U$ such that:

$$f(\hat{P}) = (r, u) \tag{4.38}$$

This immediately implies that $f \in \llbracket \Box_{\hat{P},q} \rrbracket_X(V)$ as $U \subseteq V$ by assumption.

For the second part, we proceed by case analysis, showing that for arbitrary set X elements of the set QX can be separated by our predicate liftings. Firstly assume two states $*$ and (r, u). By density of the rationals in the reals, there exists $0 < q \leq r \leq 1$ and so the set:

$$\llbracket \Box_{\hat{P},q} \rrbracket_X(X) \tag{4.39}$$

contains (r, u) but does not contain $*$.

Now we assume two distinct states (r, u) and (r', u'). Firstly if $r \neq r'$ then without loss of generality we can assume $r < r'$, and there must exist a rational $0 < r < q' < r' \leq 1$, again by density of the rationals in the reals. The set:

$$[\![\Box_{\hat{P},q'}]\!]_X(X) \tag{4.40}$$

then contains (r', u') but not (r, u). Finally if $r = r'$ we must have $u \neq u'$. Again by density of the rationals in the reals, there must exist $0 < q'' < r$. The set:

$$[\![\Box_{\hat{P},q''}]\!]_X(\{u\}) \tag{4.41}$$

then contains (r, u) but not (r, u'). □

Corollary 4.6.4. *Let κ be the successor cardinal of* $\text{card}(\mathcal{L}(\mathcal{H}))$*. Any coalgebraic modal logic for the signature functor Q, with conjunctions of cardinality at least κ, that contains at least unary modalities given by the predicate liftings in definition 4.6.2, is expressive. Furthermore these modalities are monotone.*

Proof. By lemma 4.6.1 we can apply [Schröder, 2008, Theorem 14], and so our logic will be expressive. Applying [Schröder, 2008, Theorems 27] gives monotonicity. □

Counterexample 4.6.5. The predicate liftings of definition 4.6.2 are not continuous. We can see this by considering the intersection of any empty family of subsets. We note that 2^{QX} contains functions f such that $f(\hat{P}) = *$, mapping into the left component of the coproduct. On the other hand, by definition, $[\![\Box_{\hat{Q},q}]\!]_X(X)$ does not. Logically, this means that the axiom:

$$\Box_{\hat{P},q}\top \leftrightarrow \top \tag{4.42}$$

does not hold.

These predicate liftings are very close to being continuous though. If we consider a non-empty family $(U_i \subseteq X)_{i \in I}$, we have:

$$\begin{aligned}
& f \in [\![\Box_{\hat{P},q}]\!]_X\left(\bigcap_i U_i\right) \\
\Leftrightarrow \quad & \{ \text{ definitions } \} \\
& \exists r, u. f\hat{P} = (r, u) \wedge r > q \wedge u \in \bigcap_i U_i
\end{aligned}$$

$$
\begin{aligned}
\Leftrightarrow\quad & \{\ \text{intersections}\ \} \\
& \exists r, u. f\hat{P} = (r, u) \wedge r > q \wedge \forall i. u \in U_i \\
\Leftrightarrow\quad & \{\ \text{definitions, family non-trivial}\ \} \\
& f \in \bigcap_i [\![\Box_{\hat{P},q}]\!]_X(U_i)
\end{aligned}
$$

Logically, this can be interpreted as the modality $\Box_{\hat{P},q}$ commuting with non-trivial conjunctions.

4.6.2 A Distribution Based Signature

Logics built on the signature functor Q are expressive, but their heavy emphasis on positive outcomes of Boolean tests makes them inconvenient to work with when reasoning about quantum protocols. A more practical logic for that type of application would give convenient access to the distribution of measurement outcomes. We will now construct an alternative coalgebraic model of quantum systems and an accompanying expressive modal logic.

Definition 4.6.6. Let $D : \mathbf{Set} \to \mathbf{Set}$ denote the **finite distribution functor**, defined on objects as follows:

$$D(X) := \left\{ d : X \to [0,1] \mid d \text{ has finite support and } \sum_{x \in X} d(x) = 1 \right\} \tag{4.43}$$

and on morphisms:

$$D(f : X \to Y)(d \in D(X))(y \in Y) := \sum_{x \in X. f(x) = y} d(x) \tag{4.44}$$

Remark 4.6.7. D is actually a monad in an obvious way, but we shall not require that additional structure in later sections.

We will require the following technical result:

Lemma 4.6.8. *The finite distribution functor D is ω-accessible.*

Proof. Let X be an arbitrary set, and consider a distribution $d \in DX$. Let V be the support of d, which is a finite set by the definition of D. We write $d|_V$ for the distribution given by restricting d to V. For the inclusion $i : V \hookrightarrow X$, we note that:

$$D(i)(d|_{V'}) = d \tag{4.45}$$

The result follows from proposition 5.2 in [Adámek and Porst, 2004]. $\square$

For the test based signature Q we were interested in the success probabilities of projective measurements. For our alternative distribution based signature we are interested in probability distributions over measurement outcomes, when measuring physical quantities of our system. Physical quantities are represented mathematically by self adjoint operators. We exploit the eigenvalues to distinguish the measurement outcomes.

Definition 4.6.9. For finite dimensional Hilbert space $\mathcal{H}$, let $\mathcal{A}_{\mathcal{H}}$ denote the set of self adjoint operators on $\mathcal{H}$. Define the **distribution based quantum signature functor** Q^d as follows:

$$Q^d := D(\mathbb{R} \times (-))^{\mathcal{A}_{\mathcal{H}}} \tag{4.46}$$

There is an obvious **quantum coalgebra** for this signature, mapping pure states to distributions over measurement outcomes and subsequent states.

The following technical property of the distribution based quantum signature functor is important for the analysis of expressivity:

Lemma 4.6.10. *For a finite dimensional Hilbert space $\mathcal{H}$, the functor Q^d is accessible.*

Proof. We can describe our functor as the composite:

$$Q^d = (-)^{\mathcal{A}_{\mathcal{H}}} \circ D \circ (\mathbb{R} \times (-)) \tag{4.47}$$

Polynomial functors with exponents of cardinality less than λ are λ-accessible. As accessible functors are closed under composition, the claim then follows from lemma 4.6.8. $\square$

We now define our family of predicate liftings:

Definition 4.6.11. For $\hat{A} \in \mathcal{A}_{\mathcal{H}}$, $\lambda \in \mathbb{R}$ and rational $0 < q \leq 1$ we define unary predicate lifting:

$$[\![\Box_{\hat{A},\lambda,q}]\!]_X(U) := \left\{ d \mid \sum_{u \in U} d(\hat{A})(\lambda, u) > q \right\} \tag{4.48}$$

Naturality is confirmed in lemma 4.6.12 below.

Lemma 4.6.12. *The functions $[\![\Box_{\hat{A},\lambda,q}]\!]_X$ of definition 4.6.11 are natural in X.*

Proof. We aim to establish naturality by constructing $[\![\Box_{\hat{A},\lambda,q}]\!]_X$ from simpler components. To simplify notation, we will write $2 : \mathbf{Set}^{op} \to \mathbf{Set}$ for the contravariant powerset functor in this proof. Firstly, for rational $q \in \mathbb{Q}$, we define the functions:

$$\alpha^q_X : 2^X \to 2^{DX} \tag{4.49}$$

$$U \mapsto \left\{ d \mid \sum_{u \in U} d(u) > q \right\} \tag{4.50}$$

It is easy to confirm these functions are natural in X as:

$$\begin{aligned}
& d \in \alpha^q_X \circ \varphi^{-1}(V) \\
\Leftrightarrow \quad & \{ \text{ definition of } \alpha^q_X \ \} \\
& \sum_{u \in \varphi^{-1}V} d(u) > q \\
\Leftrightarrow \quad & \{ \text{ definition of } D \ \} \\
& \sum_{v \in V} D(\varphi)(d)(v) > q \\
\Leftrightarrow \quad & \{ \text{ definition of } \alpha^q_Y \ \} \\
& D(\varphi)(d) \in \alpha^q_Y(V) \\
\Leftrightarrow \quad & \{ \text{ inverse images } \} \\
& d \in D(\varphi)^{-1} \circ \alpha^q_Y(V)
\end{aligned}$$

We also require a second simple family of functions:

$$\beta^\lambda_X : 2^X \to 2^{\mathbb{R} \times X} \tag{4.51}$$

$$U \mapsto \{(\lambda, u) \mid u \in U\} \tag{4.52}$$

Again, it is easy to confirm that the β^λ_X are natural in X:

$$\begin{aligned}
& (r, u) \in \beta^\lambda_X \circ \varphi^{-1}(V) \\
\Leftrightarrow \quad & \{ \text{ definition of } \beta^\lambda_X \ \} \\
& r = \lambda \wedge u \in \varphi^{-1}(V) \\
\Leftrightarrow \quad & \{ \text{ inverse images } \} \\
& r = \lambda \wedge \varphi(u) \in V
\end{aligned}$$

$$\begin{aligned}
\Leftrightarrow \quad & \{ \text{ definition of } \beta^{\lambda}_Y \; \} \\
& (r, \varphi(u)) \in \beta^{\lambda}_Y(V) \\
\Leftrightarrow \quad & \{ \text{ inverse images } \} \\
& (r, u) \in (1 \times \varphi)^{-1} \circ \beta^{\lambda}_Y(V)
\end{aligned}$$

Finally we require a third family of functions:

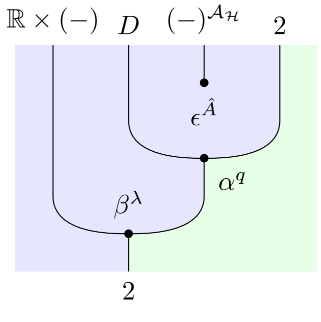
$$\epsilon^{\hat{A}}_X : X^{\mathcal{A}_{\mathcal{H}}} \to X \tag{4.53}$$

$$f \mapsto f(\hat{A}) \tag{4.54}$$

We confirm $\epsilon^{\hat{A}}_X$ is natural in X:

$$\begin{aligned}
& \varphi \circ \epsilon^{\hat{A}}(f) \\
= \quad & \{ \text{ definitions } \} \\
& \varphi(f(\hat{A})) \\
= \quad & \{ \text{ definitions } \} \\
& \epsilon^{\hat{A}}_Y(\varphi \circ f) \\
= \quad & \{ \text{ definitions } \} \\
& \epsilon^{\hat{A}} \circ (\varphi \circ (-))(f)
\end{aligned}$$

We consider the composite natural transformation of the string diagram:



(4.55)

Note that the orientation of $\epsilon^{\hat{A}}$ is reversed as that part of the diagram is in the opposite category. We then calculate:

$$((2 * \epsilon^{\hat{A}} * D(\mathbb{R} \times (-))) \circ (\alpha^q * (\mathbb{R} \times (-))) \circ \beta^{\lambda})_X(U)$$

$$
\begin{aligned}
&= \quad \{ \text{ definitions } \}\\
&\quad (\epsilon^{\hat{A}}_{D(\mathbb{R}\times X)})^{-1} \circ \alpha^{q}_{\mathbb{R}\times X} \circ \beta^{\lambda}_{X}(U)\\
&= \quad \{ \text{ definitions } \}\\
&\quad (\epsilon^{\hat{A}}_{D(\mathbb{R}\times X)})^{-1} \circ \alpha^{q}_{\mathbb{R}\times X}\{(\lambda, u) \mid u \in U\}\\
&= \quad \{ \text{ definitions } \}\\
&\quad (\epsilon^{\hat{A}}_{D(\mathbb{R}\times X)})^{-1} \left(\left\{ d \mid \sum_{u\in U} (\lambda, u) > q \right\} \right)\\
&= \quad \{ \text{ definitions, inverse images } \}\\
&\quad \left\{ f \mid \sum_{u\in U} f(\hat{A})(\lambda, u) > q \right\}\\
&= \quad \{ \text{ definitions } \}\\
&\quad [\![\Box_{\hat{A},\lambda,q}]\!]_X(U)
\end{aligned}
$$

Therefore the components $[\![\Box\hat{A}, \lambda, q]\!]_X$ are natural in X as claimed. □

Remark 4.6.13. The proof of lemma 4.6.12 is clearly not the most concise form possible. When signatures and predicate liftings become complex, naturality can be "obvious", but may involve awkward calculations to establish. We suggest that string diagram notation encourages building liftings by composition of simpler predicate liftings and other natural transformations, by clarifying how they can be composed. Most of our proof was devoted to explicitly establishing naturality of some simple components. For practical usage, a well known catalog of natural transformations could be assumed to be available, obviating the need for explicit checks. Work on such a catalog can be found in the appendix of [Bartels, 2004].

Theorem 4.6.14. *The predicate liftings of definition 4.6.11:*

1. *Are monotone*
2. *Constitute a separating set of predicate liftings*

Proof. We first check monotonicity as described in definition 4.3.7. Assume $U \subseteq V \subseteq X$ and $d \in [\![\Box_{\hat{A},\lambda,q}]\!]_X(U)$. We then have:

$$\sum_{u\in U} d(\hat{A})(\lambda, u) > q \tag{4.56}$$

Probability values are positive, and by assumption $U \subseteq V$. Therefore, we have:

$$\sum_{v \in V} d(\hat{A})(\lambda, v) \geq \sum_{u \in U} d(\hat{A})(\lambda, u) > q \tag{4.57}$$

It then follows that $d \in [\![\Box_{\hat{A},\lambda,q}]\!]_X(V)$, and so the lifting is monotone.

To see that our liftings are separating, consider distinct:

$$d_1, d_2 \in Q^d X \tag{4.58}$$

There must then exist $\hat{A} \in \mathcal{A}_{\mathcal{H}}$ such that:

$$d_1(\hat{A}) \neq d_2(\hat{A}) \tag{4.59}$$

There must then be $\lambda \in \mathbb{R}$ and $u \in X$ such that:

$$d_1(\hat{A})(\lambda, u) \neq d_2(\hat{A})(\lambda, u) \tag{4.60}$$

Without loss of generality, using the density of the rationals in the reals, we can assume there exists rational q such that:

$$d_1(\hat{A})(\lambda, u) < q < d_2(\hat{A})(\lambda, u) \tag{4.61}$$

It then follows that the set:

$$[\![\Box_{\hat{A},\lambda,q}]\!]_X(\{u\}) \tag{4.62}$$

contains d_2 but does not contain d_1, showing our liftings are separating. □

Corollary 4.6.15. *A coalgebraic modal logic, with conjunctions of sufficient cardinality, containing modalities given by the predicate liftings of definition 4.6.11, is expressive. Furthermore these modalities are monotone.*

Proof. By lemma 4.6.10, Q^d is accessible and therefore we can apply [Schröder, 2008, Theorem 14] to show expressivity. Monotonicity follows from [Schröder, 2008, Theorem 27]. □

Chapter 5

Fibred Coalgebraic Logic

5.1 Introduction

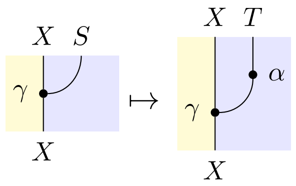
The paper [Abramsky, 2010], discussed in chapter 3, introduced a fibration of coalgebras to develop its main representation result. That paper then asks what a suitable "fibred coalgebraic logic" would look like. That question is the starting point for the work in this chapter.

We begin by investigating some quantum mechanical examples. It is shown that physically relevant features such as state evolution and restriction to subsystems give rise functors between categories of coalgebras. These functors then induce modalities in the corresponding coalgebraic logics. From this starting point, we consider the functoriality of further logical examples. Abstracting from these examples, we introduce *model transformations*. They provide a lightweight framework for extending predicate lifting style coalgebraic logic with modalities induced by suitable functors between model categories.

As a second step in our study of fibred coalgebraic logics, we consider a parameterized generalization of the logical connections introduced in chapter 4. We adopt an explicitly fibrational approach, and show that the duality theoretic mathematical structures extend smoothly to the fibred setting. We also consider the notion of an *institution condition*, establishing a contravariant harmony between transformations of the formulae and models of a logic. It is then shown that under several variations of assumptions about the semantics of our logics, appropriate institution conditions hold.

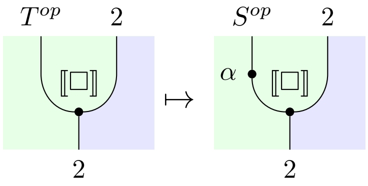
5.2 Motivating Examples

Given two signature functors $S, T : \mathbf{Set} \to \mathbf{Set}$ and a natural transformation $\alpha : S \Rightarrow T$, dually to example 2.1.1, the mapping:



(5.1)

extends to a functor S-**Coalg** $\to$ T-**Coalg**. The natural transformation α also induces a mapping from predicate liftings for T to predicate liftings for S as follows:



(5.2)

We note the difference in direction for the coalgebra (model) and predicate lifting (logic) transformations. These two operations have practical interpretations for reasoning about quantum systems. Before discussing some examples, the following lemma will be helpful:

Lemma 5.2.1. *Let $\mathcal{C}$ be a cartesian closed category, and $T : \mathcal{C} \to \mathcal{C}$ an endofunctor. Every $\mathcal{C}$-morphism $f : A \to B$ induces a natural transformation:*

$$f^* : T^B \Rightarrow T^A$$

Proof. Exponentials are bifunctorial:

$$\mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C} \tag{5.3}$$

precomposing this functor with the product functor:

$$1 \times T \tag{5.4}$$

gives a bifunctor:

$$S : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C} \tag{5.5}$$

If A is a $\mathcal{C}$-object, fixing the first argument of S to A we have:

$$S(A, -) = T^A \tag{5.6}$$

As discussed in section 2.7, every $\mathcal{C}$-morphism $f : A \to B$ then induces a natural transformation

$$f^* : T^B \Rightarrow T^A \tag{5.7}$$

where the reversal of direction is due to the contravariance of S in its first argument. □

Corollary 5.2.2. *Let $\mathcal{C}$ be a cartesian closed category, and $T : \mathcal{C} \to \mathcal{C}$ an endofunctor. Every $\mathcal{C}$-morphism $f : A \to B$ induces a functor of type:*

$$T^B\text{-}\mathbf{Coalg} \to T^A\text{-}\mathbf{Coalg} \tag{5.8}$$

Specializing to **Set** coalgebras, the intuition is:

- We can think of a signature functor of the form $T^A : \mathbf{Set} \to \mathbf{Set}$ as having "inputs" from some set A
- A function $f : A \to B$ allows us to relate A-inputs to B-inputs
- Using such an f, we can build T^A-coalgebras from T^B-coalgebras. Lemma 5.2.1 and corollary 5.2.2 say that for any arbitrary f, we can do this in a uniform way.

We will now use this approach to construct some examples:

Example 5.2.3 (Unitary Evolution)**.** For an arbitrary Hilbert space $\mathcal{H}$ we consider the signature functor Q for quantum systems of definition 3.3.1. A unitary $\hat{U}$ on $\mathcal{H}$ induces a function $\hat{P} \mapsto \hat{U}\hat{P}\hat{U}^\dagger$. Corollary 5.2.2 tells us this induces an endofunctor on Q-**Coalg**. This endofunctor produces a coalgebra giving measurement outcomes as if the unitary $\hat{U}$ had been applied. In this way, the functor encodes unitary (Heisenberg type) evolution of the system. In this approach the unitary evolution is modelled *uniformly* across each coalgebra without extending the signature functor, so for example on every model the unitaries will behave as a group.

Example 5.2.4 (Restriction to Subsystems)**.** Tensor products are used to describe composite systems in quantum mechanics. For example, the state space of a single qubit system is modelled by a Hilbert space $\mathcal{H}_2$. A two qubit system then has state space:

$$\mathcal{H}_2 \otimes \mathcal{H}_2 \tag{5.9}$$

It is common to consider measurements restricted to one of the two qubits. A projective measurement $\hat{P}$ on the left qubit is given by measuring the projection:

$$\hat{P} \otimes 1 \tag{5.10}$$

on the larger system, where 1 is the identity on $\mathcal{H}_2$. The mapping:

$$\mathcal{L}(\mathcal{H}_2) \to \mathcal{L}(\mathcal{H}_2 \otimes \mathcal{H}_2) \tag{5.11}$$

$$\hat{P} \mapsto \hat{P} \otimes 1 \tag{5.12}$$

allows us to lift projective measurements on a single qubit system to local measurements on the left qubit of the composite system. The induced functor given by corollary 5.2.2:

$$(1 + (0,1] \times 1_{\mathbf{Set}})^{\mathcal{L}(\mathcal{H}_2 \otimes \mathcal{H}_2)}\text{-}\mathbf{Coalg} \to (1 + (0,1] \times 1_{\mathbf{Set}})^{\mathcal{L}(\mathcal{H}_2)}\text{-}\mathbf{Coalg} \tag{5.13}$$

enables us to take a coalgebra modelling a composite system, and construct a new coalgebra with our attention restricted to the left qubit. We can clearly construct a similar map to restrict attention to the right qubit, and this approach can be generalized to larger composite systems in a straightforward manner.

We also mention an example of transferring modalities between logics, again using suitable natural transformations:

Example 5.2.5 (Local Measurements)**.** We consider a single qubit system. We define 0-ary predicate lifting (proposition) $[\![\hat{P}]\!]$ by:

$$[\![\hat{P}]\!]_X = \{\varphi \mid \exists x \in X. \varphi\hat{P} = (1, x)\} \tag{5.14}$$

This proposition describes certainty that projective measurement $\hat{P}$ will succeed. We check naturality as follows:

$$\begin{aligned}
& \varphi \in (Qf)^{-1}[\![\hat{P}]\!]_Y \\
\Leftrightarrow \quad & \{ \text{ definitions, inverse images } \} \\
& Q(f)(\varphi) \in \{\psi \mid \exists y \in Y. \psi\hat{P} = (1, y)\}
\end{aligned}$$

$$
\begin{aligned}
\Leftrightarrow\; & \{\text{ definitions, comprehension }\} \\
& \exists x. \varphi \hat{P} = (1, x) \wedge f x \in Y \\
\Leftrightarrow\; & \{\text{ inverse images }\} \\
& \exists x. \varphi \hat{P} = (1, x) \wedge x \in f^{-1} Y \\
\Leftrightarrow\; & \{\text{ inverse images preserve top elements }\} \\
& \exists x. \varphi \hat{P} = (1, x) \wedge x \in X \\
\Leftrightarrow\; & \{\text{ definition }\} \\
& [\![\hat{P}]\!]_X
\end{aligned}
$$

Using lemma 5.2.1, the map:

$$\hat{P} \to \hat{P} \otimes 1 \tag{5.15}$$

considered in example 5.2.4 induces a natural transformation

$$(1 + (0,1] \times 1_{\mathbf{Set}})^{\mathcal{L}(\mathcal{H}_2 \otimes \mathcal{H}_2)} \Rightarrow (1 + (0,1] \times 1_{\mathbf{Set}})^{\mathcal{L}(\mathcal{H}_2)} \tag{5.16}$$

Using this natural transformation, we can lift the predicate lifting $[\![\hat{P}]\!]$ to give a predicate lifting on the composite system, as shown in equation (5.2). This new predicate lifting describes certainty of a local projective measurement of $\hat{P}$ on the first qubit of the larger system.

These quantum examples suggest that functorial relationships between categories of coalgebras and mappings between predicate liftings for the corresponding logics are both of practical interest. We quickly examine a natural, non-quantum mechanical, example.

Example 5.2.6. For a set of labels Σ, we consider the signature functor:

$$\mathcal{P}(-)^{\Sigma} : \mathbf{Set} \to \mathbf{Set} \tag{5.17}$$

This signature is commonly used to model labelled transition systems. By applying corollary 5.2.1, every function of the form:

$$f : \Sigma \to \Sigma' \tag{5.18}$$

induces a functor between categories of labelled transition systems. These functors could be used to:

- Embed a labelled transition system for a small label alphabet into a category of systems on a large alphabet
- Permute the labels on a transition system, analogous to unitary behaviour in example 5.2.3
- Identify certain labels on transition systems

Examples of this type are considered in [Kurz and Pattinson, 2000a].

The key idea of this chapter is to consider extending logics with operations induced by functors between coalgebra categories. We must then also investigate how these functors can be connected to translations on the logical side of the relationship.

5.3 Model Transformations

All the functors discussed in section 5.2 were induced by natural transformations. We examine functors of this type in detail later, starting in section 5.6.

Before doing so, we will consider generalizations of the examples in section 5.2, including cases where the functors involved are not induced by a natural transformation. We first make an observation.

Definition 5.3.1 (Concrete Functor)**.** Let $T, T' : \mathcal{X} \to \mathcal{X}$ be endofunctors, and $U^T : T\text{-}\mathbf{Coalg} \to \mathcal{X}$ and $U^{T'} : T'\text{-}\mathbf{Coalg} \to \mathcal{X}$ the obvious forgetful functors. We will say that a functor $H : T\text{-}\mathbf{Coalg} \to T'\text{-}\mathbf{Coalg}$ is **concrete** if it commutes with the forgetful functors, that is:

$$U^T = U^{T'} \circ H \tag{5.19}$$

Concrete functors are very well behaved with respect to behavioural equivalence, due to their trivial action on morphisms:

Lemma 5.3.2. *Let $T, T' : \mathbf{Set} \to \mathbf{Set}$ be endofunctors, and:*

$$\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}, \begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix} \tag{5.20}$$

*be T-coalgebras. If $x \in X$ and $y \in Y$ are behaviourally equivalent in T-***Coalg** *and $H : T\text{-}\mathbf{Coalg} \to T'\text{-}\mathbf{Coalg}$ is a concrete functor, then x and y are behaviourally equivalent elements of $H\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$ and $H\begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix}$.*

Proof. As x and y are behaviourally equivalent, there is a cospan in T-**Coalg**:

$$\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \xrightarrow{f} \begin{pmatrix} Z \\ \downarrow \zeta \\ TZ \end{pmatrix} \xleftarrow{g} \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix} \tag{5.21}$$

such that $fx = gy$. As H is concrete we have a cospan in T'-**Coalg**:

$$\begin{pmatrix} X \\ \downarrow H\gamma \\ T'X \end{pmatrix} \xrightarrow{f} \begin{pmatrix} Z \\ \downarrow H\zeta \\ TZ \end{pmatrix} \xleftarrow{g} \begin{pmatrix} Y \\ \downarrow H\xi \\ TY \end{pmatrix} \tag{5.22}$$

and this cospan shows x and y remain behaviourally equivalent in T'-**Coalg**. □

The functors discussed in section 5.2 are all concrete functors induced by appending natural transformations to the dynamics of a coalgebra. To broaden our scope, we begin by investigating concrete functors of practical interest, but not of this form. Eventually, we will progress to examples beyond the restriction to concrete functors. We then study how these examples can be handled logically.

5.3.1 More General Examples

5.3.1.1 Iterating Dynamics

Definition 5.3.3 (Kleisli Iteration)**.** Let (T, η, μ) be a monad. For T-coalgebra $X \xrightarrow{\gamma} TX$ we define, using Kleisli composition from definition 2.4.8:

$$\gamma^1 := \gamma \qquad \gamma^{n+1} := \gamma^n \bullet \gamma \tag{5.23}$$

Remark 5.3.4. The iteration described in definition 5.3.3 is used in the context of random walks in [Jacobs, 2011b].

Lemma 5.3.5. *For monad* (T, η, μ)*, if:*

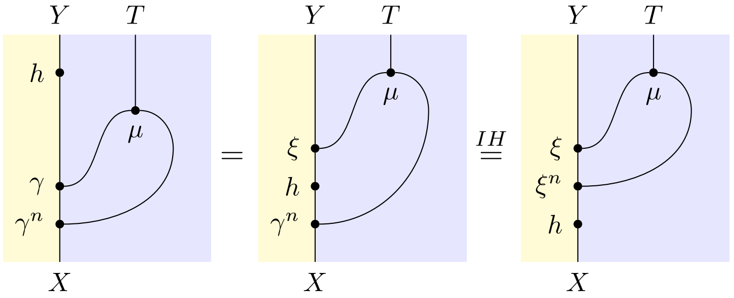
$$\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix} \tag{5.24}$$

is a T*-coalgebra homomorphism, then for all* $n \geq 1$*:*

$$\begin{pmatrix} X \\ \downarrow \gamma^n \\ TX \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow \xi^n \\ TY \end{pmatrix} \tag{5.25}$$

is a T-coalgebra homomorphism. That is, Kleisli iteration extends to a concrete functor.

Proof. We proceed by induction on n, the base case follows immediately from the assumption. For the inductive step we calculate:



(5.26)

□

Remark 5.3.6. We note that Kleisli iteration is *not* induced by simply appending a natural transformation as with our motivating examples of section 5.2. For a given formula φ in ordinary modal logic for the signature functor $\mathcal{P}$, it is possible to find a formula ψ such that for all γ:

$$[\![\varphi]\!]_{\gamma^n} = [\![\psi]\!]_{\gamma} \tag{5.27}$$

So in this case we can equally consider functorial transformations of coalgebraic models or syntactic transformations of formulae.

5.3.1.2 Closure of Binary Relations

As natural operations we may wish to perform on coalgebraic models, we consider the reflexive, transitive and symmetric closure of binary relations.

Definition 5.3.7 (Reflexive Closure)**.** For $\mathcal{P}$-coalgebra $\left(\begin{tikzcd} X \arrow[d, "\gamma"] \\ \mathcal{P}X \end{tikzcd}\right)$ we define the **reflexive closure** of γ as follows:

$$\gamma^{\mathrm{r}}\, x := \gamma\, x \cup \{x\} \tag{5.28}$$

Lemma 5.3.8. *Reflexive closure extends to a concrete endofunctor:*

$$(-)^{\mathrm{r}} : \mathcal{P}\textbf{-Coalg} \to \mathcal{P}\textbf{-Coalg} \tag{5.29}$$

Proof. It is sufficient to show that any $\mathcal{P}$-coalgebra homomorphism:

$$\left(\begin{array}{c} X \\ \downarrow \gamma \\ \mathcal{P}X \end{array}\right) \xrightarrow{h} \left(\begin{array}{c} Y \\ \downarrow \xi \\ \mathcal{P}Y \end{array}\right) \tag{5.30}$$

is also a $\mathcal{P}$-coalgebra homomorphism:

$$\left(\begin{array}{c} X \\ \downarrow \gamma^{\mathrm{r}} \\ \mathcal{P}X \end{array}\right) \xrightarrow{h} \left(\begin{array}{c} Y \\ \downarrow \xi^{\mathrm{r}} \\ \mathcal{P}Y \end{array}\right) \tag{5.31}$$

We reason as follows:

$$\begin{aligned}
& y \in \mathcal{P}h \circ \gamma^{\mathrm{r}}\, x \\
\Leftrightarrow\ & \{\ \text{definitions}\ \} \\
& y \in \mathcal{P}h(\gamma\, x \cup \{x\}) \\
\Leftrightarrow\ & \{\ \text{definitions}\ \} \\
& y \in \mathcal{P}h \circ \gamma\, x \vee y = h\, x \\
\Leftrightarrow\ & \{\ h \text{ is a } \mathcal{P}\text{-coalgebra homomorphism}\ \} \\
& y \in \xi \circ h \vee y = h\, x \\
\Leftrightarrow\ & \{\ \text{definitions}\ \} \\
& y \in \xi^{\mathrm{r}} \circ h\, x
\end{aligned}$$

□

Definition 5.3.9 (Transitive Closure)**.** Using Kleisli iteration from definition 5.3.3, we define the **transitive closure** of $\mathcal{P}$-coalgebra $\left(\begin{array}{c} X \\ \downarrow \gamma \\ \mathcal{P}X \end{array}\right)$ as:

$$\gamma^{\mathrm{t}}\, x := \bigcup_{n \geq 1} \gamma^{n}\, x \tag{5.32}$$

Lemma 5.3.10. *Transitive closure extends to a concrete functor* $\mathcal{P}$-**Coalg** $\to$ $\mathcal{P}$-**Coalg**

Proof. It is sufficient to show that if:

$$\left(\begin{array}{c} X \\ \downarrow \gamma \\ \mathcal{P}X \end{array}\right) \xrightarrow{h} \left(\begin{array}{c} Y \\ \downarrow \xi \\ \mathcal{P}Y \end{array}\right) \tag{5.33}$$

is a $\mathcal{P}$-coalgebra morphism, then so is:

$$\left(\begin{tikzcd}[row sep=small] X \arrow[d, "\gamma^{\mathrm{t}}"] \\ \mathcal{P}X \end{tikzcd}\right) \xrightarrow{h} \left(\begin{tikzcd}[row sep=small] Y \arrow[d, "\xi^{\mathrm{t}}"] \\ \mathcal{P}Y \end{tikzcd}\right) \tag{5.34}$$

Composition and identities follow from the base category. We reason as follows:

$$\begin{aligned}
& y \in \xi^{\mathrm{t}} \circ h\, x \\
\Leftrightarrow\ & \{ \text{ definition } \} \\
& \exists n \geq 1 . y \in \xi^{n} \circ h\, x \\
\Leftrightarrow\ & \{ \text{ lemma 5.3.5 } \} \\
& \exists n \geq 1 . y \in \mathcal{P}h \circ \gamma^{n}\, x \\
\Leftrightarrow\ & \{ \text{ definition } \} \\
& y \in \mathcal{P}h \circ \gamma^{\mathrm{t}}\, x
\end{aligned}$$

□

Remark 5.3.11. By considering the expressivity of ordinary modal logic, we observe that generally, for a formula φ, it will *not* be possible to find a formula ψ such that for all γ:

$$[\![\varphi]\!]_{\gamma^{\mathrm{t}}} = [\![\psi]\!]_{\gamma} \tag{5.35}$$

That is, we will not be able to find a transformation of formulae that exactly mimics the transitive closure of models.

To illustrate that not every "natural" operation we might wish to perform on binary relations is functorial, we also consider symmetric closure.

Counterexample 5.3.12 (Symmetric Closure)**.** We consider the $\mathcal{P}$-coalgebras:

$$\gamma : \{x\} \to \mathcal{P}\{x\} \qquad \xi : \{a, b\} \to \mathcal{P}\{a, b\} \tag{5.36}$$

$$\gamma\, x = \emptyset \qquad \xi\, a = \{b\} \tag{5.37}$$

$$\xi\, b = \emptyset \tag{5.38}$$

By considering bisimilarity, there is clearly a unique $\mathcal{P}$-coalgebra homomorphism:

$$\left(\begin{tikzcd}[row sep=small] \{x\} \arrow[d, "\gamma"] \\ \mathcal{P}\{x\} \end{tikzcd}\right) \xrightarrow{x \mapsto b} \left(\begin{tikzcd}[row sep=small] \{a, b\} \arrow[d, "\xi"] \\ \mathcal{P}\{a, b\} \end{tikzcd}\right) \tag{5.39}$$

Now if we again consider bisimilarity for elements in the symmetric closure of these relations, we see that there can be no homomorphisms (in either direction) between them.

5.3.1.3 Model Updates

We now consider a more realistic logical example in the context of generalized announcement logics. The induced model updates again turn out to be concretely functorial.

Definition 5.3.13 (Update)**.** An **update** [Carreiro et al., 2013] is a natural transformation of the form:

$$\Delta : T \Rightarrow (2 \rightarrowtriangle T) \tag{5.40}$$

Here the functor $2 \rightarrowtriangle T$ is defined as:

$$(2 \rightarrowtriangle T)(X) := T(X)^{\mathcal{P}(X)} \tag{5.41}$$

$$(2 \rightarrowtriangle T)(h : X \to Y) := Th \circ (-) \circ h^{-1} \tag{5.42}$$

The intuition is that an update takes the structured output from the dynamics of a T-coalgebra and the extension of some formula, and returns a modified output from the dynamics. The modification might reflect, for example, a public announcement changing the knowledge of agents in an epistemic logic.

Lemma 5.3.14. *An update* $\Delta : T \Rightarrow (2 \rightarrowtriangle T)$ *and a formula* φ*, without proposition variables, induce a concrete endofunctor:*

$$\Delta^*_\varphi : T\text{-}\mathbf{Coalg} \Rightarrow T\text{-}\mathbf{Coalg} \tag{5.43}$$

The action on objects is defined as:

$$\Delta^*_\varphi(\gamma)\, x := (\Delta_X \circ \gamma\, x)(\llbracket \varphi \rrbracket_\gamma) \tag{5.44}$$

Proof. Consider arbitrary T-coalgebra homomorphism:

$$\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix} \tag{5.45}$$

We reason as follows:

$$
\begin{aligned}
& \Delta^{*}_{\varphi}(\xi) \circ h\, x \\
= \;& \{ \text{ definition } \} \\
& (\Delta_Y \circ \xi \circ h\, x)(\llbracket \varphi \rrbracket_{\xi}) \\
= \;& \{ \; h \text{ is a coalgebra homomorphism } \} \\
& (\Delta_Y \circ Th \circ \gamma\, x)(\llbracket \varphi \rrbracket_{\xi}) \\
= \;& \{ \text{ naturality } \} \\
& ((2 \rightarrowtriangle T)h \circ \Delta_X \circ \gamma\, x)(\llbracket \varphi \rrbracket_{\xi}) \\
= \;& \{ \text{ action of } 2 \rightarrowtriangle T \text{ on morphisms } \} \\
& (Th \circ (\Delta_X \circ \gamma\, x) \circ h^{-1})(\llbracket \varphi \rrbracket_{\xi}) \\
= \;& \{ \text{ standard property of semantics over homomorphisms } \} \\
& (Th \circ (\Delta_X \circ \gamma\, x)(\llbracket \varphi \rrbracket_{\gamma}) \\
= \;& \{ \text{ definition } \} \\
& Th \circ \Delta^{*}_{\varphi}(\gamma)\, x
\end{aligned}
$$

$\square$

Remark 5.3.15. We will return to model updates briefly in example 5.3.25.

5.3.1.4 Determinization

For a label alphabet Σ, we can model a *non-deterministic* automaton by a coalgebra with the signature functor:

$$\mathcal{P}(-)^{\Sigma} \times \mathbb{B} \tag{5.46}$$

Here $\mathbb{B}$ denotes the set of Booleans, used to indicate if a state is accepting. *Deterministic* automata can also be modelled coalgebraically, this time by using the signature functor:

$$(-)^{\Sigma} \times \mathbb{B} \tag{5.47}$$

We can **determinize** a non-deterministic automaton $\left(\begin{tikzcd} X \arrow[d, "\gamma"] \\ \mathcal{P}X^{\Sigma} \times \mathbb{B} \end{tikzcd}\right)$ to produce a deterministic one. This process is referred to as the **powerset construction** or the **subset construction**, see for example [Sakarovitch, 2009] for details. The resulting automaton is defined as follows:

- The state space of the deterministic automaton is $\mathcal{P}X$

- Fix a label $\sigma \in \Sigma$ and a state $U \subseteq X$. There is then a unique σ-transition from U to the set:
$$\left\{x \in X \mid \exists u \in U.\, u \xrightarrow{\sigma} x\right\} \tag{5.48}$$
- A state $U \subseteq X$ is accepting if any $u \in U$ is accepting in the original automata
- For $x \in X$, the language accepted by $\{x\}$ is then equal to the language accepted by x in the original automaton

Determinization and other similar transformations are discussed in a coalgebraic setting in [Silva et al., 2013]. We note that determinization is *not* a concrete functor between categories of coalgebras, as it changes the state space. It is however functorial, as we now demonstrate.

Lemma 5.3.16. *Determinization defines a functor:*

$$(-)^{\mathrm{det}} : \mathcal{P}(-)^{\Sigma} \times \mathbb{B}\text{-}\mathbf{Coalg} \to (-)^{\Sigma} \times \mathbb{B}\text{-}\mathbf{Coalg} \tag{5.49}$$

Further, the unit of the powerset monad gives a natural transformation $U \Rightarrow V \circ (-)^{\mathrm{det}}$ *where* U *and* V *are the corresponding forgetful functors to* **Set**.

Proof. It is well known there is a distributive law of monads $\delta : \mathcal{P}((-)^{\Sigma}) \Rightarrow \mathcal{P}(-)^{\Sigma}$, given explicitly as follows:

$$\delta_X(U)(\sigma) := \{f\sigma \mid f \in U\} \tag{5.50}$$

Also let *join* be the map sending a set of Boolean values to their join, and $(\mathcal{P}, \eta, \mu)$ the powerset monad. We consider the composite:

$$\begin{tikzcd}
\mathcal{P}(\mathcal{P}X^{\Sigma} \times \mathbb{B}) \arrow[r, "\langle \mathcal{P}\pi_1, \mathcal{P}\pi_2 \rangle"] & \mathcal{P}(\mathcal{P}X^{\Sigma}) \times \mathcal{P}\mathbb{B} \arrow[d, "\delta_{\mathcal{P}X} \times 1"] & \\
& \mathcal{P}\mathcal{P}(X)^{\Sigma} \times \mathcal{P}\mathbb{B} \arrow[r, "\mu_X^{\Sigma} \times \mathit{join}"'] & \mathcal{P}X^{\Sigma} \times \mathbb{B}
\end{tikzcd} \tag{5.51}$$

It is straightforward to check the three components are natural in X, and therefore the composite gives a natural transformation of the form:

$$\mathcal{P}(\mathcal{P}(-)^{\Sigma} \times \mathbb{B}) \Rightarrow \mathcal{P}X^{\Sigma} \times \mathbb{B} \tag{5.52}$$

Using the dual of the result in example 2.1.1, this natural transformation induces a functor of the required type. Confirming that this functor is exactly determinization is easy to check, as is the final part. □

5.3.2 Model Transformation Modalities

We now discuss the extension of coalgebraic modal logic with modalities induced by suitable functors between categories of models. As our most general example, we abstract from determinization, exploiting the structure observed in lemma 5.3.16.

Definition 5.3.17 (Model Transformation)**.** A **model transformation** of T-coalgebras to T'-coalgebras is a pair (P, η) consisting of a functor:

$$P : T\text{-}\mathbf{Coalg} \to T'\text{-}\mathbf{Coalg} \tag{5.53}$$

and a natural transformation:

$$\eta : U^T \Rightarrow U^{T'} \circ P \tag{5.54}$$

where $U^T : T\text{-}\mathbf{Coalg} \to \mathbf{Set}$ and $U^{T'} : T'\text{-}\mathbf{Coalg} \to \mathbf{Set}$ are the appropriate forgetful functors.

Remark 5.3.18. The intuition for the definition of model transformation is that the functor P transforms models. As the state space may have changed, the natural transformation η identifies representatives of states of the original model in the resulting state space. In the common special case of concrete functors, η is then the identity natural transformation.

Lemma 5.3.19. *Let (P, η) be a model transformation. For T-coalgebras $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$ and $\begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix}$ with elements $x \in X$ and $y \in Y$:*

$$x \sim y \quad \Rightarrow \quad \eta_\gamma(x) \sim \eta_\xi(y) \tag{5.55}$$

Proof. For T-coalgebras $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$ and $\begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix}$ with elements $x \in X$ and $y \in Y$, we assume the existence of a cospan in T-**Coalg**:

$$\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix} \xrightarrow{f} \begin{pmatrix} Z \\ \downarrow\zeta \\ TZ \end{pmatrix} \xleftarrow{g} \begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix} \tag{5.56}$$

with $f\,x = g\,y$. We then have the following equalities:

$$\begin{aligned}
& UPf \circ \eta_\gamma\, x \\
= \; & \{ \text{ naturality } \} \\
& \eta_\zeta \circ f\, x \\
= \; & \{ \text{ assumption of behavioural equivalence } \} \\
& \eta_\zeta \circ g\, y \\
= \; & \{ \text{ naturality } \} \\
& UPg \circ \eta_\xi\, y
\end{aligned}$$

□

Definition 5.3.20 (Semantics)**.** We extend the semantics given in definition 4.3.6 with **model transformation modalities** given, for a model transformation (P, η), as follows:

$$[\![P\varphi]\!]_{(X,\gamma),\mathrm{v}} := \eta_X^{-1}[\![\varphi]\!]_{H(X,\gamma),\mathrm{v}} \tag{5.57}$$

As logical connections relate formulae for different signature functors, we will refer to a formula to be interpreted on a T-coalgebra as a T**-formula**.

We must show that such an extension respects behavioural equivalence, in the fragment without proposition variables. We first require a generalization of a standard technical lemma.

Lemma 5.3.21. *For endofunctor $T : \mathbf{Set} \to \mathbf{Set}$ and T-coalgebra homomorphism:*

$$\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix} \tag{5.58}$$

for formula φ without proposition variables:

$$(U^T h)^{-1}[\![\varphi]\!]_\xi = [\![\varphi]\!]_\gamma \tag{5.59}$$

We make the forgetful functor explicit in equation (5.59) *for clarity in subsequent proofs.*

Proof. We proceed by induction on the structure of φ. We consider the model transformation modalities explicitly, the other cases are straightforward and appear in the literature in similar lemmas for standard formulations of coalgebraic logic.

$$
\begin{aligned}
& (U^T h)^{-1}[\![P\varphi]\!]_\xi \\
= & \quad \{ \text{ definition } \} \\
& (U^T h)^{-1}\eta_\xi^{-1}[\![\varphi]\!]_{P\xi} \\
= & \quad \{ \text{ functoriality } \} \\
& (\eta_\xi \circ U^T h)^{-1}[\![\varphi]\!]_{P\xi} \\
= & \quad \{ \text{ naturality } \} \\
& (U^{T'} \circ Ph \circ \eta_\gamma)[\![\varphi]\!]_{P\xi} \\
= & \quad \{ \text{ functoriality } \} \\
& \eta_\gamma^{-1} \circ (U^{T'} \circ Ph)^{-1}[\![\varphi]\!]_{P\xi} \\
= & \quad \{ \text{ induction hypothesis } \} \\
& \eta_X^{-1}[\![\varphi]\!]_{P\gamma} \\
= & \quad \{ \text{ definition } \} \\
& [\![P\varphi]\!]_\gamma
\end{aligned}
$$

□

Remark 5.3.22. Invariance properties similar to those in lemma 5.3.21 were exploited in [Kurz and Rosickỳ, 2005] to develop a general notion of modal predicate.

This lemma provides the key ingredient such that we can now show adequacy of logics extended with model transformation modalities:

Proposition 5.3.23. *Let $T : \mathbf{Set} \to \mathbf{Set}$ be an endofunctor, $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$ and $\begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix}$ T-coalgebras, $x \in X$ and $y \in Y$ with $x \sim y$. Every formula φ in coalgebraic logic without proposition variables, extended with model transformation modalities, satisfies:*

$$x \models_\gamma \varphi \quad \Leftrightarrow \quad y \models_\xi \varphi \tag{5.60}$$

Proof. As $x \sim y$ there exists a cospan in T-**Coalg**:

$$\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix} \xrightarrow{f} \begin{pmatrix} Z \\ \downarrow\zeta \\ TZ \end{pmatrix} \xleftarrow{g} \begin{pmatrix} Y \\ \downarrow\xi \\ TY \end{pmatrix} \tag{5.61}$$

with $f\,x = g\,y$. We then have, using lemma 5.3.21:

$$x \models_\gamma \varphi \quad \Leftrightarrow \quad x \in [\![\varphi]\!]_\gamma \quad \Leftrightarrow \quad x \in f^{-1}[\![\varphi]\!]_\zeta \quad \Leftrightarrow \quad f\,x \models_\zeta \varphi \tag{5.62}$$

and again using lemma 5.3.21:

$$y \models_\xi \varphi \quad \Leftrightarrow \quad y \in [\![\varphi]\!]_\xi \quad \Leftrightarrow \quad y \in g^{-1}[\![\varphi]\!]_\zeta \quad \Leftrightarrow \quad g\,y \models_\zeta \varphi \tag{5.63}$$

and so as $f\,x = g\,y$:

$$x \models_\gamma \varphi \quad \Leftrightarrow \quad y \models_\xi \varphi \tag{5.64}$$

□

Model transformation modalities are complete Boolean algebra homomorphisms:

Lemma 5.3.24. *Let:*

- $\begin{pmatrix} X \\ \downarrow\gamma \\ TX \end{pmatrix}$ *be a* T*-coalgebra*
- v *be a valuation*
- φ *be a* T*-formula*
- $(\varphi_i)_{i\in I}$ *be a family of* T*-formulae*
- p *be a proposition variable*
- (P, η) *be a model transformation*

We have:

$$[\![P\bigwedge_{i\in I}\varphi_i]\!]_{(X,\gamma),\mathrm{v}} = [\![\bigwedge_{i\in I}P\varphi_i]\!]_{(X,\gamma),\mathrm{v}} \tag{5.65}$$

$$[\![P\neg\varphi]\!]_{(X,\gamma),\mathrm{v}} = [\![\neg P\varphi]\!]_{(X,\gamma),\mathrm{v}} \tag{5.66}$$

$$[\![Pp]\!]_{(X,\gamma),\mathrm{v}} = [\![p]\!]_{(X,\gamma),\eta^{-1}_{U\gamma}\circ\mathrm{v}} \tag{5.67}$$

Proof. This follows from expanding definitions and noting that inverse image maps are complete Boolean algebra homomorphisms. □

Example 5.3.25 (Model updates)**.** We consider the functors induced by model updates discussed in section 5.3.1.3. For update Δ and formulae φ, ψ, the semantics of a model update modality is given [Carreiro et al., 2013] by:

$$[\![\Delta_\psi\varphi]\!]_\gamma := [\![\varphi]\!]_{x\mapsto(\Delta_X\circ\gamma)([\![\psi]\!]_\gamma)} \tag{5.68}$$

Informally, the model update modifies the underlying coalgebra, based on the extension of a formula. As described in the paper, it "takes the T-description of an element, and returns an updated T-element".

The model transformation modality for the concrete functor described in section 5.3.1.3 gives us the same semantics. This can be seen as follows, in the case without proposition variables:

$$\begin{aligned}
& [\![\Delta^*_\psi \varphi]\!]_\gamma \\
= & \quad \{ \text{ definition 5.3.20 } \} \\
& [\![\varphi]\!]_{\Delta^*_\psi \gamma} \\
= & \quad \{ \text{ lemma 5.3.14 } \} \\
& [\![\varphi]\!]_{x \mapsto (\Delta_X \circ \gamma x)([\![\psi]\!]_\gamma)} \\
= & \quad \{ \text{ equation (5.68) } \} \\
& [\![\Delta_\psi \varphi]\!]_\gamma
\end{aligned}$$

We have so far made very weak technical assumptions. We will now move on to consider a restricted, but still reasonably broad setting. Our restricted assumptions remain sufficient for the motivating quantum examples of section 5.2. In this well behaved set up we can then establish some strong results, relating functors transforming models and transformations of logical formulae.

5.4 Fibrations

In the remainder of this chapter we will analyze logics varying over a parameter. This will require the categorical notion of fibration, formalizing the idea of categories parameterized by some base category. To keep this thesis reasonably self contained, and in order to fix notation, we now provide a brief introduction to the aspects of fibred category theory that will be required in later sections. Standard references for this material are [Jacobs, 1999, Borceux, 1994, Phoa, 1992, Streicher, 1999], the presentation and notational choices in this section is heavily influenced by those sources.

A fibration is simply a functor satisfying some additional axioms. Such functors are typically drawn vertically:

$$\left(\begin{tikzcd} \mathcal{E} \arrow[d, "p"] \\ \mathcal{B} \end{tikzcd}\right) \tag{5.69}$$

We will denote fibrations using lowercase letters. Where the functor p is clear from the context, it may be omitted from the notation. The category $\mathcal{B}$ is referred to as the **base category**, and the category $\mathcal{E}$ as the **total category**. We will often refer to objects and morphisms in the base category as **base objects** and **base morphisms**. An object $\tilde{X}$ in the total category is said to be **over**, or **above**, base object X if $p\tilde{X} = X$. Similarly, a morphism $\tilde{u}$ in the total category is said to be **over**, or **above**, base morphism u if $p\tilde{u} = u$. This is often drawn suggestively as in the following diagram:

$$\begin{tikzcd}
\tilde{X} \arrow[r, "\tilde{u}"] & \tilde{Y} \\
X \arrow[r, "u"'] & Y
\end{tikzcd} \tag{5.70}$$

Clearly there may be many morphisms in the total category above any given morphism in the base.

Definition 5.4.1 (Cartesian Morphism)**.** A morphism $f : V \to W$ is **cartesian** over $u : Y \to Z$ if both:

1. f is over u
2. If there exists a base morphism $t : X \to Y$ and $\mathcal{E}$ morphism $g : U \to W$ such that $u \circ t = pg$ then there exists a unique $! : U \to V$ over t such that g factors as $f \circ !$, as in the following diagram:

$$\begin{tikzcd}
U \arrow[dr, dashed, "!"'] \arrow[drr, bend left, "g"] & & \\
& V \arrow[r, "f"'] & W \\
X \arrow[dr, "t"'] \arrow[drr, bend left, "u \circ t"] & & \\
& Y \arrow[r, "u"'] & Z
\end{tikzcd} \tag{5.71}$$

Definition 5.4.2 (Cartesian Lift)**.** For a functor $\begin{pmatrix}\mathcal{E}\\ \downarrow{\scriptstyle p}\\ \mathcal{B}\end{pmatrix}$, given a base morphism $u : X \to Y$ and total category object $\tilde{Y}$ over Y, a **cartesian lift** of $(u, \tilde{Y})$ is a cartesian morphism f over u as in the following diagram:

$$\begin{tikzcd}
V \arrow[r, "f"] & \tilde{Y} \\
X \arrow[r, "u"'] & Y
\end{tikzcd} \tag{5.72}$$

The term **p-cartesian lifting** or simply **p-lifting** will be used when we need to clarify which functor the lifting is with respect to. We may also say that f is **cartesian over** (or **p-cartesian over**) u.

Definition 5.4.3 (Fibration)**.** A functor $\left(\begin{smallmatrix}\mathcal{E}\\ \downarrow\\ \mathcal{B}\end{smallmatrix}\right)$ is said to be a **fibration**, or **fibred category**, if every base morphism $u : X \to Y$ and total category object $\tilde{Y}$ over Y has a cartesian lift.

For a $\mathcal{B}$ object B the **fibre** over B, denoted $\mathcal{E}_B$, is the subcategory of $\mathcal{E}$ with:

1. **Objects**: Total category objects over B
2. **Morphisms**: Total category morphisms over the identity 1_B

Morphisms over an identity in the base category are said to be **vertical**.

Definition 5.4.4 (Cloven Fibration)**.** Let $\left(\begin{smallmatrix}\mathcal{E}\\ \downarrow p\\ \mathcal{B}\end{smallmatrix}\right)$ be a fibration. A **cleavage** for p is a specified choice of cartesian lifting for each base morphism $u : X \to Y$ and object $\tilde{Y}$ above Y:

$$\begin{tikzcd}
u^*(\tilde{Y}) \arrow[r, "\overline{u}(\tilde{Y})"] & \tilde{Y} \\
X \arrow[r, "u"'] & Y
\end{tikzcd} \tag{5.73}$$

Following the convention in [Jacobs, 1999], we will often write the morphism $\overline{u}(\tilde{Y})$ as simply $\overline{u}$, leaving the object $\tilde{Y}$ implicit. A **cloven fibration** is a fibration with a given cleavage.

Remark 5.4.5. Assuming the axiom of choice, a fibration can always be given a cleavage.

Definition 5.4.6 (Split Fibration)**.** A **splitting** is a cleavage such that for all

$X, Y, Z, \tilde{Z}, u : X \to Y, v : Y \to Z$ and $\tilde{Z}$ over Z:

$$\begin{tikzcd} u^*(v^*\tilde{Z}) \arrow[r, "\overline{u}"] & v^*\tilde{Z} \arrow[r, "\overline{v}"] & \tilde{Z} & (v \circ u)^*\tilde{Z} \arrow[r, "\overline{v \circ u}"] & \tilde{Z} \\ X \arrow[r, "u"'] & Y \arrow[r, "v"'] & Z & X \arrow[r, "v \circ u"'] & Z \end{tikzcd} \quad = \tag{5.74}$$

A **split fibration** is a fibration with a given splitting.

Remark 5.4.7. Although, as noted in remark 5.4.5, assuming the axiom of choice every fibration can be given a cleavage, not every fibration has a splitting.

Definition 5.4.8 (Substitution Functors)**.** Given a cloven fibration:

$$\begin{pmatrix} \mathcal{E} \\ \downarrow \\ \mathcal{B} \end{pmatrix} \tag{5.75}$$

every base morphism $u : X \to Y$ induces a **substitution functor**:

$$u^* : \mathcal{E}_Y \to \mathcal{E}_X \tag{5.76}$$

For a vertical morphism $U \xrightarrow{f} V$ the action on morphisms is defined using the universal property of cartesian morphisms as in the following diagram:

$$\begin{tikzcd} u^*(V) \arrow[r, "\overline{u}(V)"] \arrow[d, dashed, "u^*(f)"'] & V \arrow[d, "f"] \\ u^*(U) \arrow[r, "\overline{u}(U)"'] & U \\ X \arrow[r, "u"] \arrow[d, "1_X"'] & Y \arrow[d, "1_Y"] \\ X \arrow[r, "u"'] & Y \end{tikzcd} \tag{5.77}$$

In general we only have isomorphisms:

$$1_{\mathcal{E}_B} \cong 1_B^* \quad \text{and} \quad u^* \circ v^* \cong (v \circ u)^* \tag{5.78}$$

In the case of split fibrations, the isomorphisms are equalities.

Definition 5.4.9. The 2-category $\mathfrak{Fib}$ consists of:

- **0-cells**: Fibrations
- **1-cells**: A 1-cell of type:

$$\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix} \to \begin{pmatrix} \mathcal{E}' \\ \downarrow p' \\ \mathcal{B}' \end{pmatrix} \tag{5.79}$$

  is a pair of functors (F, G) such that the following diagram commutes:

$$\begin{tikzcd} \mathcal{E} \arrow[r, "F"] \arrow[d, "p"'] & \mathcal{E}' \arrow[d, "p'"] \\ \mathcal{B} \arrow[r, "G"'] & \mathcal{B}' \end{tikzcd} \tag{5.80}$$

  and F preserves cartesian morphisms. For each $\mathcal{B}$ object B, F restricts to a functor between fibres that we shall denote:

$$F_B : \mathcal{E}_B \to \mathcal{E}'_{G(B)} \tag{5.81}$$

- **2-cells**: A 2-cell of type:

$$(F, G) \Rightarrow (F', G') \tag{5.82}$$

  is a pair of natural transformations (β, ϵ) with types as in the following diagram:

$$\begin{tikzcd} \mathcal{E} \arrow[r, bend left, "F"{name=U}] \arrow[r, bend right, "F'"'{name=D}] \arrow[d, "p"'] \arrow[Rightarrow, from=U, to=D, "\epsilon"] & \mathcal{E}' \arrow[d, "p'"] \\ \mathcal{B} \arrow[r, bend left, "G"{name=U2}] \arrow[r, bend right, "G'"'{name=D2}] \arrow[Rightarrow, from=U2, to=D2, "\beta"] & \mathcal{B}' \end{tikzcd} \tag{5.83}$$

  such that ϵ is above β.

We write **Fib** for the corresponding ordinary category where we ignore 2-cells in $\mathfrak{Fib}$.

Definition 5.4.10. The 2-category $\mathfrak{Fib}_{\text{split}}$ consists of:

- **0-cells**: Split fibrations

- **1-cells**: A 1-cell of type:

$$\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix} \to \begin{pmatrix} \mathcal{E}' \\ \downarrow p' \\ \mathcal{B}' \end{pmatrix} \tag{5.84}$$

is a pair of functors (F, G) such that the following diagram commutes:

$$\begin{tikzcd} \mathcal{E} \arrow[r, "F"] \arrow[d, "p"'] & \mathcal{E}' \arrow[d, "p'"] \\ \mathcal{B} \arrow[r, "G"'] & \mathcal{B}' \end{tikzcd} \tag{5.85}$$

and that the splitting is preserved "on the nose". Concretely, this condition requires that for $\mathcal{B}$-morphism $u : X \to Y$ and $\mathcal{E}$-object $\tilde{Y}$ over Y:

$$F\left(\overline{u}(\tilde{Y})\right) = \overline{G(u)}(F\tilde{Y}) \tag{5.86}$$

As in definition 5.4.9, for each $\mathcal{B}$ object, F restricts to a functor between the appropriate fibres.

- **2-cells**: A 2-cell of type:

$$(F, G) \Rightarrow (F', G') \tag{5.87}$$

is a pair of natural transformations (β, ϵ) with types as in the following diagram:

$$\begin{tikzcd} \mathcal{E} \arrow[r, bend left, "F", ""{name=U, below}] \arrow[r, bend right, "F'"', ""{name=D, above}] \arrow[d, "p"'] & \mathcal{E}' \arrow[d, "p'"] \\ \mathcal{B} \arrow[r, bend left, "G", ""{name=U2, below}] \arrow[r, bend right, "G'"', ""{name=D2, above}] & \mathcal{B}' \arrow[from=U, to=D, Rightarrow, "\epsilon"] \arrow[from=U2, to=D2, Rightarrow, "\beta"] \end{tikzcd} \tag{5.88}$$

such that ϵ is above β.

We write $\mathbf{Fib}_{\text{split}}$ for the ordinary category where we drop the 2-cells in $\mathfrak{Fib}_{\text{split}}$.

Definition 5.4.11. For category $\mathcal{B}$ the 2-category $\mathfrak{Fib}(\mathcal{B})$ consists of:

- **0-cells**: Fibrations over $\mathcal{B}$

- **1-cells**: A 1-cell of type:

$$\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix} \to \begin{pmatrix} \mathcal{E}' \\ \downarrow p' \\ \mathcal{B} \end{pmatrix} \tag{5.89}$$

  is a functor F such that the following diagram commutes:

$$\begin{tikzcd} \mathcal{E} \arrow[rr, "F"] \arrow[dr, "p"'] & & \mathcal{E}' \arrow[dl, "p'"] \\ & \mathcal{B} & \end{tikzcd} \tag{5.90}$$

  and that preserves cartesian morphisms. Similarly to definition 5.4.9, for each $\mathcal{B}$ object B there is an induced functor between fibres, denoted:

$$F_B : \mathcal{E}_B \to \mathcal{E}'_B \tag{5.91}$$

- **2-cells**: A 2-cell of type $F \Rightarrow F'$ is a natural transformation ϵ as in the following diagram:

$$\begin{tikzcd} \mathcal{E} \arrow[rr, bend left, "F", ""{name=U, below}] \arrow[rr, bend right, "F'"', ""{name=D, above}] \arrow[dr, "p"'] & & \mathcal{E}' \arrow[dl, "p'"] \\ & \mathcal{B} & \arrow[Rightarrow, from=U, to=D, "\epsilon"] \end{tikzcd} \tag{5.92}$$

  such that the components of ϵ are vertical.

We write **Fib**($\mathcal{B}$) for the ordinary category where we drop the 2-cells in $\mathfrak{Fib}(B)$.

Definition 5.4.12. For category $\mathcal{B}$ the 2-category $\mathfrak{Fib}_{\text{split}}(\mathcal{B})$ consists of:

- **0-cells**: Split fibrations over $\mathcal{B}$
- **1-cells**: A 1-cell of type:

$$\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix} \to \begin{pmatrix} \mathcal{E}' \\ \downarrow p' \\ \mathcal{B} \end{pmatrix} \tag{5.93}$$

is a functor F such that the following diagram commutes:

$$\begin{tikzcd}
\mathcal{E} \arrow[rr, "F"] \arrow[dr, "p"'] & & \mathcal{E}' \arrow[dl, "p'"] \\
& \mathcal{B} &
\end{tikzcd} \tag{5.94}$$

and that preserves the splitting "on the nose". Concretely we require that for base morphism $u : X \to Y$ an $\mathcal{E}$-object $\tilde{Y}$ that:

$$F\left(\overline{u}(\tilde{Y})\right) = \overline{u}(F\tilde{Y}) \tag{5.95}$$

- **2-cells**: A 2-cell of type $F \Rightarrow F'$ is a natural transformation ϵ as in the following diagram:

$$\begin{tikzcd}
\mathcal{E} \arrow[rr, bend left, "F"{name=U}] \arrow[rr, bend right, "F'"'{name=D}] \arrow[dr, "p"'] & & \mathcal{E}' \arrow[dl, "p'"] \\
& \mathcal{B} &
\arrow[Rightarrow, from=U, to=D, "\epsilon"]
\end{tikzcd} \tag{5.96}$$

  such that the components of ϵ are vertical.

We write $\mathbf{Fib}_{\text{split}}(B)$ for the ordinary category where we drop the 2-cells in $\mathfrak{Fib}_{\text{split}}(B)$.

We will also need to define adjunctions in arbitrary 2-categories, generalizing the situation in $\mathfrak{Cat}$ discussed in section 2.3:

Definition 5.4.13. For an arbitrary 2-category $\mathfrak{C}$, an **adjunction** in $\mathfrak{C}$ is a pair of 1-cells $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ and 2-cells $\eta : 1 \Rightarrow G \circ F$ and $\epsilon : F \circ G \Rightarrow 1$ satisfying the snake equations (2.23a) and (2.23b). As with "ordinary" adjunctions between categories, we then write $F \dashv G$ to emphasize that this is an asymmetric notion. An **equivalence** in $\mathfrak{C}$ is an adjunction in $\mathfrak{C}$ where both η and ϵ are isomorphisms.

The notions of an **adjunction between fibrations** and an **equivalence of fibrations** is then given by instantiating definition 5.4.13 in the appropriate 2-category of fibrations.

We introduce two important fibrations with substitution functors that are useful for building up fibrations.

Lemma 5.4.14. *There are fibrations:*

$$\begin{pmatrix} \mathbf{Fib} \\ \downarrow \\ \mathbf{Cat} \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} \mathbf{Fib}_{split} \\ \downarrow \\ \mathbf{Cat} \end{pmatrix} \tag{5.97}$$

The substitution functors are given by pullbacks in **Cat***, and are referred to as* **change of base** *functors.(See for example [Jacobs, 1999, Lemma 1.5.1 and Lemma 1.7.2])*

The following trivial form of fibration will be needed later:

Definition 5.4.15. For categories $\mathcal{I}$ and $\mathcal{C}$ the **constant fibration** is given by the projection functor:

$$\begin{pmatrix} \mathcal{I} \times \mathcal{C} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \tag{5.98}$$

This is a split fibration with the splitting given by liftings:

$$\begin{tikzcd} (I, C) \arrow[r, "{(f, 1)}"] & (J, C) \\ I \arrow[r, "f"'] & J \end{tikzcd} \tag{5.99}$$

Each of the fibres is isomorphic to $\mathcal{C}$ and the substitution functors are the identity. In more detail, if we consider the projection functor:

$$\pi_1 : \mathcal{C} \times \mathcal{I} \to \mathcal{C} \tag{5.100}$$

for each $\mathcal{I}$ object I the restriction to the fibre:

$$(\mathcal{C} \times \mathcal{I})_I \xrightarrow{\pi_1} \mathcal{C} \tag{5.101}$$

is an isomorphism. Further, for $\mathcal{I}$ morphism $u : I \to J$, the following diagram commutes:

$$\begin{tikzcd} (\mathcal{C} \times \mathcal{I})_J \arrow[rr, "u^*"] \arrow[dr, "\pi_1"'] & & (\mathcal{C} \times \mathcal{I})_I \arrow[dl, "\pi_1"] \\ & \mathcal{C} & \end{tikzcd} \tag{5.102}$$

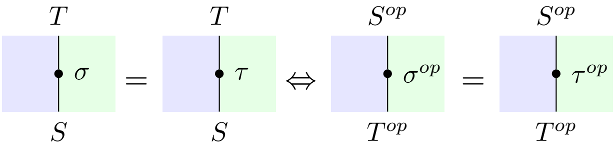
5.5 A Useful 2-Functor

We will continue to use string diagrams in our calculations in this chapter. As we will be considering algebras, coalgebras and related dualities, it will be useful to consider the 2-functor:

$$(-)^{op} : \mathfrak{Cat}^{co} \to \mathfrak{Cat} \tag{5.103}$$

This is the obvious 2-functor that maps categories to their opposite category, functors to the corresponding functor between opposite categories, and natural transformations to a natural transformation in the opposite direction as the orientation of their components are reversed.

In terms of string diagrams, this 2-functor flips diagrams upside down vertically. As $(-)^{op}$ is its own inverse an equation between diagrams holds if and only if the corresponding equation between the opposite diagrams holds. That is:



(5.104)

In the sequel we will invoke this 2-functor silently to avoid cluttering diagrams. This is often done so we may consider coalgebras as algebras in the opposite category, when convenient for calculations.

5.6 Parameterized Algebras and Coalgebras

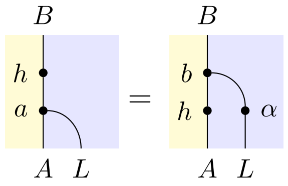
We begin our investigations by describing fibrations of algebras and coalgebras varying over a parameter category. Previous work investigating parameterization of this type is given in [Kurz and Pattinson, 2000a,b, Pattinson, 2002]. Our aim will be to parallel the discussion of ordinary (unparameterized) logical connections in section 4.4. We progress from isomorphisms between fibrations of coalgebras and algebras, to equivalences, and finally to natural transformations inducing fibred algebraization functors. We will then proceed to analyze in some detail the impact of the fibred structure on the relationship between the semantics in different fibres.

To begin with, we construct "large" (in a non-technical sense) fibrations of algebras and coalgebras, varying over the category of signature functors and natural transformations between them. Smaller fibrations capturing a specific form of parameterization will then be constructed by change of base.

Definition 5.6.1. For category $\mathcal{C}$ we define the category $\mathbf{Alg}(\mathcal{C})$ as follows:

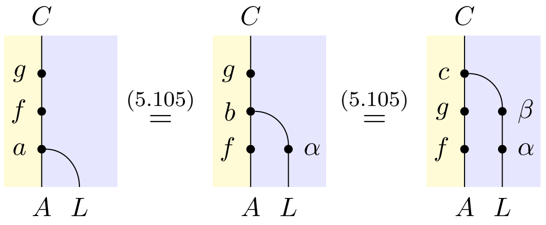
- **Objects**: Pairs consisting of:
  1. An endofunctor $L : \mathcal{C} \to \mathcal{C}$
  2. An L-algebra $\begin{pmatrix} LA \\ \downarrow a \\ A \end{pmatrix}$
- **Morphism**: A morphism $\begin{pmatrix} L, & \begin{matrix} LA \\ \downarrow a \\ A \end{matrix} \end{pmatrix} \to \begin{pmatrix} M, & \begin{matrix} MB \\ \downarrow b \\ B \end{matrix} \end{pmatrix}$ is a pair consisting of:
  1. A natural transformation $\alpha : L \Rightarrow M$
  2. A $\mathcal{C}$ morphism $h : A \to B$

  Such that:



(5.105)

Composition is defined componentwise in the obvious manner. That this is well defined follows from the equalities:



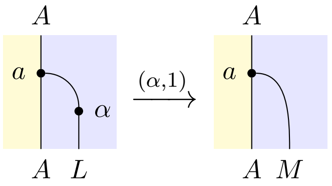
(5.106)

Lemma 5.6.2. *For a category $\mathcal{C}$ the forgetful functor:*

$$\begin{pmatrix} \mathbf{Alg}(\mathcal{C}) \\ \downarrow \\ [\mathcal{C}, \mathcal{C}] \end{pmatrix} \tag{5.107}$$

*is a split fibration. The fibre over L is isomorphic to the category L-**Alg** defined in example 2.1.1.*

Proof. For endofunctors $L, M : \mathcal{C} \to \mathcal{C}$, natural transformation $\alpha : L \Rightarrow M$ and algebra $\begin{pmatrix} MA \\ \downarrow a \\ A \end{pmatrix}$ a cartesian lifting is given by the morphism:



(5.108)

The existence and uniqueness of the cartesian morphism property can then be seen from the following diagram:

$$\begin{tikzcd}
\begin{pmatrix} NB \\ \downarrow b \\ B \end{pmatrix} \arrow[dr, "(\beta, h)"'] \arrow[drr, bend left, "(\alpha \circ \beta, h)"] & & \\
& \begin{pmatrix} LA \\ \downarrow a \circ \alpha_A \\ A \end{pmatrix} \arrow[r, "(\alpha, 1)"'] & \begin{pmatrix} MA \\ \downarrow a \\ A \end{pmatrix} \\
N \arrow[dr, "\beta"'] \arrow[drr, bend left, "\alpha \circ \beta"] & & \\
& M \arrow[r, "\alpha"'] & L
\end{tikzcd} \tag{5.109}$$

That these liftings give a splitting then follows immediately from the definition of composition in $\mathbf{Alg}(\mathcal{C})$. The fibre structure is then straightforward to check. □

Lemma 5.6.3. *For each category $\mathcal{C}$, there is a $\mathbf{Fib}_{split}([\mathcal{C}, \mathcal{C}])$ "forgetful" morphism to the constant fibration:*

$$\begin{pmatrix} \mathbf{Alg}(\mathcal{C}) \\ \downarrow p \\ [\mathcal{C}, \mathcal{C}] \end{pmatrix} \xrightarrow{V} \begin{pmatrix} [\mathcal{C}, \mathcal{C}] \times \mathcal{C} \\ \downarrow q \\ [\mathcal{C}, \mathcal{C}] \end{pmatrix} \tag{5.110}$$

Proof. We define V as follows:

$$\mathbf{Alg}(\mathcal{C}) \to [\mathcal{C},\mathcal{C}] \times \mathcal{C} \tag{5.111}$$

$$\left(L, \begin{array}{c} LA \\ \downarrow a \\ A \end{array}\right) \mapsto (L, A) \tag{5.112}$$

$$(\alpha, h) \mapsto (\alpha, h) \tag{5.113}$$

That this is functorial is immediate from the definition of composition in $\mathbf{Alg}(\mathcal{C})$. It is easy to see that $q \circ V = p$. To see that the splitting is preserved, consider a natural transformation $\alpha : L \Rightarrow M$ and $\mathbf{Alg}(\mathcal{C})$ object $\left(M, \begin{array}{c} MA \\ \downarrow a \\ A \end{array}\right)$. We have:

$$\begin{aligned}
& V\overline{\alpha}(M, \begin{array}{c} MA \\ \downarrow a \\ A \end{array}) \\
= \quad & \{ \ p\text{-liftings} \ \} \\
& V(\alpha, 1_A) \\
= \quad & \{ \ \text{definition} \ \} \\
& (\alpha, 1_A) \\
= \quad & \{ \ q\text{-liftings} \ \} \\
& \overline{\alpha}(V\left(M, \begin{array}{c} MA \\ \downarrow a \\ A \end{array}\right))
\end{aligned}$$

□

Definition 5.6.4. For categories $\mathcal{I}$ and $\mathcal{A}$, and endofunctor:

$$L : \mathcal{I} \times \mathcal{A} \to \mathcal{A} \tag{5.114}$$

we define split fibration $\left(\begin{array}{c} L_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{array}\right)$ via change of base as in the following pullback:

$$\begin{tikzcd}
L_{(-)}\text{-}\mathbf{Alg} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{Alg}(\mathcal{A}) \arrow[d] \\
\mathcal{I} \arrow[r, "\hat{L}"'] & {[\mathcal{A},\mathcal{A}]}
\end{tikzcd} \tag{5.115}$$

where $\hat{L}$ is the obvious functor:

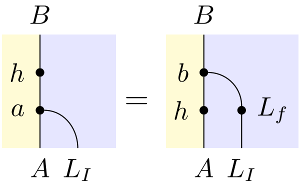
$$\hat{L}(I)(A) := L(I, A) \tag{5.116}$$

$$\hat{L}(f)(h) := L(f, h) \tag{5.117}$$

Concretely we describe the total category $L_{(-)}$-**Alg** as having:

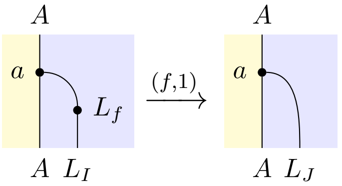
- **Objects**: Pairs consisting of:
  1. An $\mathcal{I}$ object I
  2. An L_I-algebra $\begin{pmatrix} L_I A \\ \downarrow a \\ A \end{pmatrix}$
- **Morphisms**: A morphism $\begin{pmatrix} I, & \begin{matrix} L_I A \\ \downarrow a \\ A \end{matrix} \end{pmatrix} \rightarrow \begin{pmatrix} J, & \begin{matrix} L_J B \\ \downarrow b \\ B \end{matrix} \end{pmatrix}$ is a pair consisting:
  1. An $\mathcal{I}$ morphism $f : I \rightarrow J$
  2. An $\mathcal{A}$ morphism $h : A \rightarrow B$

  such that:



(5.118)

The fibre over I is then L_I-**Alg**. The splitting is given explicitly for $\mathcal{I}$ morphism $f : I \rightarrow J$ and $L_{(-)}$-**Alg** object $\begin{pmatrix} J, & \begin{matrix} L_J A \\ \downarrow a \\ A \end{matrix} \end{pmatrix}$ as the lifting:



(5.119)

We will write L_f-**Alg** for the substitution functor induced by f.

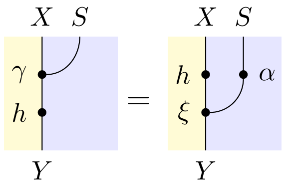
Recall change of base is functorial, by lemma 5.4.14. Using lemma 5.6.3 we see there is a split fibred "forgetful" functor:

$$V : \begin{pmatrix} L_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \rightarrow \begin{pmatrix} \mathcal{I} \times \mathcal{A} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \tag{5.120}$$

Definition 5.6.5. For category $\mathcal{C}$ we define the category $\mathbf{Coalg}(\mathcal{C})^{\bullet}$ as follows:

- **Objects**: Pairs consisting of:
  1. An endofunctor $T : \mathcal{C} \to \mathcal{C}$
  2. A T-coalgebra $\left(\begin{array}{c} X \\ \downarrow \gamma \\ TX \end{array}\right)$
- **Morphisms**: A morphism $\left(S, \begin{array}{c} X \\ \downarrow \gamma \\ SX \end{array}\right) \to \left(T, \begin{array}{c} Y \\ \downarrow \xi \\ TY \end{array}\right)$ is a pair consisting of:
  1. A natural transformation $\alpha : T \Rightarrow S$
  2. A $\mathcal{C}$ morphism $h : Y \to X$

  such that:

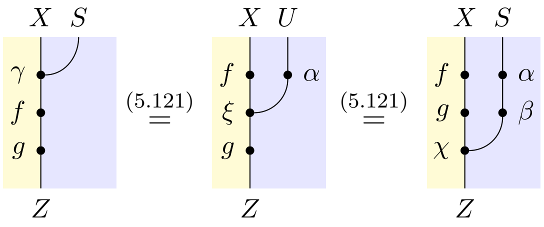


(5.121)

Identities are given componentwise, and composition is performed componentwise contravariantly:

$$(\beta, g) \circ (\alpha, f) := (\alpha \circ \beta, f \circ g) \tag{5.122}$$

That this is well defined follows from the equations:



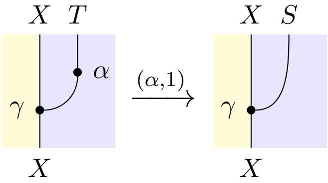
(5.123)

Lemma 5.6.6. *The forgetful functor:*

$$\begin{tikzcd} \mathbf{Coalg}(\mathcal{C})^{\bullet} \arrow[d] \\ {[\mathcal{C}, \mathcal{C}]^{op}} \end{tikzcd} \tag{5.124}$$

is a split fibration. The fibre over T is isomorphic to T-$\mathbf{Coalg}^{op}$.

Proof. For endofunctors $S, T : \mathcal{C} \to \mathcal{C}$, natural transformation $\alpha : S \Rightarrow T$ and coalgebra $\left(\begin{array}{c} X \\ \downarrow \gamma \\ SX \end{array}\right)$ a cartesian lifting is given by the morphism:



(5.125)

The existence and uniqueness of the cartesian morphism property can then be seen from the following diagram:

$$\begin{tikzcd}
\left(\begin{array}{c} Y \\ \downarrow \xi \\ UY \end{array}\right) \arrow[dr, "{(\beta, h)}"'] \arrow[drr, bend left, "{(\beta \circ \alpha, h)}"] & & \\
& \left(\begin{array}{c} X \\ \downarrow \alpha_X \circ \gamma \\ TX \end{array}\right) \arrow[r, "{(\alpha, 1)}"'] & \left(\begin{array}{c} X \\ \downarrow \gamma \\ SX \end{array}\right) \\
U \arrow[dr, "\beta"'] \arrow[drr, bend left, "\beta \circ \alpha"] & & \\
& T \arrow[r, "\alpha"'] & S
\end{tikzcd} \tag{5.126}$$

That these liftings give a splitting then follows immediately from the definition of composition in $\mathbf{Coalg}(\mathcal{C})^\bullet$. The fibre structure is then straightforward to check, noting the contravariant composition in the second component of the morphisms. □

Lemma 5.6.7. *For each category* $\mathcal{C}$*, there is a* $\mathbf{Fib}_{split}([\mathcal{C}, \mathcal{C}]^{op})$ *"forgetful" morphism to the constant fibration:*

$$\left(\begin{array}{c} \mathbf{Coalg}(\mathcal{C})^\bullet \\ \downarrow p \\ [\mathcal{C}, \mathcal{C}]^{op} \end{array}\right) \xrightarrow{U} \left(\begin{array}{c} [\mathcal{C}, \mathcal{C}]^{op} \times \mathcal{C}^{op} \\ \downarrow q \\ [\mathcal{C}, \mathcal{C}]^{op} \end{array}\right) \tag{5.127}$$

Proof. We define U as follows:

$$\mathbf{Coalg}(\mathcal{C})^{\bullet} \to [\mathcal{C},\mathcal{C}]^{op} \times \mathcal{C}^{op} \tag{5.128}$$

$$\left(T, \begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right) \mapsto (T, X) \tag{5.129}$$

$$(\alpha, h) \mapsto (\alpha, h) \tag{5.130}$$

That this is functorial follows from the definition of composition in $\mathbf{Coalg}(\mathcal{C})^{\bullet}$. It is easy to see that $q \circ U = p$. To see that the splitting is preserved, consider natural transformation $\alpha : T \Rightarrow S$ and $\mathbf{Coalg}(\mathcal{C})^{\bullet}$ object $\left(T, \begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right)$. We have:

$$\begin{array}{cl} & U\overline{\alpha}(T, \begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}) \\ = & \{ \ p\text{-liftings} \ \} \\ & U(\alpha, 1_X) \\ = & \{ \ \text{definition} \ \} \\ & (\alpha, 1_X) \\ = & \{ \ q\text{-liftings} \ \} \\ & \overline{\alpha}(U\left(T, \begin{array}{c} X \\ \downarrow\gamma \\ TX \end{array}\right)) \end{array}$$

□

Definition 5.6.8. For categories $\mathcal{I}$ and $\mathcal{X}$, and bifunctor:

$$T : \mathcal{I} \times \mathcal{X}^{op} \to \mathcal{X}^{op} \tag{5.131}$$

we define the endofunctor:

$$\overline{T} := \mathcal{I} \xrightarrow{\hat{T}} [\mathcal{X}^{op}, \mathcal{X}^{op}] \xrightarrow{\cong} [\mathcal{X}, \mathcal{X}]^{op} \tag{5.132}$$

Definition 5.6.9. For categories $\mathcal{I}$ and $\mathcal{X}$, and bifunctor:

$$T : \mathcal{I} \times \mathcal{X}^{op} \to \mathcal{X}^{op} \tag{5.133}$$

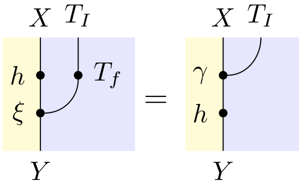
we define split fibration $\left(\begin{array}{c} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{array}\right)$ via change of base as in the pullback square:

$$\begin{tikzcd}
T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & \mathbf{Coalg}(\mathcal{X})^{\bullet} \arrow[d] \\
\mathcal{I} \arrow[r, "\overline{T}"'] & {[\mathcal{X},\mathcal{X}]^{op}}
\end{tikzcd} \tag{5.134}$$

Explicitly the category $T^{op}_{(-)}$-$\mathbf{Coalg}^{op}$ is given by:

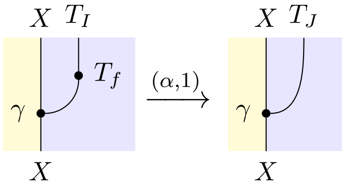
- **Objects**: Pairs consisting of:
  1. An $\mathcal{I}$ object I
  2. A T_I-coalgebra $\left(\begin{array}{c} X \\ \downarrow\gamma \\ T_I X \end{array}\right)$
- **Morphisms**: A morphism $\left(I, \begin{array}{c} X \\ \downarrow\gamma \\ T_I X \end{array}\right) \to \left(J, \begin{array}{c} Y \\ \downarrow\xi \\ T_J Y \end{array}\right)$ is a pair consisting of:
  1. An $\mathcal{I}$ morphism $f : I \to J$
  2. An $\mathcal{X}$ morphism $h : Y \to X$

  such that:



(5.135)

The fibre over I is then T^{op}_I-$\mathbf{Coalg}^{op}$, hence our slightly unusual choice of notation for the total category. The splitting is given explicitly for $\mathcal{I}$ morphism $f : I \to J$ and $T^{op}_{(-)}$-$\mathbf{Coalg}^{op}$ object $\left(J, \begin{array}{c} X \\ \downarrow\gamma \\ T_J X \end{array}\right)$ as the lifting:



(5.136)

We will write T^{op}_f-$\mathbf{Coalg}^{op}$ for the substitution functor induced by f.

Recall change of base is functorial, by lemma 5.4.14. Using lemma 5.6.7 we see there is a split fibred "forgetful" functor:

$$U : \left(\begin{array}{c} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{array}\right) \to \left(\begin{array}{c} \mathcal{I} \times \mathcal{X}^{op} \\ \downarrow \\ \mathcal{I} \end{array}\right) \tag{5.137}$$

Given any bifunctor:

$$T : \mathcal{I} \times \mathcal{C}^{op} \to \mathcal{C}^{op} \tag{5.138}$$

we have an isomorphism in $\mathfrak{Fib}(\mathcal{I})$:

$$\left(\begin{array}{c} T_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{array}\right) \cong \left(\begin{array}{c} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{array}\right) \tag{5.139}$$

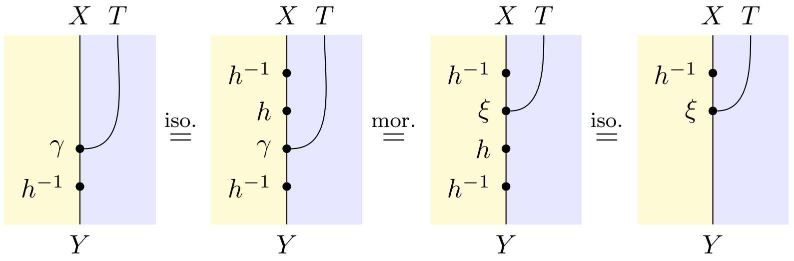
this is then the fibred analogue of equation 4.17, with isomorphisms on the fibres:

$$T_I\text{-}\mathbf{Alg} \cong T_I^{op}\text{-}\mathbf{Coalg}^{op} \tag{5.140}$$

We now consider lifting dual adjunctions and equivalences to our fibred (co)algebraic setting. This will require a simple fact about the coalgebra forgetful functor, which we prove as a short exercise with string diagrams:

Lemma 5.6.10. *For any endofunctor $T : \mathcal{C} \to \mathcal{C}$ the forgetful functor $U : T\text{-}\mathbf{Coalg} \to \mathcal{C}$ reflects isomorphisms.*

Proof. Assuming a T-coalgebra morphism $\left(\begin{array}{c} X \\ \downarrow \gamma \\ TX \end{array}\right) \xrightarrow{h} \left(\begin{array}{c} Y \\ \downarrow \xi \\ TY \end{array}\right)$ such that h is an isomorphism in $\mathcal{C}$, we have the following equalities:



(5.141)

□

Theorem 5.6.11 (Lifting)**.** *Let $\mathcal{A}, \mathcal{X}$ and $\mathcal{I}$ be arbitrary categories and $F : \mathcal{X}^{op} \to \mathcal{A}$ a contravariant functor.*

1. *Given a natural transformation* δ

$$\delta_{I,X} : L(I, FX) \Rightarrow F \circ T(I, X) \tag{5.142}$$

natural in both I *and* X*, then* F *lifts to a* $\mathfrak{Fib}_{split}(\mathcal{I})$ *morphism:*

$$\begin{pmatrix} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \xrightarrow{\overline{F}} \begin{pmatrix} L_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \tag{5.143}$$

2. *If additionally* F *has a right adjoint* G *as in:*

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "G"] \arrow[r, phantom, "\top"] & \mathcal{X}^{op} \arrow[l, bend left, "F"] \end{tikzcd} \tag{5.144}$$

and δ *is invertible, then the adjunction lifts to an adjunction in* $\mathfrak{Fib}_{split}(\mathcal{I})$*:*

$$\begin{tikzcd} \begin{pmatrix} L_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \arrow[r, bend left, "\overline{G}"] \arrow[r, phantom, "\top"] & \begin{pmatrix} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \arrow[l, bend left, "\overline{F}"] \end{tikzcd} \tag{5.145}$$

3. *Under the conditions of part 1 the following diagram commutes on the nose in* $\mathfrak{Fib}_{split}(\mathcal{I})$*:*

$$\begin{tikzcd} \begin{pmatrix} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \arrow[r, "\overline{F}"] \arrow[d, "U"'] & \begin{pmatrix} L_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \arrow[d, "V"] \\ \begin{pmatrix} \mathcal{I} \times \mathcal{X}^{op} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \arrow[r, "F"'] & \begin{pmatrix} \mathcal{I} \times \mathcal{A} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \end{tikzcd} \tag{5.146}$$

where V *and* U *are the functors defined in lemmas 5.6.3 and 5.6.7.*

4. *Under the conditions of part 2 the following diagram commutes on the nose in*

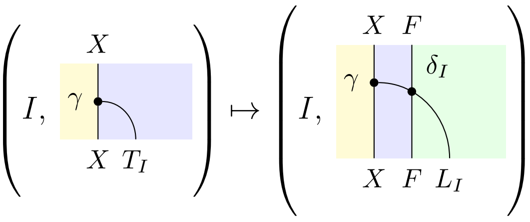
$\mathfrak{Fib}_{split}(\mathcal{I})$:

$$\begin{tikzcd}
\begin{pmatrix} L_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \arrow[r, "\overline{G}"] \arrow[d, "V"'] & \begin{pmatrix} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \arrow[d, "U"] \\
\begin{pmatrix} \mathcal{I} \times \mathcal{A} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \arrow[r, "G"'] & \begin{pmatrix} \mathcal{I} \times \mathcal{X}^{op} \\ \downarrow \\ \mathcal{I} \end{pmatrix}
\end{tikzcd} \tag{5.147}$$

5. *Under the conditions of part 2:*

   - *If the unit of the adjunction $F \dashv G$ is an isomorphism, the unit of the adjunction $\overline{F} \dashv \overline{G}$ is also an isomorphism*
   - *If the counit of the adjunction $F \dashv G$ is an isomorphism, the counit of the adjunction $\overline{F} \dashv \overline{G}$ is also an isomorphism*

Proof. For part 1. We define the action of $\overline{F}$ on objects as:

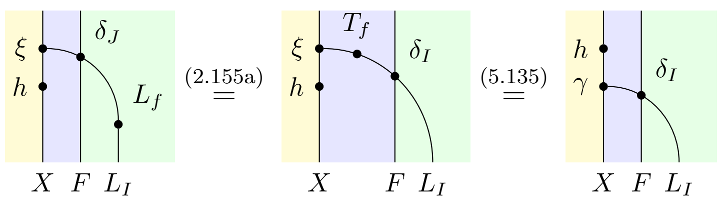


(5.148)

The action of $\overline{F}$ on morphisms is:

$$(f, h) \mapsto (f, Fh) \tag{5.149}$$

To see this is well defined we consider the following equations:

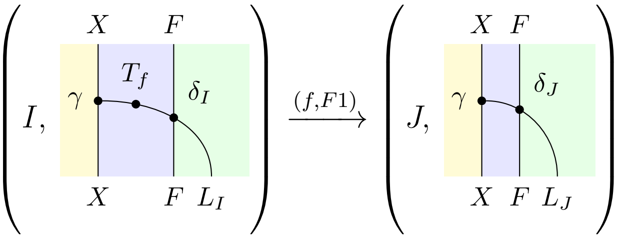


(5.150)

Preservation of identities and composition is easy to see from the contravariant functoriality of F.

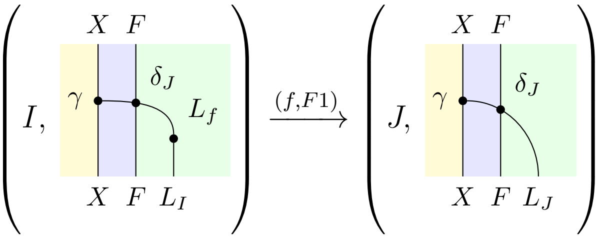
To confirm that the splitting is preserved "on the nose", as described in definition

5.4.10, we consider $\mathcal{I}$ morphism $f : I \to J$ and $T^{op}_{(-)}$-$\mathbf{Coalg}^{op}$ object $\left(J, \begin{array}{c} X \\ \downarrow\gamma \\ T_J X \end{array}\right)$. Lifting and then applying the functor $\overline{F}$ gives the morphism:



(5.151)

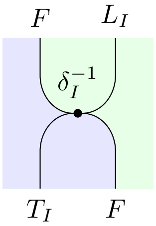
Applying $\overline{F}$ and then taking the cartesian lifting gives the morphism:



(5.152)

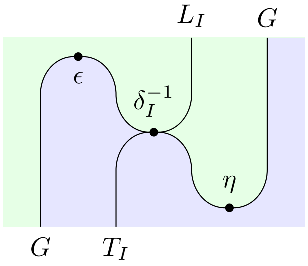
We then note the domains are equal by naturality of δ_I in I as in equation (2.155a).

For part 2, we now assume δ is invertible so there is a family of natural transformations indexed by objects of $\mathcal{I}$:



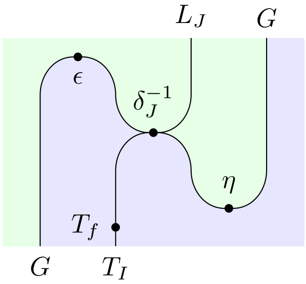
(5.153)

Using the unit η and counit ϵ of the adjunction $F \dashv G$ we can construct the mate of each δ_I^{-1}:

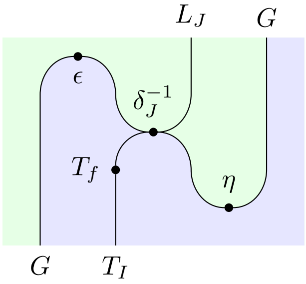


(5.154)

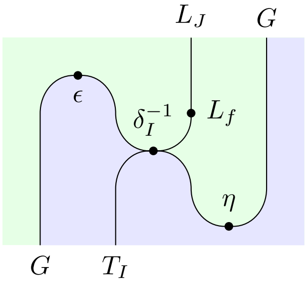
This family of natural transformations is natural in the parameter I, as can be seen from the following equalities:



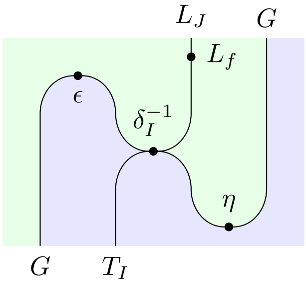
$=$ { naturality }



$=$ { naturality in I }

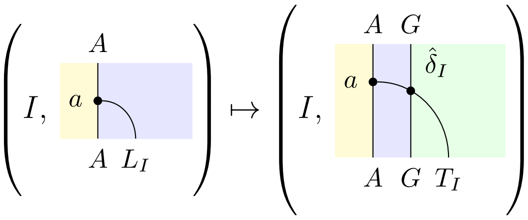


$=$ { naturality }



We define $\hat{\delta}$ as this composite natural transformation. We can use $\hat{\delta}$ to construct $\overline{G}$ in a similar way to the construction of $\overline{F}$ is part 1. Specifically, we define the of $\overline{G}$

on objects as:



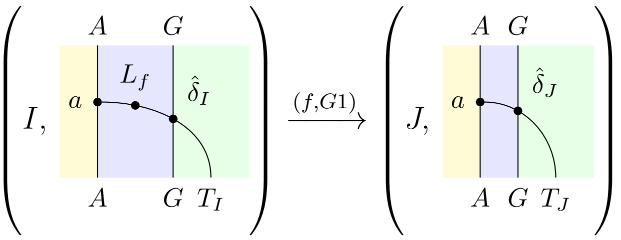
(5.155)

The action of $\overline{G}$ on morphisms is:

$$(f, h) \mapsto (f, Gh) \tag{5.156}$$

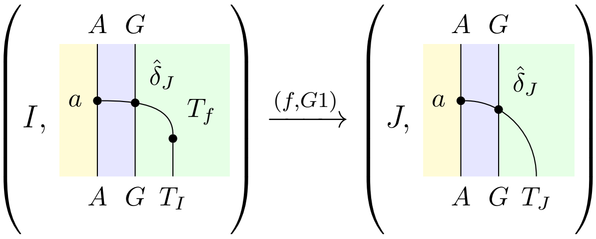
That this is well defined follows in the same way as in the proof for $\overline{F}$, exploiting the naturality of $\hat{\delta}_I$ in I. Preservation of identities and composition is again straightforward. It remains to check the splitting is preserved, this proceeds similarly to the lifting $\overline{F}$.

Lifting and then applying the functor $\overline{G}$ gives the morphism:



(5.157)

Applying $\overline{G}$ and then taking the cartesian lifting gives the morphism:

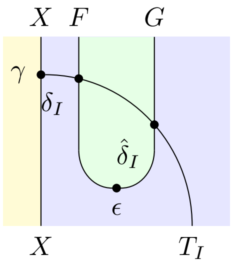


(5.158)

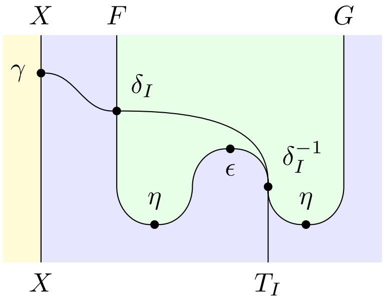
Again, we then observe the domains are equal by naturality of $\hat{\delta}_I$ in I as in equation (2.155a).

If order to show that $\overline{F} \dashv \overline{G}$ we aim to show a suitable unit and counit with vertical components.

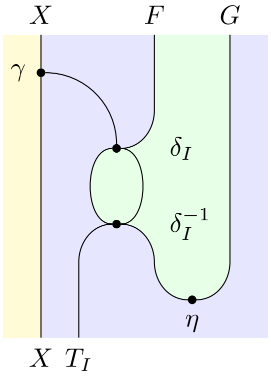
For the unit we have equations:



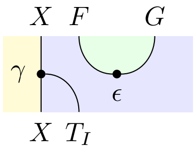
$= \quad \{$ definition of $\hat{\delta}_I$ $\}$



$= \quad \{$ equation (2.23a) $\}$

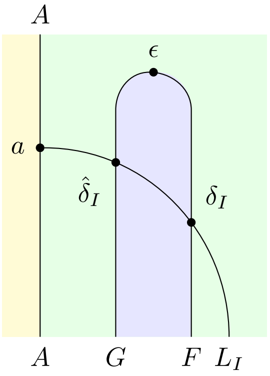


$= \quad \{$ inverses $\}$

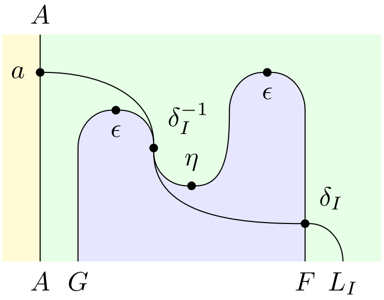


This shows that the components η are coalgebra homomorphisms and we can therefore conclude that the unit of the adjunction $F \dashv G$ lifts to a unit for the adjunction $\overline{F} \dashv \overline{G}$, with vertical components.

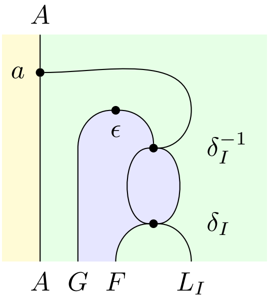
Similarly for the counit we have equations:



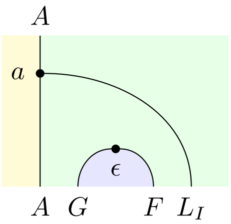
$$= \quad \{ \text{ definition of } \hat{\delta}_I \text{ } \}$$



$$= \quad \{ \text{ equation (2.23a) } \}$$



$$= \quad \{ \text{ inverses } \}$$



This shows that the components of ϵ are algebra homomorphisms and therefore the counit from the adjunction $F \dashv G$ lifts to a counit for the adjunction $\overline{F} \dashv \overline{G}$, with vertical components. Part 5 then follows from lemma 5.6.10.

Parts 3 and 4 are immediate from the definitions of $\overline{F}$, $\overline{G}$, V and U. ☐

We note that we have phrased the lifting results of theorem 5.6.11 as a contravariant relationship between fibrations of coalgebras and algebras, anticipating our later logical applications. In other settings this theorem could be rephrased as a lifting result for functors between fibrations of algebras. A dual lifting result for fibrations of coalgebras is also clearly possible, we do not pursue this explicitly as it will not be needed in subsequent sections.

We are now in a position to describe a parameterized analogue of the equivalence relating algebras and coalgebras in equation (4.20).

Theorem 5.6.12. *Let $\mathcal{I}, \mathcal{A}, \mathcal{X}$ be categories. Assume there exists a bifunctor:*

$$T : \mathcal{I} \times \mathcal{X}^{op} \to \mathcal{X}^{op} \tag{5.159}$$

and a dual equivalence:

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "G"] \arrow[r, phantom, "\simeq"] & \mathcal{X}^{op} \arrow[l, bend left, "F"] \end{tikzcd} \tag{5.160}$$

Then there exists a bifunctor:

$$L : \mathcal{I} \times \mathcal{A} \to \mathcal{A} \tag{5.161}$$

such that there is an equivalence:

$$\begin{pmatrix} L_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \simeq \begin{pmatrix} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \tag{5.162}$$

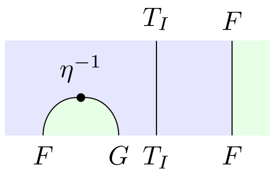
in $\mathfrak{Fib}(\mathcal{I})$.

Proof. We define the functor $L : \mathcal{I} \times \mathcal{A} \to \mathcal{A}$ as the composite:

$$\mathcal{I} \times \mathcal{A} \xrightarrow{1 \times G} \mathcal{I} \times \mathcal{X}^{op} \xrightarrow{T} \mathcal{X}^{op} \xrightarrow{F} \mathcal{A} \tag{5.163}$$

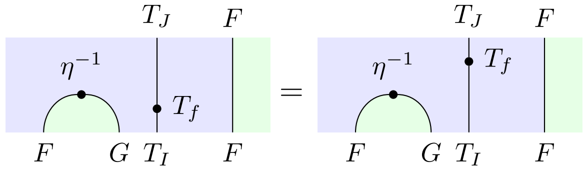
We define the components of an I indexed family of natural isomorphisms $L(I, F(-)) \Rightarrow$

$F \circ T(I, -)$ as:



(5.164)

Naturality in the first parameter then follows from:



(5.165)

We can then use theorem 5.6.11 to lift the dual equivalence $F \dashv G$ to a dual equivalence $\overline{F} \dashv \overline{G}$. □

5.7 Parameterized Logical Connections

Exploiting the parameterized algebras and coalgebras of section 5.6, we now generalize the notion of a logical connection in definition 4.4.1 to the parameterized setting.

Definition 5.7.1 (Parameterized Logical Connection)**.** Given a dual adjunction:

$$\begin{tikzcd}
\mathcal{A} \arrow[r, bend left, "\mathrm{Pt}"] \arrow[r, phantom, "\perp"] & \mathcal{X}^{op} \arrow[l, bend left, "\Omega"]
\end{tikzcd} \tag{5.166}$$

a **parameterized logical connection** consists of the following data:

- A **parameterized coalgebraic signature bifunctor**:

$$T : \mathcal{I} \times \mathcal{X}^{op} \to \mathcal{X}^{op} \tag{5.167}$$

- A **parameterized logical signature bifunctor**:

$$L : \mathcal{I} \times \mathcal{A} \to \mathcal{A} \tag{5.168}$$

- A family of $\mathcal{A}$ morphisms:

  $$\delta_{I,X} : L(I, \Omega(X)) \to \Omega \circ T(I, X) \tag{5.169}$$

  where Ω is the right adjoint of the adjunction described above. These morphisms are natural in both I and X.

To avoid lots of repetition, throughout the remainder of this section we will implicitly assume the existence of a parameterized logical connection with definitions and notation as in definition 5.7.1.

We note that parameterized logical connections are simply the distributive laws in part 1 of theorem 5.6.11. The change of name is intended to emphasise their logical significance. We can therefore see parameterized logical connections as defining a suitable functor relating coalgebraic models to algebraic semantics for a corresponding logic.

Definition 5.7.2. For a fixed parameterized logical connection δ, we define the **algebraization** functor as the split fibred functor:

$$\mathbf{Alg}^{\delta} : \begin{pmatrix} T^{op}_{(-)}\text{-}\mathbf{Coalg}^{op} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \to \begin{pmatrix} L_{(-)}\text{-}\mathbf{Alg} \\ \downarrow \\ \mathcal{I} \end{pmatrix} \tag{5.170}$$

defined in theorem 5.6.11.

5.7.1 Semantics

We will consider a few slightly different formulations of semantics for logics built upon parameterized logical connections. The variations considered will illustrate different points and provide some elements of technical freedom in their scope of applicability. The various forms of semantics are parameterized versions of the usual formulations using (unparameterized) logical connections in the literature.

5.7.1.1 Initial Algebra Semantics

The simplest way to provide semantics for our logic is to assume the existence of suitable initial algebras. Throughout this section we assume that for each $\mathcal{I}$ object I, an initial algebra $\begin{pmatrix} L_I\,\mathrm{Form}_I \\ \downarrow in_I \\ \mathrm{Form}_I \end{pmatrix}$ exists. Conceptually we view Form_I to be an object representing (equivalence classes) of formulae.

Definition 5.7.3 (Initial Algebra Semantics)**.** For an arbitrary T_I-coalgebra $\begin{pmatrix} X \\ \downarrow\gamma \\ T_I X \end{pmatrix}$ we have a unique morphism given by the universal property of initial objects:

$$\begin{pmatrix} L_I \,\mathrm{Form}_I \\ \downarrow in_I \\ \mathrm{Form}_I \end{pmatrix} \xrightarrow{!_\gamma} \mathbf{Alg}^{\delta}{}_I \begin{pmatrix} X \\ \downarrow\gamma \\ T_I X \end{pmatrix} \tag{5.171}$$

We then define the **semantics morphism** as the underlying $\mathcal{A}$ morphism:

$$[\![-]\!]_{I,\gamma} := V!_\gamma \tag{5.172}$$

We will also need to relate formulae in different fibres.

Definition 5.7.4 (Initial Algebra Formula Translation)**.** By the universal property of initial objects, for each $\mathcal{I}$ morphism $f : I \to J$ there is a unique morphism:

$$\begin{pmatrix} L_I \,\mathrm{Form}_I \\ \downarrow in_I \\ \mathrm{Form}_I \end{pmatrix} \xrightarrow{!_f} L_f\text{-}\mathbf{Alg} \begin{pmatrix} L_J \,\mathrm{Form}_J \\ \downarrow in_J \\ \mathrm{Form}_J \end{pmatrix} \tag{5.173}$$

We then define the **formula translation morphism** as the underlying morphism:

$$\mathrm{Form}_f := V!_f \tag{5.174}$$

Lemma 5.7.5. *The formula translations in definition 5.7.4 extend to a functor:*

$$\mathrm{Form}_{(-)} : \mathcal{I} \to \mathcal{A} \tag{5.175}$$

$$I \mapsto \mathrm{Form}_I \tag{5.176}$$

$$f \mapsto \mathrm{Form}_f \tag{5.177}$$

Proof. Follows immediately from the universal property of initial objects. □

5.7.1.2 Free Algebra Semantics

We now consider extending the semantics of section 5.7.1.1 so that our logic supports proposition variables. In order to do this we will require a suitable definition of valuation. Throughout this section we will assume that there exists a forgetful functor $W : \mathcal{A} \to \mathbf{Set}$ with left adjoint $F : \mathbf{Set} \to \mathcal{A}$. We also assume the existence of free L_I-algebras as necessary.

Definition 5.7.6 (Valuations)**.** For a given logical connection, a **proposition variables functor** is a functor:

$$\mathrm{Prop}_{(-)} : \mathcal{I} \to \mathbf{Set} \tag{5.178}$$

Let I be a $\mathcal{I}$ object and $\left(\begin{array}{c} X \\ \downarrow\gamma \\ T_I X \end{array}\right)$ be a T_I-coalgebra. A **valuation** is a morphism:

$$v : \mathrm{Prop}_I \to WV\mathbf{Alg}^\delta \left(\begin{array}{c} X \\ \downarrow\gamma \\ T_I X \end{array}\right) \tag{5.179}$$

We denote the transpose under the adjunction of this morphism as:

$$\hat{v} : F\,\mathrm{Prop}_I \to V\mathbf{Alg}^\delta \left(\begin{array}{c} X \\ \downarrow\gamma \\ T_I X \end{array}\right) \tag{5.180}$$

Definition 5.7.7 (Free Algebra Semantics)**.** For an $\mathcal{I}$ object I and T_I-coalgebra $\left(\begin{array}{c} X \\ \downarrow\gamma \\ T_I X \end{array}\right)$ with corresponding valuation v, we define the **semantics morphism** in context I using the universal property of free L_I-algebras:

$$\begin{tikzcd}
F\,\mathrm{Prop}_I \arrow[r, "\eta^I_{F\,\mathrm{Prop}_I}"] \arrow[dr, "\hat{v}"'] & V\left(\begin{array}{c} L_I\,\mathrm{Form}(\mathrm{Prop})_I \\ \downarrow \mathit{free}_I \\ \mathrm{Form}(\mathrm{Prop})_I \end{array}\right) \arrow[d, "\llbracket - \rrbracket_{I,(X,\gamma),v}"] \\
 & V\mathbf{Alg}^\delta{}_I \left(\begin{array}{c} X \\ \downarrow\gamma \\ T_I X \end{array}\right)
\end{tikzcd} \tag{5.181}$$

We also will need to be able to relate formulae in different contexts:

Definition 5.7.8. For $\mathcal{I}$ morphism $f : I \to J$ we define the **formula translation**

morphism, $\operatorname{form}(\mathrm{Prop})_f$, using the universal property of free L_I-algebras:

$$\begin{tikzcd}
F\,\mathrm{Prop}_I \arrow[r, "\eta^I_{F\,\mathrm{Prop}_I}"] \arrow[d, "F\,\mathrm{Prop}_f"] & V\begin{pmatrix} L_I\,\mathrm{Form}(\mathrm{Prop})_I \\ \downarrow \mathit{free}_I \\ \mathrm{Form}(\mathrm{Prop})_I \end{pmatrix} \arrow[d, "V\,\mathrm{form}(\mathrm{Prop})_f"] \\
F\,\mathrm{Prop}_J \arrow[r, "\eta^J_{F\,\mathrm{Prop}_J}"'] & V\begin{pmatrix} L_J\,\mathrm{Form}(\mathrm{Prop})_J \\ \downarrow \mathit{free}_J \\ \mathrm{Form}(\mathrm{Prop})_J \end{pmatrix} = V_I L_f\text{-}\mathbf{Alg}\begin{pmatrix} L_J\,\mathrm{Form}(\mathrm{Prop})_J \\ \downarrow \mathit{free}_J \\ \mathrm{Form}(\mathrm{Prop})_J \end{pmatrix}
\end{tikzcd} \tag{5.182}$$

The equality in the diagram follows from lemma 5.6.7. We then define:

$$\mathrm{Form}(\mathrm{Prop})_f := V\,\mathrm{form}(\mathrm{Prop})_f \tag{5.183}$$

Lemma 5.7.9. *The formula translation morphisms of definition 5.7.8 extend to a functor:*

$$\mathrm{Form}(\mathrm{Prop})_{(-)} : \mathcal{I} \to \mathcal{A} \tag{5.184}$$
$$I \mapsto \mathrm{Form}(\mathrm{Prop})_I \tag{5.185}$$
$$f : I \to J \mapsto \mathrm{Form}(\mathrm{Prop})_f \tag{5.186}$$

Proof. The claim follows from applying the universal property of free objects. □

5.7.1.3 Initial Sequence Semantics

We can vary our assumptions in a different direction by dropping the assumption of the existence of initial or free algebras. We instead define our semantics in terms of the initial sequences of the algebraic signature functors, assuming the existence of an initial object 0 in $\mathcal{A}$.

Definition 5.7.10 (Depth n formulae)**.** We define the depth n I-formulae $\mathcal{A}$ object, Form^n_I, inductively as follows:

$$\mathrm{Form}^0_I := 0 \tag{5.187}$$
$$\mathrm{Form}^{n+1}_I := L_I\,\mathrm{Form}^n_I \tag{5.188}$$

Definition 5.7.11 (Initial Sequence Semantics)**.** Let I be an $\mathcal{I}$ object, and $\begin{pmatrix} X \\ \downarrow\gamma \\ T_I X \end{pmatrix}$

a T_I-coalgebra. We inductively define an family of **semantics morphisms** $[\![-]\!]^n_{I,\gamma}$: $\mathrm{Form}^n_I \to V \circ \mathbf{Alg}^\delta(\gamma)$ in $\mathcal{A}$:

$$[\![-]\!]^0_{I,\gamma} := 0 \xrightarrow{!} V\mathbf{Alg}^\delta(\gamma) \tag{5.189}$$

$$[\![-]\!]^{n+1}_{I,\gamma} := \mathbf{Alg}^\delta(\gamma) \circ L_I[\![-]\!]^n_{I,\gamma} \tag{5.190}$$

As always, we will also need to be able to relate formulae in different fibres.

Definition 5.7.12 (Formula Translation)**.** For $\mathcal{I}$ morphism $f : I \to J$ we inductively define **formula translation morphisms** $\mathrm{Form}^n_f : \mathrm{Form}^n_I \to \mathrm{Form}^n_J$ in $\mathcal{A}$ as:

$$\mathrm{Form}^0_f := 1_0 \tag{5.191}$$

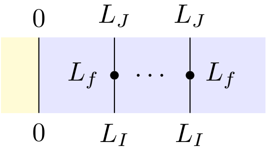
$$\mathrm{Form}^{n+1}_f := L_J \,\mathrm{Form}^n_f \circ L(f, \mathrm{Form}_n I) \tag{5.192}$$

Lemma 5.7.13. *The formula translation morphisms:*

$$\mathrm{Form}^n_{(-)} : \mathcal{I} \to \mathcal{A} \tag{5.193}$$

of definition 5.7.12 are functorial.

Proof. In terms of string diagrams, the n^{th} formula translation morphism can be drawn as:



(5.194)

where the ellipsis represents n-fold repetition. Functoriality is then easy to see from the diagrammatic properties of bifunctors in section 2.7. ☐

5.7.1.4 Free Sequence Semantics

We can also extend the initial sequence semantics of section 5.7.1.3 to support proposition variables, similarly to the approach in section 5.7.1.2. Throughout this section, we assume the required limits and colimits used in the definitions exist. As in section 5.7.1.2, will also assume that there exists a forgetful functor $W : \mathcal{A} \to \mathbf{Set}$ with left adjoint $F : \mathbf{Set} \to \mathcal{A}$. Proposition variable functors and valuations are as in definition 5.7.6.

For our fixed logical connection, it will be convenient to define an auxiliary bifunctor:

Definition 5.7.14. We define bifunctor $\underline{L} : \mathcal{I} \times \mathcal{A} \to \mathcal{A}$ as follows:

$$\underline{L}(I, X) := L(I, X) + F\,\mathrm{Prop}_I \tag{5.195}$$

$$\underline{L}(f, g) := L(f, g) + F\,\mathrm{Prop}_f \tag{5.196}$$

For $\mathcal{I}$ object I we then define the depth n I-formula $\mathcal{A}$ object, $\mathrm{Form}^n_I(\mathrm{Prop})$, as follows:

$$\mathrm{Form}^0_I(\mathrm{Prop}) := 0 \tag{5.197}$$

$$\mathrm{Form}^{n+1}_I(\mathrm{Prop}) := \underline{L}_I\,\mathrm{Form}^n_I(\mathrm{Prop}) \tag{5.198}$$

Definition 5.7.15 (Free Sequence Semantics)**.** Let I be an $\mathcal{I}$ object, $\begin{pmatrix} X \\ \downarrow\gamma \\ T_I X \end{pmatrix}$ a T_I-coalgebra and v a valuation. We inductively define an indexed family of semantics morphisms $[\![-]\!]^n_{I,\gamma,v} : \mathrm{Form}^n_I(\mathrm{Prop}) \to V \circ \mathbf{Alg}^\delta(\gamma)$:

$$[\![-]\!]^0_{I,(X,\gamma),v} := 0 \xrightarrow{!} V\mathbf{Alg}^\delta(\gamma) \tag{5.199}$$

$$[\![-]\!]^{n+1}_{I,(X,\gamma),v} := \left[\mathbf{Alg}^\delta(\gamma), \hat{v}\right] \circ \underline{L}_I[\![-]\!]^n_{I,(X,\gamma),v} \tag{5.200}$$

Again we require the ability to relate formulae in different fibres.

Definition 5.7.16. For $\mathcal{I}$ morphism $f : I \to J$ we inductively define a family of **formula translation morphisms**, $\mathrm{Form}^n_f(\mathrm{Prop}) : \mathrm{Form}^n_I(\mathrm{Prop}) \to \mathrm{Form}^n_J(\mathrm{Prop})$:

$$\mathrm{Form}^0_f(\mathrm{Prop}) := 1_0 \tag{5.201}$$

$$\mathrm{Form}^{n+1}_f(\mathrm{Prop}) := \underline{L}_J\,\mathrm{Form}^n_f(\mathrm{Prop}) \circ \underline{L}(f, \underline{L}^n_I 0) \tag{5.202}$$

Lemma 5.7.17. *The formula translation morphisms:*

$$\mathrm{Form}^n_{(-)}(\mathrm{Prop}) : \mathcal{I} \to \mathcal{A} \tag{5.203}$$

of definition 5.7.16 are functorial.

Proof. The proof is similar to that of lemma 5.7.13. ☐

5.7.2 Institution Type Conditions

The paper [Goguen and Burstall, 1992] introduces the notion of an **institution** to capture the relationship between different logical systems. Key components are transformations of models and formulae that relate the different logical systems. A central assumption is that these transformations are related informally as follows:

> Transforming a formula forwards and evaluating its semantics on a model is the same thing as transforming the model backwards and evaluating the semantics of the original formula.

So we have a contravariant relationship between formula transformations and model transformations; we will refer to similar, but more abstract relationships in the following sections as **institution conditions**. Previous work relating coalgebras to institutions is given in [Pattinson, 2002].

The results in theorems 5.7.18, 5.7.19, 5.7.20 and 5.7.21 are closely related. We spend the time to develop all four as they reflect different aspects of the algebraic semantics, and show that the existence of these institution conditions is reasonably stable under varying assumptions. The initial algebra semantics are possibly the most easy to motivate and understand. We start with these assumptions in section 5.7.2.1 and proceed roughly in order of increasing complexity in the subsequent sections.

5.7.2.1 Initial Algebra Semantics

We now demonstrate the existence of a suitable form of institution condition for the initial algebra semantics of section 5.7.1.1.

Theorem 5.7.18. *Let* $\begin{pmatrix} X \\ \downarrow\gamma \\ T_J \end{pmatrix}$ *be an arbitrary* T_J*-coalgebra and* $f : I \to J$ *an* $\mathcal{I}$ *morphism. Using the definitions of section 5.7.1.1:*

$$[\![-]\!]_{I,T_f^{op}\text{-}\mathbf{Coalg}^{op}(\gamma)} = [\![-]\!]_{J,\gamma} \circ \mathrm{Form}_f \tag{5.204}$$

Proof. The following diagram commutes in L_I-**Alg**, where each of the morphisms ! is given by the universal property of the initial object in the appropriate fibre:

$$\begin{tikzcd}
\begin{pmatrix} L_I\,\mathrm{Form}_I \\ \downarrow in_I \\ \mathrm{Form}_I \end{pmatrix} \arrow[r, "!"] \arrow[dr, "!"'] & L_f\text{-}\mathbf{Alg}\begin{pmatrix} L_J\,\mathrm{Form}_J \\ \downarrow in_J \\ \mathrm{Form}_J \end{pmatrix} \arrow[d, "L_f\text{-}\mathbf{Alg}(!)"] \\
& L_f\text{-}\mathbf{Alg} \circ \mathbf{Alg}^{\delta}\begin{pmatrix} X \\ \downarrow\gamma \\ T_J X \end{pmatrix}
\end{tikzcd} \tag{5.205}$$

Applying V, using definitions 5.7.3 and 5.7.4, and noting algebraization is a fibred functor:

$$L_f\text{-}\mathbf{Alg} \circ \mathbf{Alg}^{\delta} = \mathbf{Alg}^{\delta} \circ T_f^{op}\text{-}\mathbf{Coalg}^{op} \tag{5.206}$$

we have the following diagram in $\mathcal{A}$:

$$\begin{tikzcd}
V\begin{pmatrix} L_I \,\mathrm{Form}_I \\ \downarrow in_I \\ \mathrm{Form}_I \end{pmatrix} \arrow[r, "\mathrm{Form}_f"] \arrow[dr, "\llbracket - \rrbracket_{I,\gamma}"'] & V L_f\text{-}\mathbf{Alg}\begin{pmatrix} L_J \,\mathrm{Form}_J \\ \downarrow in_J \\ \mathrm{Form}_J \end{pmatrix} \arrow[d, "V \circ L_f\text{-}\mathbf{Alg}(!)"] \\
 & V \circ \mathbf{Alg}^{\delta} \circ T_f^{op}\text{-}\mathbf{Coalg}^{op}\begin{pmatrix} X \\ \downarrow \gamma \\ T_J X \end{pmatrix}
\end{tikzcd} \tag{5.207}$$

Finally we note that as the forgetful functor V is fibred over the constant fibration:

$$V \circ L_f\text{-}\mathbf{Alg}(!) = V! = \llbracket - \rrbracket_{J,\gamma} \tag{5.208}$$

$\square$

5.7.2.2 Free Algebra Semantics

The reasoning of section 5.7.2.1 extends to the setting in which we have proposition variables. In this case the model transformation includes a modification of the corresponding valuation. We now prove an institution type condition for the semantics of section 5.7.1.2.

Theorem 5.7.19. *Let* $\begin{pmatrix} X \\ \downarrow \gamma \\ T_J(X) \end{pmatrix}$ *be a* T_J*-coalgebra,* $f : I \to J$ *an* $\mathcal{I}$ *morphism and* $v : \mathrm{Prop}_J \to \Omega(X)$ *a valuation. Using the definitions of section 5.7.1.2:*

$$\llbracket - \rrbracket_{I, T_f^{op}\text{-}\mathbf{Coalg}^{op}(X,\gamma), v \circ F\,\mathrm{Prop}_f} = \llbracket - \rrbracket_{J,(X,\gamma),v} \circ \mathrm{Form}(\mathrm{Prop})_f \tag{5.209}$$

Proof. Consider the diagram in figure 5.7.2.2. The upper rectangle is derived from the definition of the formula translation morphisms. The lower triangle is from the definition of the semantics morphism. The equalities described in the diagram follow from lemma 5.6.3 and $\mathbf{Alg}^{\delta}$ being a split fibred functor. The claim then follows from the definition of the semantics morphism and the universal property of free algebras. $\square$

$$\begin{tikzcd}
F\,\mathrm{Prop}_I \arrow[r, "\eta^I_{F\,\mathrm{Prop}_I}"] \arrow[d, "F\,\mathrm{Prop}_f"'] & V\begin{pmatrix} L_I\,\mathrm{Form}(\mathrm{Prop})_I \\ \downarrow \mathit{free}_I \\ \mathrm{Form}(\mathrm{Prop})_I \end{pmatrix} \arrow[d, "\mathrm{Form}(\mathrm{Prop})_f"] \\
F\,\mathrm{Prop}_J \arrow[r, "\eta^J_{F\,\mathrm{Prop}_J}"] \arrow[dr, "\hat{v}"'] & V\begin{pmatrix} L_J\,\mathrm{Form}(\mathrm{Prop})_J \\ \downarrow \mathit{free}_J \\ \mathrm{Form}(\mathrm{Prop})_J \end{pmatrix} = V \circ L_f\text{-}\mathbf{Alg}\begin{pmatrix} L_J\,\mathrm{Form}(\mathrm{Prop})_J \\ \downarrow \mathit{free}_J \\ \mathrm{Form}(\mathrm{Prop})_J \end{pmatrix} \arrow[d, "V[\![-]\!]_{J,(X,\gamma),v} = V \circ L_f\text{-}\mathbf{Alg}[\![-]\!]_{J,(X,\gamma),v}"] \\
& V \circ \mathbf{Alg}^{\delta}\begin{pmatrix} X \\ \downarrow \gamma \\ T_J(X) \end{pmatrix} = V \circ \mathbf{Alg}^{\delta} \circ T_f^{op}\text{-}\mathbf{Coalg}^{op}\begin{pmatrix} X \\ \downarrow \gamma \\ T_J(X) \end{pmatrix}
\end{tikzcd}$$

Figure 5.1: Commuting diagram for theorem 5.7.19

5.7.2.3 Initial Sequence Semantics

We now see that if we change our semantics to the formulation in section 5.7.1.3 we obtain a family of institution conditions, paralleling the stratified nature of the semantics.

Theorem 5.7.20. *Let* $\begin{pmatrix} X \\ \downarrow \gamma \\ T_J \end{pmatrix}$ *be a* T_J*-coalgebra and* $f : I \to J$ *a* $\mathcal{I}$ *morphism. Using the definitions in section 5.7.1.3, for each* $n \geq 0$*:*

$$[\![-]\!]^n_{I,T_f^{op}\text{-}\mathbf{Coalg}^{op}(\gamma)} = [\![-]\!]^n_{J,\gamma} \circ \mathrm{Form}^n_f \tag{5.210}$$

Proof. We proceed by induction. The base case is trivial. We aim to show that if the hypothesis is valid for n then it is valid for $n+1$.

$$\begin{aligned}
& [\![-]\!]^{n+1}_{J,\gamma} \circ \mathrm{Form}^{n+1}_f \\
= & \quad \{ \text{ definitions 5.7.11 and 5.7.12 } \} \\
& \mathbf{Alg}^{\delta}(\gamma) \circ L_J[\![-]\!]^n_{J,\gamma} \circ L_J\,\mathrm{Form}^n_f \circ L(f, \mathrm{Form}^n_I) \\
= & \quad \{ \text{ induction hypothesis } \} \\
& \mathbf{Alg}^{\delta}(\gamma) \circ L_J[\![-]\!]^n_{I,T_f^{op}\text{-}\mathbf{Coalg}^{op}(\gamma)} \circ L(f, \mathrm{Form}^n_I) \\
= & \quad \{ \text{ bifunctoriality } \} \\
& \mathbf{Alg}^{\delta}(\gamma) \circ L(f, V\mathbf{Alg}^{\delta}(\gamma)) \circ L(I, [\![-]\!]^n_{I,T_f^{op}\text{-}\mathbf{Coalg}^{op}(\gamma)})
\end{aligned}$$

$$
\begin{aligned}
&= \{\ \text{definition 5.6.4}\ \}\\
&\quad L_f\text{-}\mathbf{Alg}(\mathbf{Alg}^{\delta}(\gamma)) \circ L(I, [\![-]\!]^{n}_{I,T_f^{op}\text{-}\mathbf{Coalg}^{op}(\gamma)})\\
&= \{\ \mathbf{Alg}^{\delta} \text{ is a split fibred functor }\ \}\\
&\quad \mathbf{Alg}^{\delta}(T_f^{op}\text{-}\mathbf{Coalg}^{op}(\gamma)) \circ L(I, [\![-]\!]^{n}_{I,T_f^{op}\text{-}\mathbf{Coalg}^{op}(\gamma)})\\
&= \{\ \text{definition 5.7.11}\ \}\\
&\quad [\![-]\!]^{n+1}_{I,T_f^{op}\text{-}\mathbf{Coalg}^{op}(\gamma)}
\end{aligned}
$$

□

5.7.2.4 Free Sequence Semantics

As a final case of the institution type conditions we have investigated, we consider the situation for the free sequence semantics supporting proposition variables, as described in section 5.7.1.4. Again, the model transformation includes an adjustment to the original valuation morphism.

Theorem 5.7.21. *Let* $\begin{pmatrix} X \\ \downarrow \gamma \\ T_J X \end{pmatrix}$ *be a* T_J*-coalgebra,* $f : I \to J$ *an* $\mathcal{I}$ *morphism and* $v : \mathrm{Prop}_J \to \Omega(X)$ *a valuation. Using the definitions in section 5.7.1.4, for each* $n \geq 0$*:*

$$[\![-]\!]^{n}_{I,T_f^{op}\text{-}\mathbf{Coalg}^{op}(X,\gamma),v\circ \mathrm{Prop}_f} = [\![-]\!]^{n}_{J,(X,\gamma),v} \circ \mathrm{Form}^{n}_{f}(\mathrm{Prop}) \tag{5.211}$$

Proof. We proceed by induction on n. The base case is trivial. For the inductive step, we consider the diagram in figure 5.7.2.4: The diagram commutes by applying naturality, bifunctoriality, definitions and the induction hypothesis in the triangle labelled I.H. The claim can then be read off the paths around the perimeter of the diagram. □

5.7.2.5 Discussion

We could interpret the institution conditions in the previous sections as showing that extending a logic with modalities capturing transformation between different coalgebraic signature functors is simple "syntactic sugar". This perspective oversimplifies the situation, as such modalities localize a coalgebraic model transformation within a formula. For example, if in some hypothetical logic we have a formula:

$$U\varphi \tag{5.212}$$

$$\begin{tikzcd}
 & \underline{L}_I^{n+1}0 \arrow[r, "\mathrm{Form}_f^{n+1}(\mathrm{Prop})"] \arrow[d, "\underline{L}(f, \underline{L}_I^n 0)"'] \arrow[ddl, bend right, "\underline{L}_I \llbracket - \rrbracket^n_{I, T_f^{op}\textbf{-Coalg}^{op}(X,\gamma), v}"] \arrow[dr, phantom, "\boxed{\text{I.H.}}"] & \underline{L}_J^{n+1}0 \arrow[r, "\llbracket - \rrbracket^{n+1}_{J,(X,\gamma),v}"] \arrow[d, "\underline{L}_J \llbracket - \rrbracket^n_{J,(X,\gamma),v}"] & \Omega(X) \\
 & \underline{L}_J \underline{L}_I^n 0 \arrow[r, "\underline{L}_J \llbracket - \rrbracket^n_{I, T_f^{op}\textbf{-Coalg}^{op}(X,\gamma), v}"] \arrow[ur, "\underline{L}_J \, \mathrm{Form}_f^n(\mathrm{Prop})"] & \underline{L}_J \, \Omega(X) \arrow[r, "\delta_{J,X} + 1"'] & \Omega \, TX + F \, \mathrm{Prop}_I \arrow[u, "{[\Omega(\gamma), \hat{v}]}"'] \\
\underline{L}_I \, \Omega(X) \arrow[rrr, "\delta_{I,X} + 1"'] \arrow[urr, "\underline{L}(f, \Omega \, X)"'] & & & \Omega \, T_I X + F \, \mathrm{Prop}_I \arrow[u, "\Omega \, T(f, X) + F \, \mathrm{Prop}_f"']
\end{tikzcd}$$

Figure 5.2: Commuting diagram for theorem 5.7.21

with informal interpretation “after unitary U is applied proposition φ holds”, we could find an equivalent formula ψ. In this example, ψ may have to be constructed by performing modifications throughout the structure of φ. These changes would be required to adjust modalities referring to measurements to compensate for the application of the unitary U. This transformation obscures our intent, and globalizes a previously local modification.

The fact we can “de-sugar” our formulae means that questions of adequacy, expressivity etc. can be reduced to equivalent problems for the logics on the fibres, for which there are many technical results in the literature.

5.8 Summary

In this chapter we studied modalities induced by functors between categories of coalgebras. Section 5.2 began by considering two different functors acting on coalgebraic models of quantum systems:

- An endofunctor corresponding to the application of a unitary transformation
- A functor between coalgebras with different signatures, corresponding to restriction to a subsystem

Both these functors are induced by natural transformations, in a manner dual to that described in example 2.1.1.

Before examining examples of this specific type in detail, we took a detour to look at more general classes of functors modelling other computational and logical examples. Section 5.3 investigated functors between categories of coalgebras, not necessarily induced by natural transformations, including:

- A simple Kleisli iteration example that changes the dynamics of the underlying coalgebra
- Functors corresponding to reflexive and transitive closure of relational models
- A functorial form of the model updates of [Carreiro et al., 2013]
- A functor corresponding to determinization of labelled transition systems

In order to cover all these examples, we introduced model transformations. These functor, natural transformation pairs were shown to induce modalities that allow us to move between different coalgebraic models, whilst respecting behavioural equivalence. The transitive closure example demonstrates that in general, these modalities cannot be replaced by equivalent formulae in the underlying coalgebraic logic.

Model transformations are very general, and so can only be shown to satisfy very basic properties. We therefore moved to a restricted setting in which stronger properties can be established. We considered a parameterized generalization of the logical connection style duality theoretic formulations of coalgebraic logic. This setting corresponds to the case where transformations of coalgebraic models are induced by natural transformations, and so encompasses our two quantum examples.

We developed an explicitly fibred (co)algebraic setting in section 5.6. In section 5.7 this machinery was then used to provide an abstract semantics for parameterized coalgebraic logics. We then introduced the idea of institution conditions, capturing the idea of a harmonious contravariant relationship between transformations of logics and transformations of models, based on similar ideas in [Goguen and Burstall, 1992]. The semantics of our logics were shown to satisfy these institution conditions, meaning every change of model can equivalently be replaced with a corresponding syntactic transformation, and vice versa. These institution conditions can then provide the foundations for reducing logical questions about parameterized logics to the usual unparameterized theory.

Chapter 6

Fibred Predicates

Motivated by applications in modelling quantum systems, in chapter 5 we considered generalizations of coalgebraic logics to the parameterized setting. A second major theme in the logical aspects of coalgebra are fibrations of predicates, sometimes referred to as invariants, over a category of coalgebras. Central to this approach are suitable liftings of coalgebraic signature functors. These liftings enable the construction of a fibration of coalgebras on predicates over a base category of "ordinary" coalgebras. In this chapter we study the impact of parameterization on the structure of these predicates. We investigate a suitable generalization of the lifting condition on signature functors to the parameterized setting. It is then shown that these liftings induce further fibred structure on the fibrations of invariants.

Our emphasis in this section is to explore the inherent mathematical structure, rather than a computational example such as the quantum examples used in previous sections. Given this more mathematical motivation we proceed at a fairly high level of abstraction and generality.

6.1 Predicate Liftings

From a logical perspective, a key fibration is the fibration of predicates on sets.

Definition 6.1.1 (Predicate Fibration)**.** The category **Pred** is defined as having:

- **Objects**: A set X and a specified subset $\left(\begin{array}{c} U \\ \downarrow \\ X \end{array}\right)$
- **Morphisms**: A morphism $\left(\begin{array}{c} U \\ \downarrow \\ X \end{array}\right) \to \left(\begin{array}{c} V \\ \downarrow \\ Y \end{array}\right)$ is a function $f : X \to Y$ such that:

$$f(U) \subseteq V \tag{6.1}$$

Composition and identities are given as in **Set**.

The forgetful functor:

$$\mathbf{Pred} \to \mathbf{Set} \tag{6.2}$$

$$\begin{pmatrix} U \\ \downarrow \\ X \end{pmatrix} \mapsto X \tag{6.3}$$

$$f \mapsto f \tag{6.4}$$

is a split fibration, referred to as the **predicate fibration**, with cartesian liftings:

$$\begin{tikzcd} \begin{pmatrix} f^{-1}V \\ \downarrow \\ X \end{pmatrix} \arrow[r, "f"] & \begin{pmatrix} V \\ \downarrow \\ Y \end{pmatrix} \\ X \arrow[r, "f"'] & Y \end{tikzcd} \tag{6.5}$$

The fibre over X is isomorphic to the powerset $\mathcal{P}X$. The substitution functor induced by a **Set** morphism $f : X \to Y$ is the inverse image map $f^{-1} : \mathcal{P}Y \to \mathcal{P}X$.

This fibration carries a lot of additional structure, and organizes "local" propositional logics on each set into a more powerful logical structure [Lawvere, 1969, 1970]. It can be useful to consider more general fibrations $\begin{pmatrix} \mathcal{E} \\ \downarrow \\ \mathcal{B} \end{pmatrix}$ as a fibration of abstract predicates over some base category $\mathcal{B}$, generalizing the predicate fibration. This point of view is emphasized in [Jacobs, 1999].

In the case of a **Set** coalgebra, a form of predicate that is of particular interest is that of an **invariant**. Invariants are subsets of the state space of a coalgebra that is closed under the dynamics. An elementary discussion of coalgebraic invariants can be found in [Jacobs, 2012b, Chapter 6]. As is described in [Jacobs, 2011a], invariants can be described as coalgebras for an endofunctor $\mathbf{Pred}(T)$, on the category **Pred** of predicates, such that:

$$(\mathbf{Pred}(T), T) : \begin{pmatrix} \mathbf{Pred} \\ \downarrow \\ \mathbf{Set} \end{pmatrix} \to \begin{pmatrix} \mathbf{Pred} \\ \downarrow \\ \mathbf{Set} \end{pmatrix} \tag{6.6}$$

is a 1-cell in $\mathfrak{Fib}$. These coalgebras can be organized into a fibration:

$$\begin{pmatrix} \textbf{Pred}(T)\text{-}\textbf{Coalg} \\ \downarrow \\ T\text{-}\textbf{Coalg} \end{pmatrix} \tag{6.7}$$

We are therefore interested in lifting endofunctors from a base category to the corresponding category of predicates, in a suitably compatible manner. This leads to the following definition:

Definition 6.1.2 (Hermida Jacobs Predicate Lifting)**.** For a fibration $\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix}$ and an endofunctor $T : \mathcal{B} \to \mathcal{B}$ a **predicate lifting** is a 1-cell in $\mathfrak{Fib}$, as in the following diagram:

$$\begin{tikzcd}
\mathcal{E} \arrow[r, "\overline{T}"] \arrow[d, "p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{B} \arrow[r, "T"'] & \mathcal{B}
\end{tikzcd} \tag{6.8}$$

This definition differs from the definition of predicate liftings in the sense of [Pattinson, 2003] given in definition 4.3.2, although the two are closely related. Throughout this section, any reference to a predicate lifting will refer to definition 6.1.2. Each predicate lifting induces a fibration:

$$\begin{pmatrix} \overline{T}\text{-}\textbf{Coalg} \\ \downarrow \\ T\text{-}\textbf{Coalg} \end{pmatrix} \tag{6.9}$$

We can think of such a structure as a generalization of the fibration of invariants in (6.7). Similar fibrations in algebraic and coalgebraic settings are considered in [Hermida and Jacobs, 1998, Ghani et al., 2010, 2012, Atkey et al., 2012].

6.2 Another Fibration of Coalgebras

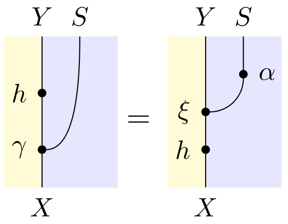
We will be interested in the appropriate generalization of the fibration in (6.7) when the signature functors vary over some parameter category.

Firstly we must organize coalgebras into a suitable "large" fibration. We actually require a different, and simpler, structure than that used in definition 5.6.5. This is because we no longer require opposite categories on the fibres as we did in the duality theoretic setting of chapter 5.

Definition 6.2.1. For category $\mathcal{C}$ we define the category $\mathbf{Coalg}(\mathcal{C})$ as follows:

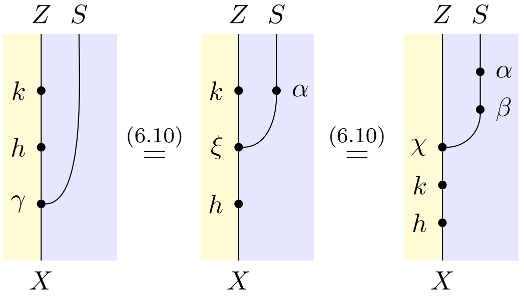
- **Objects**: Pairs consisting of:
  1. An endofunctor $T : \mathcal{C} \to \mathcal{C}$
  2. A T-coalgebra $\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix}$
- **Morphisms**: A morphism $\begin{pmatrix} S, & \begin{matrix} X \\ \downarrow \gamma \\ SX \end{matrix} \end{pmatrix} \to \begin{pmatrix} T, & \begin{matrix} Y \\ \downarrow \xi \\ TY \end{matrix} \end{pmatrix}$ is a pair consisting of:
  1. A natural transformation $\alpha : T \to S$
  2. A $\mathcal{C}$ morphism $h : X \to Y$

  such that:



(6.10)

Composition is defined componentwise as in $[\mathcal{C}, \mathcal{C}]^{op}$ in the first component and as in $\mathcal{C}$ in the second. That this is well defined can be seen from the following equalities:



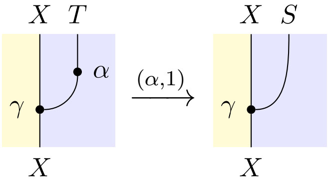
(6.11)

Lemma 6.2.2. *For a category $\mathcal{C}$ the forgetful functor:*

$$\begin{pmatrix} \mathbf{Coalg}(\mathcal{C}) \\ \downarrow \\ [\mathcal{C}, \mathcal{C}]^{op} \end{pmatrix} \tag{6.12}$$

is a split fibration. The fibre over T is isomorphic to T-$\mathbf{Coalg}$.

Proof. For endofunctors $S, T : \mathcal{C} \to \mathcal{C}$, natural transformation $\alpha : S \Rightarrow T$ and coalgebra $\begin{pmatrix} X \\ \downarrow \gamma \\ SX \end{pmatrix}$ a cartesian lifting is given by the morphism:



(6.13)

The existence and uniqueness of the cartesian morphism property can then be seen from the following diagram:

$$\begin{tikzcd}
\begin{pmatrix} X \\ \downarrow \xi \\ UX \end{pmatrix} \arrow[dr, "(\beta, h)"'] \arrow[drr, bend left, "(\beta \circ \alpha, h)"] & & \\
& \begin{pmatrix} X \\ \downarrow \alpha_Y \circ \gamma \\ TY \end{pmatrix} \arrow[r, "(\alpha, 1)"'] & \begin{pmatrix} X \\ \downarrow \gamma \\ SY \end{pmatrix} \\
U \arrow[dr, "\beta"'] \arrow[drr, bend left, "\beta \circ \alpha"] & & \\
& T \arrow[r, "\alpha"'] & S
\end{tikzcd} \tag{6.14}$$

That these liftings give a splitting then follows immediately from the definition of composition in $\mathbf{Coalg}(\mathcal{C})$. The fibre structure is then straightforward to check. □

We now introduce parameterization over a parameter category $\mathcal{I}$ by considering the fibration induced by a bifunctor of type $\mathcal{I}^{op} \times \mathcal{B} \to \mathcal{B}$.

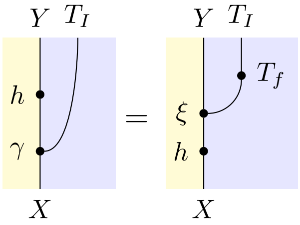
Definition 6.2.3. For categories $\mathcal{I}$ and $\mathcal{B}$ and bifunctor:

$$T : \mathcal{I}^{op} \times \mathcal{B} \to \mathcal{B} \tag{6.15}$$

we define split fibration $\begin{pmatrix} T_{(-)}\text{-}\mathbf{Coalg} \\ \downarrow \\ \mathcal{I} \end{pmatrix}$ by change of base. Explicitly the category $T_{(-)}$-**Coalg** is given by:

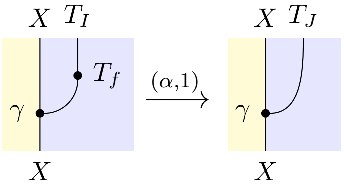
- **Objects**: Pairs consisting of:
    1. An $\mathcal{I}$ object I
    2. A T_I-coalgebra $\begin{pmatrix} X \\ \downarrow\gamma \\ T_I X \end{pmatrix}$
- **Morphisms**: A morphism $\left(I, \begin{matrix} X \\ \downarrow\gamma \\ T_I X \end{matrix}\right) \to \left(J, \begin{matrix} Y \\ \downarrow\xi \\ T_J Y \end{matrix}\right)$ is a pair consisting of:
    1. An $\mathcal{I}$ morphism $f : I \to J$
    2. An $\mathcal{B}$ morphism $h : X \to Y$

  such that:



(6.16)

The fibre over I is then T_I-**Coalg**. The splitting is given explicitly for $\mathcal{I}$ morphism $f : I \to J$ and $T_{(-)}$-**Coalg** object $\left(J, \begin{matrix} X \\ \downarrow\gamma \\ T_J X \end{matrix}\right)$ as lifting:



(6.17)

6.3 Predicate Lifting in the Parameterized Setting

We now introduce a notion of predicate lifting varying over a parameter category, generalizing that of definition 6.1.2.

Definition 6.3.1. For category $\mathcal{I}$, a bifunctor:

$$T : \mathcal{I}^{op} \times \mathcal{B} \to \mathcal{B} \tag{6.18}$$

and a fibration:

$$\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix} \tag{6.19}$$

a **parameterized predicate lifting** of T is a bifunctor $\overline{T}$ such that the following diagram commutes:

$$\begin{tikzcd}
\mathcal{I}^{op} \times \mathcal{E} \arrow[r, "\overline{T}"] \arrow[d, "1 \times p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{I}^{op} \times \mathcal{B} \arrow[r, "T"'] & \mathcal{B}
\end{tikzcd} \tag{6.20}$$

and for every $\mathcal{I}$ object I, the following is a 1-cell in $\mathfrak{Fib}$:

$$\begin{tikzcd}
\mathcal{E} \arrow[r, "\overline{T}_I"] \arrow[d, "p"'] & \mathcal{E} \arrow[d, "p"'] \\
\mathcal{B} \arrow[r, "T_I"'] & \mathcal{B}
\end{tikzcd} \tag{6.21}$$

That is, for each object I in $\mathcal{I}$, $\overline{T}_I$ is a predicate lifting of T_I in the sense of definition 6.1.2.

Lemma 6.3.2. *Let* $\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix}$ *be a fibration and* $T : \mathcal{B} \to \mathcal{B}$ *an endofunctor with predicate lifting:*

$$\begin{tikzcd}
\mathcal{E} \arrow[r, "\overline{T}"] \arrow[d, "p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{B} \arrow[r, "T"'] & \mathcal{B}
\end{tikzcd} \tag{6.22}$$

Given a $\overline{T}$*-coalgebra* $\begin{pmatrix} U \\ \downarrow \tilde{\xi} \\ \overline{T}U \end{pmatrix}$ *over* T*-coalgebra* $\begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix}$ *and a* T*-coalgebra homomorphism:*

$$\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix} \tag{6.23}$$

for each cartesian lifting:

$$\begin{tikzcd}
h^*U \arrow[r, "\overline{h}"] & U \\
X \arrow[r, "h"'] & Y
\end{tikzcd} \tag{6.24}$$

there exists a unique $\overline{T}$*-coalgebra* $\begin{pmatrix} h^*U \\ \downarrow h^*\tilde{\xi} \\ \overline{T}h^*U \end{pmatrix}$ *over* $\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix}$ *such that* $\overline{h}$ *is a* $\overline{T}$*-coalgebra homomorphism.*

Proof. As $\overline{T}$ is a morphism of fibrations it preserves cartesian lifts and so $\overline{T}(\overline{h})$ is a cartesian morphism. We can then define our $\overline{T}$-coalgebra as the unique fill in morphism in the following situation:

$$\begin{tikzcd}
h^*U \arrow[r, "\overline{h}"] \arrow[dr, dashed, "h^*\tilde{\xi}"'] & U \arrow[dr, "\tilde{\xi}"] & \\
& \overline{T}h^*U \arrow[r, "\overline{T}(\overline{h})"'] & \overline{T}U \\
X \arrow[r, "h"] \arrow[dr, "\gamma"'] & Y \arrow[dr, "\xi"] & \\
& TX \arrow[r, "Th"'] & TY
\end{tikzcd} \tag{6.25}$$

□

Lemma 6.3.3. *Let* $\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix}$ *be a fibration and* $T : \mathcal{I}^{op} \times \mathcal{B} \to \mathcal{B}$ *a bifunctor with parameterized predicate lifting:*

$$\begin{tikzcd}
\mathcal{I}^{op} \times \mathcal{E} \arrow[r, "\overline{T}"] \arrow[d, "1 \times p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{I}^{op} \times \mathcal{B} \arrow[r, "T"'] & \mathcal{B}
\end{tikzcd} \tag{6.26}$$

Let I, J, K *be* $\mathcal{I}$ *objects and* $u : I \to J$ *and* $t : K \to I$ $\mathcal{I}$ *morphisms. Given a* $\overline{T}_J$*-coalgebra* $\begin{pmatrix} U \\ \downarrow \tilde{\xi} \\ \overline{T}_J U \end{pmatrix}$ *over* T_J*-coalgebra* $\begin{pmatrix} Y \\ \downarrow \xi \\ T_J Y \end{pmatrix}$ *and a* T_I*-coalgebra homomorphism:*

$$\begin{pmatrix} X \\ \downarrow \gamma \\ T_I X \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow T(u, U) \circ \xi \\ T_I Y \end{pmatrix} \tag{6.27}$$

using the definitions in lemma 6.3.2 we have equality:

$$\overline{T}(t, h^*U) \circ h^*(\overline{T}(u, U) \circ \tilde{\xi}) = h^*(\overline{T}(u \circ t, U) \circ \tilde{\xi}) \tag{6.28}$$

Proof. We note that the following diagram commutes, using bifunctoriality and the definitions of lemma 6.3.2:

$$\begin{tikzcd}
h^*U \arrow[r, "\overline{h}"] \arrow[dd, "h^*(\overline{T}(u,U)\circ\tilde{\xi})"'] & U \arrow[d, "\xi"] \\
 & \overline{T}_K U \arrow[d, "\overline{T}(u,U)"] \\
\overline{T}_J h^*U \arrow[r, "\overline{T}_J\overline{h}"] \arrow[d, "\overline{T}(t,h^*U)"'] & \overline{T}_J U \arrow[d, "\overline{T}(t,U)"] \\
\overline{T}_I h^*U \arrow[r, "\overline{T}_I\overline{h}"'] & \overline{T}_I U
\end{tikzcd} \tag{6.29}$$

The claim then follows from the uniqueness property in lemma 6.3.2. $\square$

Proposition 6.3.4. *Let* $\begin{pmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{pmatrix}$ *be a fibration and* $T : \mathcal{B} \to \mathcal{B}$ *an endofunctor with predicate lifting:*

$$\begin{tikzcd}
\mathcal{E} \arrow[r, "\overline{T}"] \arrow[d, "p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{B} \arrow[r, "T"'] & \mathcal{B}
\end{tikzcd} \tag{6.30}$$

Given:

- *A* $\overline{T}$*-coalgebra* $\begin{pmatrix} U \\ \downarrow \tilde{\xi} \\ \overline{T}U \end{pmatrix}$ *over* T*-coalgebra* $\begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix}$
- *A* $\overline{T}$*-coalgebra* $\begin{pmatrix} W \\ \downarrow \tilde{\chi} \\ \overline{T}W \end{pmatrix}$ *over* T*-coalgebra* $\begin{pmatrix} Z \\ \downarrow \chi \\ TZ \end{pmatrix}$
- *A* T*-coalgebra* $\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix}$ *and morphism:*

$$\begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \xrightarrow{h} \begin{pmatrix} Y \\ \downarrow \xi \\ TY \end{pmatrix} \tag{6.31}$$

- *A T-coalgebra morphism:*

$$\begin{pmatrix} Z \\ \downarrow \chi \\ TZ \end{pmatrix} \xrightarrow{g} \begin{pmatrix} X \\ \downarrow \gamma \\ TX \end{pmatrix} \tag{6.32}$$

- *A $\overline{T}$-coalgebra morphism:*

$$\begin{pmatrix} W \\ \downarrow \tilde{\chi} \\ \overline{T} \end{pmatrix} \xrightarrow{f} \begin{pmatrix} U \\ \downarrow \tilde{\xi} \\ \overline{T}U \end{pmatrix} \tag{6.33}$$

  over the composite $h \circ g$

There is then a unique $\overline{T}$-coalgebra morphism:

$$\begin{pmatrix} W \\ \downarrow \tilde{\chi} \\ \overline{T}W \end{pmatrix} \xrightarrow{\tilde{g}} \begin{pmatrix} h^*U \\ \downarrow h^*\tilde{\xi} \\ \overline{T}h^*U \end{pmatrix} \tag{6.34}$$

over g such that $f = \overline{h} \circ \tilde{g}$, where $h^*\tilde{\xi}$ and $\overline{h}$ are as defined in lemma 6.3.2.

Proof. As $\overline{h}$ is cartesian, there is a unique $\tilde{g}$ over g such that $f = \overline{h} \circ \tilde{g}$. It then remains to show that $\tilde{g}$ is a $\overline{T}$-coalgebra morphism. The following diagram commutes in $\mathcal{E}$:

$$\begin{tikzcd}
W \arrow[d, "\tilde{\chi}"'] \arrow[rrd, bend left, "f"] & & \\
\overline{T}W \arrow[rd, "\overline{T}\tilde{g}"'] & & U \arrow[d, "\tilde{\xi}"] \\
& \overline{T}h^*U \arrow[r, "\overline{T}(\overline{h})"'] & \overline{T}U
\end{tikzcd} \tag{6.35}$$

The following diagram also commutes:

$$\begin{tikzcd}
W \arrow[dr, "\tilde{g}"'] \arrow[drr, bend left, "\tilde{\xi}"] & & \\
& h^*U \arrow[d, "h^*\tilde{\xi}"'] & U \arrow[d, "\tilde{\xi}"] \\
& \overline{T}h^*U \arrow[r, "\overline{T}(\overline{h})"'] & \overline{T}U
\end{tikzcd} \tag{6.36}$$

As $\overline{T}$ is a morphism of fibrations, $\overline{T}(\overline{h})$ is cartesian. As g is a T-coalgebra homomorphism, both the composites $h^*\tilde{\xi} \circ \tilde{g}$ and $\overline{T}\tilde{g} \circ \tilde{\chi}$ lie above the same morphism in $\mathcal{B}$. It then follows by the universal property of cartesian lifts that $\tilde{g}$ is a $\overline{T}$-coalgebra homomorphism. □

Proposition 6.3.5. *Let* $\begin{pmatrix}\mathcal{E}\\ \downarrow p\\ \mathcal{B}\end{pmatrix}$ *be a fibration and* $T : \mathcal{I}^{op} \times \mathcal{B} \to \mathcal{B}$ *a bifunctor with parameterized predicate lifting:*

$$\begin{tikzcd}
\mathcal{I}^{op} \times \mathcal{E} \arrow[r, "\overline{T}"] \arrow[d, "1 \times p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{I}^{op} \times \mathcal{B} \arrow[r, "T"'] & \mathcal{B}
\end{tikzcd} \tag{6.37}$$

There is then a fibration:

$$\begin{pmatrix}\overline{T}_{(-)}\text{-}\mathbf{Coalg}\\ \downarrow\\ T_{(-)}\text{-}\mathbf{Coalg}\end{pmatrix} \tag{6.38}$$

This fibration is split if $\begin{pmatrix}\mathcal{E}\\ \downarrow\\ \mathcal{B}\end{pmatrix}$ *is split.*

Proof. We define the obvious functor induced by p, noting $p \circ \overline{T}_I = T_I \circ p$:

$$\overline{T}_{(-)}\text{-}\mathbf{Coalg} \to T_{(-)}\text{-}\mathbf{Coalg} \tag{6.39}$$

$$\left(I, \begin{matrix} U \\ \downarrow \gamma \\ \overline{T}_I U\end{matrix}\right) \mapsto \left(I, \begin{matrix} pU \\ \downarrow p\gamma \\ T_I pU\end{matrix}\right) \tag{6.40}$$

$$(u, h) \mapsto (u, ph) \tag{6.41}$$

We consider liftings of the form:

$$\begin{tikzcd}
\begin{pmatrix} h^*U \\ \downarrow h^*(\overline{T}(u,U)\circ\tilde{\xi}) \\ \overline{T_I}h^*U \end{pmatrix} \arrow[r, "{(u,\overline{h})}"] & \begin{pmatrix} U \\ \downarrow\tilde{\xi} \\ \overline{T}_JU \end{pmatrix} \\
\begin{pmatrix} X \\ \downarrow\gamma \\ T_IX \end{pmatrix} \arrow[r, "{(u,h)}"'] & \begin{pmatrix} Y \\ \downarrow\xi \\ T_JY \end{pmatrix}
\end{tikzcd}$$

(6.42)

That the upper line is a valid morphism follows from lemma 6.3.2. We must show that such liftings have the appropriate universal property. We assume the following, trivially commuting, diagram in $T_{(-)}$-**Coalg**:

$$\begin{tikzcd}
\begin{pmatrix} Z \\ \downarrow\chi \\ T_KZ \end{pmatrix} \arrow[dr, "{(t,g)}"'] \arrow[drr, bend left, "{(u\circ t,h\circ g)}"] & & \\
& \begin{pmatrix} X \\ \downarrow\gamma \\ T_IX \end{pmatrix} \arrow[r, "{(u,h)}"'] & \begin{pmatrix} Y \\ \downarrow\xi \\ T_JY \end{pmatrix}
\end{tikzcd}$$

(6.43)

That this commutes is equivalent to the following diagram commuting, also in $T_{(-)}$-**Coalg**:

$$\begin{tikzcd}
\begin{pmatrix} Z \\ \downarrow\chi \\ T_KZ \end{pmatrix} \arrow[d, "{(1,g)}"] \arrow[dr, bend left, "{(1,h\circ g)}"] & \\
\begin{pmatrix} X \\ \downarrow T(t,X)\circ\gamma \\ T_KX \end{pmatrix} \arrow[r, "{(1,h)}"'] & \begin{pmatrix} Y \\ \downarrow T(u\circ t)\circ\xi \\ T_KY \end{pmatrix}
\end{tikzcd}$$

(6.44)

We now assume the following situation above the previous diagram:

$$\begin{tikzcd}
\begin{pmatrix} W \\ \downarrow\tilde{\chi} \\ \overline{T}_KW \end{pmatrix} \arrow[dr, bend left, "{(1,f)}"] & \\
\begin{pmatrix} h^*U \\ \downarrow\overline{T}(t,h^*U)\circ h^*(\overline{T}(u,U)\circ\tilde{\xi}) \\ \overline{T}_Kh^*U \end{pmatrix} \arrow[r, "{(1,\overline{h})}"'] & \begin{pmatrix} U \\ \downarrow\overline{T}(u\circ t)\circ\tilde{\xi} \\ \overline{T_K}U \end{pmatrix}
\end{tikzcd}$$

(6.45)

Applying lemma 6.3.3 we have:

$$\overline{T}(t, h^*U) \circ h^*(\overline{T}(u, U) \circ \tilde{\xi}) = h^*(\overline{T}(u \circ t, U) \circ \tilde{\xi}) \tag{6.46}$$

We therefore can apply proposition 6.3.4 to conclude there exists a unique fill in of the form $(1, \tilde{g})$, and this corresponds to $(t, \tilde{g})$ being the unique fill in of the diagram:

$$\begin{tikzcd}
\begin{pmatrix} W \\ \downarrow \tilde{\chi} \\ \overline{T}_K W \end{pmatrix} \arrow[dr, dashed, "{(t, \tilde{g})}"] \arrow[drr, bend left, "{(u \circ t, f)}"] & & \\
& \begin{pmatrix} X \\ \downarrow h^*(\overline{T}(u, U) \circ \tilde{\xi}) \\ \overline{T}_I X \end{pmatrix} \arrow[r, "{(u, \overline{h})}"'] & \begin{pmatrix} Y \\ \downarrow \tilde{\xi} \\ \overline{T}_J Y \end{pmatrix}
\end{tikzcd} \tag{6.47}$$

If $\begin{pmatrix} \mathcal{E} \\ \downarrow \\ \mathcal{B} \end{pmatrix}$ is split then by using that splitting, the liftings we have constructed will also then be a splitting, as the domain coalgebra is uniquely defined. □

Corollary 6.3.6. *For fibration* $\begin{pmatrix} \mathcal{E} \\ \downarrow \\ \mathcal{B} \end{pmatrix}$ *and endofunctor* $T : \mathcal{B} \to \mathcal{B}$ *with predicate lifting* $\overline{T} : \mathcal{E} \to \mathcal{E}$ *there is a fibration:*

$$\begin{tikzcd}
\overline{T}\text{-}\mathbf{Coalg} \arrow[d] \\
T\text{-}\mathbf{Coalg}
\end{tikzcd} \tag{6.48}$$

This fibration is split if $\begin{pmatrix} \mathcal{E} \\ \downarrow \\ \mathcal{B} \end{pmatrix}$ *is split.*

Proof. Take $\mathcal{I}$ to be the terminal category and consider T and $\overline{T}$ as bifunctors with a trivial first parameter. □

Proposition 6.3.7. *Let* $\begin{pmatrix} \mathcal{E} \\ \downarrow \\ \mathcal{B} \end{pmatrix}$ *be a fibration and* $S, T : \mathcal{B} \to \mathcal{B}$ *endofunctors with* *respective predicate liftings* $\overline{S}, \overline{T} : \mathcal{E} \to \mathcal{E}$*. Every 2-cell in* $\mathfrak{Fib}$ *as in the diagram:*

$$\begin{tikzcd}
\mathcal{E} \arrow[r, bend left, "\overline{S}"{name=U}] \arrow[r, bend right, "\overline{T}"'{name=D}] \arrow[d] & \mathcal{E} \arrow[d] \\
\mathcal{B} \arrow[r, bend left, "S"{name=U2}] \arrow[r, bend right, "T'"'{name=D2}] & \mathcal{B}
\arrow[Rightarrow, from=U, to=D, "\overline{\alpha}"]
\arrow[Rightarrow, from=U2, to=D2, "\alpha"]
\end{tikzcd} \tag{6.49}$$

induces a $\mathfrak{Fib}$ *morphism:*

$$\begin{tikzcd}
\overline{S}\text{-}\mathbf{Coalg} \arrow[r, "\overline{\alpha}^*"] \arrow[d] & \overline{T}\text{-}\mathbf{Coalg} \arrow[d] \\
S\text{-}\mathbf{Coalg} \arrow[r, "\alpha^*"'] & T\text{-}\mathbf{Coalg}
\end{tikzcd} \tag{6.50}$$

Proof. $\overline{\alpha}^*$ and α^* are the usual functors between categories of coalgebras induced by natural transformations between the corresponding signature functors. That the required diagram commutes then follows from the natural transformation $\overline{\alpha}$ being above α. Lemma 6.3.2 then implies cartesian morphisms are preserved.

In the case that $\left(\begin{smallmatrix} \mathcal{E} \\ \downarrow \\ \mathcal{B} \end{smallmatrix}\right)$ has a splitting, as $\overline{\alpha}^*$ and α^* both act as the identity on morphisms it is easy to see that the induced splitting of $\left(\begin{smallmatrix} \overline{S}\text{-}\mathbf{Coalg} \\ \downarrow \\ S\text{-}\mathbf{Coalg} \end{smallmatrix}\right)$ is mapped onto the induced splitting of $\left(\begin{smallmatrix} \overline{T}\text{-}\mathbf{Coalg} \\ \downarrow \\ T\text{-}\mathbf{Coalg} \end{smallmatrix}\right)$. $\square$

We now have the available technical results and definitions to describe a parameterized analogue of the fibrations of invariants discussed in section 6.1.

Theorem 6.3.8. *Let* $\left(\begin{smallmatrix} \mathcal{E} \\ \downarrow p \\ \mathcal{B} \end{smallmatrix}\right)$ *be a fibration,* $T : \mathcal{I}^{op} \times \mathcal{B} \to \mathcal{B}$ *a bifunctor with parameterized predicate lifting:*

$$\begin{tikzcd}
\mathcal{I}^{op} \times \mathcal{E} \arrow[r, "\overline{T}"] \arrow[d, "1 \times p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{I}^{op} \times \mathcal{B} \arrow[r, "T"'] & \mathcal{B}
\end{tikzcd} \tag{6.51}$$

Then:

1. *The forgetful functor* $\begin{pmatrix} \overline{T}_{(-)}\text{-}\mathbf{Coalg} \\ \downarrow \\ \mathcal{I} \end{pmatrix}$ *is a fibration in* $\mathfrak{Fib}(\mathcal{I})$*.*

2. *For each* $\mathcal{I}$ *object* I *the fibration over* I *is* $\begin{pmatrix} \overline{T_I}\text{-}\mathbf{Coalg} \\ \downarrow \\ T_I\text{-}\mathbf{Coalg} \end{pmatrix}$ *as described in corollary 6.3.6*

3. *For each* $\mathcal{I}$ *morphism* $u : I \to J$*, the substitution* $\mathfrak{Fib}$ *morphism:*

$$\begin{pmatrix} \overline{T}_J\text{-}\mathbf{Coalg} \\ \downarrow \\ T_J\text{-}\mathbf{Coalg} \end{pmatrix} \to \begin{pmatrix} \overline{T}_I\text{-}\mathbf{Coalg} \\ \downarrow \\ T_I\text{-}\mathbf{Coalg} \end{pmatrix} \tag{6.52}$$

   is the morphism induced by the natural transformations $\overline{T}(u, -)$ *and* $T(u, -)$ *as described in proposition 6.3.7.*

Proof. The forgetful functors factor as in the following diagram:

$$\begin{tikzcd}
\overline{T}_{(-)}\text{-}\mathbf{Coalg} \arrow[rr] \arrow[dr] & & T_{(-)}\text{-}\mathbf{Coalg} \arrow[dl] \\
& \mathcal{I} &
\end{tikzcd} \tag{6.53}$$

For part 1, the two vertical functors are fibrations over $\mathcal{I}$ as described in definition 6.2.3. The horizontal functor is also a fibration by proposition 6.3.5. It is easy to check that the horizontal functor preserves cartesian liftings and is therefore a morphism in $\mathbf{Fib}(\mathcal{I})$. The claim then follows from the usual theory of fibrations, see for example [Jacobs, 1999, Proposition 9.4.3].

Part 2 is a straightforward calculation. Similarly part 3 is straightforward, noting that $\overline{T}(u, -)$ is a natural transformation over $T(u, -)$ by the assumption of being a parameterized lifting. □

Chapter 7

Conclusion

In this thesis we have used category theoretic methods to investigate several topics at the boundary between quantum computation and logic in theoretical computer science. Our two main themes of string diagrams and logical aspects of parameterized coalgebras both have scope for further developments. We now make some concluding remarks about both these areas.

7.1 String Diagrams

The work of chapter 2 illustrates how string diagrams can be used to perform calculational categorical proofs exploiting both our topological intuition and all the available type information. We could not hope to provide comprehensive coverage given the available space, for example we have not discussed dual adjunctions, equivalences, exponentials or many common types of limits and colimits. Hopefully though, we have provided sufficient background to demonstrate the practicality of the approach.

Tool support for developing and documenting proofs in this diagrammatic style would be tremendously useful, and is a direction for further investigation. The proofs we produce, although conceptually compact, often physically take up a large amount of page space and the proof style also makes essential use of colour. These attributes are not particularly "publication friendly", tool support could also aim to provide (preferably bidirectional) translations between string based proofs and more traditional formats.

Unfortunately string diagrams are rarely presented in introductory courses on category theory. This is surprising as they are straightforward to understand and simplify many aspects of calculations. A version of the material in chapter 2 was presented to students in the final lecture of the 2013 Categories, Proofs and Processes course at Oxford. Further experience introducing these tools earlier in category theory courses would be of great interest.

Many fields of mathematics and computer science “live” within a suitable bicategory. It would be useful to continue a program of identifying the necessary bicategorical structure for performing practical graphical calculations in these fields. Some example applications include:

- Quantum computation, the source of inspiration for our current work
- Linear algebra - This is suggested by the authors of [Bonchi et al., 2014a,b] using the machinery in those papers
- Concurrency theory - This is really the main thrust of the work in [Bonchi et al., 2014a,b] and many related papers
- Computational linguistics shares common mathematical structure with the string diagrammatic perspective on quantum computation, see [Coecke et al., 2010].
- Relational algebra has structure suitable for graphical calculi, this can be done at a very high level of generality, see [Carboni and Walters, 1987]. Allegories [Freyd and Scedrov, 1990] provide a second categorical formulation of categories of relations. The book [Bird and de Moor, 1997] uses allegories in order to calculate computationally efficient implementations for functional programs from potentially non-deterministic specifications. It should be possible to present a graphical version of this approach. Some related work in a graphical direction appears in [Brown and Hutton, 1994, Brown and Jeffrey, 1994].

Each of these areas have different technical requirements, for example, most quantum computation examples work with monoidal categories and are “in black and white”. By contrast, our categorical work in chapter 2 involves multiple 0-cells and is therefore “in colour”. Linguistic applications may drop assumptions about the existence of symmetries or braidings to better model their problem domain, and relational algebra would ideally include support for inequations rather than just equations. These examples suggest that development of tools and general mathematical theory would benefit from the serious investigation of a broad spectrum of these bicategorical calculi.

7.2 Fibrations of Coalgebras and Corresponding Logics

In chapter 3, following earlier work in [Abramsky, 2009, 2010], we developed an elementary formulation of a representation result for the unitary transformations by generalizing the notion of coalgebra morphism. These results suggest that generalized categories of coalgebras may be a fruitful environment in which to "do coalgebra" when additional modelling freedom is required. Earlier work investigating parameterized coalgebra [Kurz and Pattinson, 2000a,b, Pattinson, 2002] also supports this perspective. The coalgebraic investigation of binary methods in [Tews, 2000] provides an example with a different flavour, also exploiting signature functors with an additional parameter.

This line of thought can been seen in the early parts of chapter 3, and leads to the adoption of fibrations, rather than indexed categories, in chapters 5 and 6. The total categories of these fibrations are exactly the generalized categories of coalgebras of independent interest. We have phrased some of our results at a fairly high level of generality, for example, the fibred functor and adjunction liftings of theorem 5.6.11, the institution conditions of section 5.7.2, and the internal fibration in $\mathfrak{Fib}(\mathcal{I})$ of theorem 6.3.8. We anticipate that if the total categories of fibrations are taken seriously as a setting to do coalgebra, then hopefully these results will be of further interest.

Our definitions and proofs in chapters 3, 4, 5 and 6 continued to apply the string diagrammatic techniques of chapter 2 to research problems in coalgebra and coalgebraic logic. We believe that this style of reasoning is particularly effective for typical applications in these fields. Adjunctions, monads, distributive laws and other special natural transformations are commonplace in coalgebraic calculations. String diagrams are therefore possibly under-appreciated in that community.

Bibliography

S. Abramsky. Big toy models: Representing physical systems as Chu spaces. *CoRR*, abs/0910.2393, 2009.

S. Abramsky. Coalgebras, Chu spaces, and representations of physical systems. In *LICS*, pages 411–420. IEEE Computer Society, 2010.

S. Abramsky. A cook's tour of the finitary non-well-founded sets. *CoRR*, abs/1111.7148, 2011.

S. Abramsky. Relational databases and bells theorem. In *In Search of Elegance in the Theory and Practice of Computation*, pages 13–35. Springer, 2013.

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*. Elsevier, 2008.

S. Abramsky and L. Hardy. Logical bell inequalities. *Physical Review A*, 85(6):062114, 2012.

J. Adámek. Free algebras and automata realizations in the language of categories. *Commentationes Mathematicae Universitatis Carolinae*, 15(4):589–602, 1974.

J. Adámek and V. Koubek. Least fixed point of a functor. *Journal of Computer and System Sciences*, 19(2):163–178, 1979.

J. Adámek and H.E. Porst. On tree coalgebras and coalgebra presentations. *Theor. Comput. Sci.*, 311(1-3):257–283, 2004.

J. Adámek and J. Rosický. *Locally Presentable and Accessible Categories*. Number 189 in London Mathematical Society Lecture Notes. Cambridge University Press, 1994.

J. Adámek and V. Trnková. *Automata and algebras in categories*, volume 37. Springer, 1990.

J. Adámek, S. Milius, and L.S. Moss. Initial algebras and terminal coalgebras: a survey. *TU Braunschweig.*

R. Atkey, N. Ghani, B. Jacobs, and P. Johann. Fibrational induction meets effects. In *Foundations of Software Science and Computational Structures*, pages 42–57. Springer, 2012.

R.C. Backhouse. Make formality work for us. *EATCS Bulletin*, 38:219–249, 1989.

J.C. Baez and M. Stay. Physics, topology, logic and computation: A Rosetta stone. In B. Coecke, editor, *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, pages 95–174. Springer, 2011.

A. Baltag and S. Smets. LQP: the dynamic logic of quantum information. *Mathematical Structures in Computer Science*, 16(3):491–525, 2006.

M. Barr and C. Wells. Toposes, triples and theories. *Reprints in Theory and Applications of Categories*, 1:1–289, 2005.

F. Bartels. *On Generalized Coinduction and Probabilistic Specification Formats.* PhD thesis, Vrije Universiteit Amsterdam, 2004.

J. Beck. Distributive laws. In H. Appelgate, M. Barr, J. Beck, F.W. Lawvere, F. Linton, E. Manes, M. Tierney, and F. Ulmer, editors, *Seminar on Triples and Categorical Homotopy Theory*, volume 80 of *Lecture Notes in Mathematics*, pages 119–140. Springer, 1969.

J. Bénabou. Introduction to bicategories. In A. Dold and B. Eckmann, editors, *Reports of the Midwest Category Seminar*, volume 47 of *Lecture Notes in Mathematics*, pages 1–77. Springer, 1967.

R.S. Bird and O. de Moor. *Algebra of programming.* Prentice Hall International series in computer science. Prentice Hall, 1997.

G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Pure Mathematics*, 37(4):823–843, 1936.

R. Blute, J.R.B. Cockett, and R.A.G. Seely. The logic of linear functors. *Mathematical Structures in Computer Science*, 12(4):513–539, 2002.

F. Bonchi, P. Sobociński, and F. Zanasi. Interacting Hopf algebras. Technical report, arXiv:1403.7048, 2014a.

F. Bonchi, P. Sobociński, and F. Zanasi. A categorical semantics of signal flow graphs. In *CONCUR'14*, 2014b. To appear.

M.M. Bonsangue and A. Kurz. Duality for logics of transition systems. In V. Sassone, editor, *FoSSaCS*, volume 3441 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 2005.

F. Borceux. *Handbook of Categorical Algebra*, volume 2. Cambridge University Press, 1994.

C. Brown and G. Hutton. Categories, allegories and circuit design. In *LICS*, pages 372–381. IEEE Computer Society, 1994.

C. Brown and A. Jeffrey. Allegories of circuits. In A. Nerode and Y.V. Matiyasevich, editors, *LFCS*, volume 813 of *Lecture Notes in Computer Science*, pages 56–68. Springer, 1994.

A. Carboni and R.F.C. Walters. Cartesian bicategories I. *Journal of Pure and Applied Algebra*, 49(12):11 – 32, 1987.

F. Carreiro, D. Gorín, and L. Schröder. Coalgebraic announcement logics. In *Automata, Languages, and Programming*, pages 101–112. Springer, 2013.

J.R.B. Cockett and R.A.G. Seely. Linearly distributive functors. *The Barrfestschrift, Journal of Pure and Applied Algebra*, 143:133–173, 1999.

B. Coecke and R. Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.

B. Coecke, M. Sadrzadeh, and S. Clark. Mathematical foundations for distributed compositional model of meaning. Lambek festschrift. *Linguistic Analysis*, 36:345–384, 2010.

P.-L. Curien. The joy of string diagrams. In *Computer Science Logic*, pages 15–22. Springer, 2008.

E.W. Dijkstra and C.S. Scholten. *Predicate calculus and program semantics.* Texts and monographs in computer science. Springer, 1990.

A. Döring. Topos quantum logic and mixed states. In B. Coecke, P. Panangaden, and P. Selinger, editors, *Proceedings of the 6th International Conference on Quantum Physics and Logic (QPL 2009)*, volume 270, 2011.

A. Döring. Topos based logic for quantum systems and bi-Heyting algebras. arXiv:1202.2750v1, 2012.

E.J. Dubuc and M. Szyld. A Tannakian context for Galois theory. *Advances in Mathematics*, 234:528–549, 2013.

C. Ehresmann. Catégories structurées. *Ann. Sci. Ecole Norm. Sup.*, 80:349–425, 1963.

S. Eilenberg and J.C. Moore. Adjoint functors and triples. *Illinois J. Math.*, 9: 381–398, 1965.

M.M. Fokkinga. *A Gentle Introduction to Category Theory — the calculational approach*, volume Part I, pages 1–72. University of Utrecht, Utrecht, Netherlands, Sep 1992a.

M.M. Fokkinga. Calculate categorically! *Formal Aspects of Computing*, 4(4):673–692, 1992b.

M.M. Fokkinga and L. Meertens. Adjunctions. Technical Report Memoranda Inf 94-31, University of Twente, Enschede, Netherlands, Jun 1994.

P.J. Freyd and A. Scedrov. *Categories, Allegories.* North-Holland, 1990.

N. Ghani, P. Johann, and C. Fumex. Fibrational induction rules for initial algebras. In *Computer Science Logic*, pages 336–350. Springer, 2010.

N. Ghani, P. Johann, and C. Fumex. Generic fibrational induction. *arXiv preprint arXiv:1206.0357*, 2012.

J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *J. ACM*, 39(1):95–146, 1992.

J.A. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright. An introduction to categories, algebraic theories and algebras. Technical report, IBM Thomas J. Watson Research Centre, Yorktown Heights, 1975.

J.A. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright. Initial algebra semantics and continuous algebras. *Journal of the ACM*, 24(1):68–95, 1977.

J.W. Gray. *Formal Category Theory: Adjointness for 2-Categories*, volume 391 of *Lecture Notes in Mathematics*. Springer, 1974.

D. Gries and F.B. Schneider. *A Logical Approach to Discrete Math*. Springer, 1993.

T. Hagino. *A Categorical Programming Language*. PhD thesis, University of Edinburgh, 1987a.

T. Hagino. A typed lambda calculus with categorical type constructors. In D.H. Pitt, A. Poigné, and D.E. Rydeheard, editors, *Category Theory and Computer Science*, volume 283 of *Lecture Notes in Computer Science*, pages 140–157. Springer, 1987b.

C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. Comput.*, 145(2):107–152, 1998.

C. Heunen, N.P. Landsman, and B. Spitters. A topos for algebraic quantum theory. *Communications in Mathematical Physics*, 291(1):63–110, 2009.

C. Heunen, N.P. Landsman, and B. Spitters. Bohrification. In H. Halvorson, editor, *Deep Beauty*, pages 271–315. Cambridge University Press, 2011.

R. Hinze. Kan extensions for program optimisation or: Art and Dan explain an old trick. In J. Gibbons and P. Nogueira, editors, *MPC*, volume 7342 of *Lecture Notes in Computer Science*, pages 324–362. Springer, 2012.

R. Hinze and D. Marsden. Beautiful proofs using string diagrams. In preparation, 2014.

B. Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. Elsevier, 1999.

B. Jacobs. The temporal logic of coalgebras via Galois algebras. *Mathematical Structures in Computer Science*, 12(6):875–903, 2002.

B. Jacobs. Bases as coalgebras. In A. Nerode and Y.V. Matiyasevich, editors, *Logical Foundations of Computer Science*, volume 813. Springer, 2011a.

B. Jacobs. Coalgebraic walks, in quantum and Turing computation. In M. Hofmann, editor, *FOSSACS*, volume 6604 of *Lecture Notes in Computer Science*, pages 12–26. Springer, 2011b.

B. Jacobs. New directions in categorical logic, for classical, probabilistic and quantum logic. *CoRR*, abs/1205.3940, 2012a.

B. Jacobs. Introduction to coalgebra, towards mathematics of states and observation. Book in preparation, 2012b.

B. Jacobs and J.J.M.M Rutten. An introduction to (co)algebras and (co)induction. In D. Sangiorgi and J.J.M.M. Rutten, editors, *Advanced Topics in Coalgebras and Coinduction*. Cambridge University Press, 2011.

B. Jacobs and A. Sokolova. Exemplaric expressivity of modal logics. *J. Log. Comput.*, 20(5):1041–1068, 2010.

G. M. Kelly and R. Street. Review of the elements of 2-categories. In *Sydney Category Seminar*, volume 420 of *Lecture Notes in Mathematics*, pages 75–103. Springer, 1974.

A. Kissinger, A. Merry, B. Frot, B. Coecke, D. Quick, L. Dixon, M. Soloviev, R. Duncan, and V. Zamdzhiev. The Quantomatic tool website. https://sites.google.com/site/quantomatic/, 2014.

H. Kleisli. Every standard construction is induced by a pair of functors. *Proc. Am. Math. Soc.*, 16:544–546, 1965.

B. Klin. *An Abstract Coalgebraic Approach to Process Equivalence for Well-Behaved Operational Semantics*. PhD thesis, University of Aarhus, 2004.

B. Klin. Coalgebraic modal logic beyond sets. *Electr. Notes Theor. Comput. Sci.*, 173:177–201, 2007.

A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwig-Maximilians-Universität München, 2000.

A. Kurz and D. Pattinson. Notes on coalgebras, cofibrations and concurrency. *Electr. Notes Theor. Comput. Sci.*, 33:196–229, 2000a.

A. Kurz and D. Pattinson. Coalgebras and modal logic for parameterized endofunctors. Technical Report SEN-R0040, CWI, 2000b.

A. Kurz and J. Rosickỳ. Operations and equations for coalgebras. *Mathematical Structures in Computer Science*, 15(01):149–166, 2005.

A. Kurz and J. Rosický. Strongly complete logics for coalgebras. *Logical Methods in Computer Science*, 8, 2012.

A. Kurz and J. Velebil. Enriched logical connections. *Applied Categorical Structures*, 21(4):349–377, 2013.

F.W. Lawvere. Adjointness in foundations. *Dialectica*, 23, 1969.

F.W. Lawvere. Equality in hyperdoctrines and the comprehension schema as an adjunction. In A. Heller, editor, *Applications of Categorical Algebra*, pages 1–14. American Mathematical Society, 1970.

S. MacLane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer, 1998.

M. Makkai and R. Paré. *Accessible categories: the foundations of categorical model theory*, volume 104. American Mathematical Society, 1989.

G. Malcolm. *Algebraic Data Types and Program Transformation*. PhD thesis, Rijksuniversiteit Groningen, 1990a.

G. Malcolm. Data structures and program transformation. *Sci. Comput. Program.*, 14(2-3):255–279, 1990b.

E.G. Manes. *Algebraic Theories*, volume 26 of *Graduate Texts in Mathematics*. Springer, 1976.

D. Marsden. Coalgebras with symmetries and modelling quantum systems. In R. Heckel and S. Milius, editors, *CALCO*, volume 8089 of *Lecture Notes in Computer Science*, pages 205–219. Springer, 2013a.

D. Marsden. Fibred coalgebraic logic and quantum protocols, 2013b. To appear, proceedings of QPL 2013.

D. Marsden. Category theory using string diagrams. *arXiv preprint arXiv:1401.7220*, 2014.

P.-A. Melliès. Functorial boxes in string diagrams. In Z. Ésik, editor, *CSL*, volume 4207 of *Lecture Notes in Computer Science*, pages 1–30. Springer, 2006.

P.-A. Melliès. Game semantics in string diagrams. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 481–490. IEEE, 2012.

N.D. Mermin. *Quantum Computer Science: An Introduction*. Cambridge University Press, 2007.

E. Moggi. Notions of computation and monads. *Inf. Comput.*, 93(1):55–92, 1991.

L. Moss. Coalgebraic logic. *Ann. Pure Appl. Logic*, 96, 1999.

M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition, 2010.

P.H. Palmquist. The double category of adjoint squares. In J. W. Gray and S. MacLane, editors, *Reports of the Midwest Category Seminar V*, volume 195 of *Lecture Notes in Mathematics*. Springer, 1971.

D. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer Berlin Heidelberg, 1981.

D. Pattinson. Translating logics for coalgebras. In M. Wirsing, D. Pattinson, and R. Hennicker, editors, *WADT*, volume 2755 of *Lecture Notes in Computer Science*, pages 393–408. Springer, 2002.

D. Pattinson. Coalgebraic modal logic, soundness, completeness and decidability of local consequence. *Theoretical Computer Science*, 309:177–193, 2003.

D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame Journal of Formal Logic*, 45(1):19–33, 2004.

W. Phoa. An introduction to fibrations, topos theory, the effective topos and modest sets. Technical report, University of Edinburgh, 1992.

J. Power and H. Watanabe. Combining a monad and a comonad. *Theor. Comput. Sci.*, 280(1-2):137–162, 2002.

J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.

J. Sakarovitch. *Elements of automata theory*. Cambridge University Press, 2009.

L. Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. *Theor. Comput. Sci.*, 390(2-3):230–247, 2008.

P. Selinger. A survey of graphical languages for monoidal categories. In B. Coecke, editor, *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, pages 289–355. Springer, 2011.

A. Silva, F. Bonchi, M.M. Bonsangue, and J.J.M.M. Rutten. Generalizing determinization from automata to coalgebras. *Logical Methods in Computer Science*, 9 (1), 2013.

M. Stay and J. Vicary. Bicategorical semantics for nondeterministic computation. *CoRR*, abs/1301.3393, 2013.

R. Street. The formal theory of monads. *J. Pure Appl. Algebra*, 2:149–168, 1972.

R. Street. Low-dimensional topology and higher-order categories. In *Proceedings of CT95*, 1995.

R. Street and S. Lack. The formal theory of monads II. *J. Pure Appl. Algebra*, 175: 243–265, 2002.

T. Streicher. Fibred categories. *Lecture notes, Spring School on Categorical Methods in Logic and Computer Science, LMU, Muenchen*, 1999.

I. Stubbe and B. van Steirteghem. Propositional systems, Hilbert lattices and generalized Hilbert spaces. In K. Ensegger, D. M. Gabbay, and D. Lehmann, editors, *Handbook of Quantum Logic and Quantum Structures: Quantum Structures*, pages 477–523. Elsevier, 2007.

H. Tews. Coalgebras for binary methods. *Electr. Notes Theor. Comput. Sci.*, 33:316, 2000.

A.J.M. van Gasteren. *On the Shape of Mathematical Arguments*, volume 445 of *Lecture Notes in Computer Science*. Springer, 1990.

J. Vicary. Higher semantics of quantum protocols. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 606–615. IEEE, 2012.

P. Wadler. Monads for functional programming. In J. Jeuring and E. Meijer, editors, *Advanced Functional Programming*, volume 925 of *Lecture Notes in Computer Science*, pages 24–52. Springer, 1995.

Index