

IAN JOSEPH PRICE

WEIHRAUCH REDUCIBILITY FROM A CATEGORICAL LENS



Swansea University
Prifysgol Abertawe

Weihrach Reducibility from a Categorical Lens

Ian Joseph Price

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy



Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

October 2026

Abstract

Over the past 80 years, category theory, a general theory of mathematical structures, has revolutionised many fields of mathematics, most spectacularly in algebraic topology and algebraic geometry. Its influence on computer science, on the other hand, is more mixed. While areas such as functional programming and semantics have adopted its techniques and vocabulary, the same cannot be said of other areas such as computability theory, with limited exceptions such as the work of Yanofsky ¹.

Much of the work in Weihrauch complexity ², an area of computable analysis, goes into comparing problems of interest some well-known benchmark problems using computable reductions. Such benchmark problems include determining if an infinite binary sequence is constantly 0, finding infinite paths in infinite trees, and generalisations of the pigeonhole principle. It is also possible to look at order-theoretic properties of the lattice of problems, or to study the algebraic structure given by new operators, such as the sequential composition of problems and its iterations.

This thesis contends that the Weihrauch lattice is a quotient of the category of containers ³ over a well-chosen subcategory of represented spaces. As such, much of the algebraic structure of the Weihrauch lattice exists for general reasons, rather than being tied to type-2 computability. This means that Weihrauch reducibility can be straightforwardly adapted to settings such as transfinite computability. It also adds another application of containers to logic and computer science, alongside polynomial functors ⁴, lenses in functional programming ⁵, and in the topological complexity of algorithms ⁶.

In addition to showing that Weihrauch reductions are container morphisms, we are able to account for algebraic operations on degrees, such as the lattice structure, parallel product, and composition of degrees. In order to handle composition, a weak composition of containers is introduced, which relies on the full subcategory of projectives. We also study fixpoints of endofunctors of containers, which we use to define parallelisation and iteration operations. Finally, we are able to give a category-theoretic treatment of strong reducibility using an extension of containers, known as dependent adaptors, due to Braithwaite et al ⁷.

¹ Yanofsky, *Theoretical Computer Science for the Working Category Theorist*.

² Brattka, Gherardi and Pauly, 'Weihrauch Complexity in Computable Analysis'.

³ Abbott, Altenkirch and Ghani, 'Categories of Containers'.

⁴ Gambino and Kock, 'Polynomial functors and polynomial monads'.

⁵ Kmett and contributors, *lens: Lenses, Folds and Traversals*.

⁶ Hirsch, 'Applications of topology to lower bound estimates in computer science'.

⁷ Braithwaite, Capucci, Gavranović, Hedges and Rischel, 'Fibre optics II'.

Declarations

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed (candidate)

Date

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed (candidate)

Date

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed (candidate)

Date

The University's ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

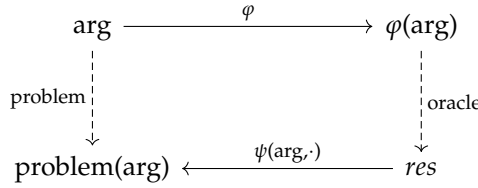
Signed (candidate)

Date

Preface

IT'S A TRUISM in academia that you never end up doing the work you set out to do. When I came to Swansea, it was to work with Cécilia Pradic on “Implicit Automata in Typed Lambda Calculi”⁸; that work won't appear anywhere in this thesis, but we did write a paper on planar automata⁹.

Then, in July 2024, I attended the [Computability and Complexity in Analysis](#) conference, which Swansea University was hosting. There were several talks on Weihrauch reducibility from people at Swansea, and I'm sure I'd heard talks on it in our theory seminar. However, I was paying attention to Manlio Valenti's talk, and I saw a familiar-looking diagram and thought, “Huh, that's a lens”.



I had a short chat with Cécilia to confirm my suspicions, talking about this connection and how the operators on Weihrauch degrees relate to those on polynomial functors. It turns out she had already had this idea, and I realised she had previously mentioned a “project with polynomials” to me during a discussion on how automata can be viewed as polynomial functors¹⁰. Her work later became a paper¹¹ which used this connection primarily as an inspiration, rather than working out the full correspondence. She also encouraged me to write this connection up as a blog post, which I did¹², but the relationship between reducibility and lenses seemed straightforward¹³, and we didn't think there was a paper in it. We were wrong about that¹⁴.

It's now been a little over two years since noticing the Weihrauch reducibility/lenses connection. The work has ballooned from one paper¹⁵, to two¹⁶, to my entire PhD Thesis. A planned paper with Cécilia on comparison-free polyregular functions¹⁷ never got written up. This was only supposed to be a blog post.

Ian Joseph Price
October 2026

⁸ Nguyễn, Nôûs and Pradic, ‘Implicit automata in typed λ -calculi II: streaming transducers vs categorical semantics’; Nguyễn and Pradic, ‘Implicit automata in typed λ -calculi I: aperiodicity in a non-commutative logic’.

⁹ Pradic and Price, ‘Implicit automata in λ -calculi III: affine planar string-to-string functions’.

¹⁰ Niu and Spivak, *Polynomial Functors: A Mathematical Theory of Interaction*, Chapter 4.

¹¹ Pradic, *The equational theory of the Weihrauch lattice with (iterated) composition*.

¹² Price, *Weihrauch Reducibility as a Lens*.

¹³ I believe anyone in the intersection of computable analysis & category theory could have seen it — it's just that intersection is very small!

¹⁴ Thanks to Arno Pauly for suggesting to use to write the work up more formally.

¹⁵ Pradic and Price, ‘Weihrauch Problems as Containers’.

¹⁶ Pradic and Price, ‘Problems with Fix-points of Polynomials of Polynomials’.

¹⁷ Nguyễn, Nôûs and Pradic, ‘Comparison-Free Polyregular Functions’.

Acknowledgements

I have always depended on the
kindness of strangers.

Blanche DuBois
“A Streetcar Named Desire”

IT IS TYPICAL to acknowledge the support of family, friends and colleagues, and I will get to them, but first I need to acknowledge the work of many medical and mental health professionals without whom I would not be here (literally). Unfortunately, I cannot name most of the people who spoke to me at my worst, and who talked me out of the most serious decision a person can make. As such, if this work is of any value to you, I can only encourage you to make a small donation to [the Samaritans](#). For those who I can name, I can only thank Dr Adam, Dr Heatley, Anna Shea, Lynn, the University of Aberdeen counselling service, and the Wellbeing and Disability service at Swansea University, especially Gemma Price¹⁸. I made it thanks to all of your hard work.

I’d like to thank Ben Martin of Aberdeen University for allowing me significant leeway to shape my Honours project and for keeping me employed during Covid; Richard Hepworth for his discussions and friendship; and Nir Oren for making sure I got my degree when I was willing to throw it away.

Thanks to my brothers and sisters for putting up with me all this time; it’s been difficult since our parents died, but we’ve done what we could. I wish my mum and dad could be here to see me finally achieve something. My friends Freya Bradley, Sarah Caird, James Dykes, David Emslie, Denis Kneip and Andrew McManus kept me sane through RPG and movie nights.

I’d like to acknowledge all the lovely people that I’ve met at various conferences and graduate schools, particularly the gang from Birmingham (Abhishek De, George Kaye, Bruno da Rocha Paiva, & co.), who were always good for a laugh and a pint. The Applied Category Theory 2022 conference was a surprisingly pivotal point in my life. It was where I met Cécilia for the first time, leading me to the PhD I ended up doing under her. The people at Strathclyde university have always been supportive. Special thanks have to go to Fredrik Nordvall Forsberg for putting Cécilia and me in touch, and to Jules Hedges¹⁹ for sharing their draft work with me. Thanks also to Andrej Bauer, Vasco Brattka for hosting me during a research visit, Tom de Jong, Alex Kavvos, Takayuki Kihara, David Spivak

Colophon: This document was prepared using the [Modernized Tufte L^AT_EX](#) package. Some diagrams were prepared using the [lpe editor](#).

¹⁸ No relation.

¹⁹ No longer at Strathclyde.

and everyone else who had a kind word. Thanks to Nina Kalinina for Russian help.

Doing a PhD is tough and you need other people to go through it with you, so thanks go to the computability group at Swansea (Arno Pauly, Manlio Valenti, Eike Neumann) and to my fellow PhD students (including Genevieve Clifford, Harry Bryant, Sam Oliver and especially Margret Tembo). Arno stepped in during Cécilia's leave in my first year and after Cécilia fell victim to the latest round of staff redundancies. His help with the Weihrauch literature was especially appreciated.

But most of all, I'd like to acknowledge the hard work, prompt replies and patience of my supervisor, Cécilia Pradic. There really aren't enough words, but hopefully this thesis suffices.

I received funding under a Swansea University/EPSRC PhD Scholarship and under the  Marie Curie Staff Exchange Network “New Frontiers for Computability”.

I do not acknowledge the use of Generative AI, because none was used. All thinking was done manually.

Contents

1	<i>Introduction</i>	1
1.1	<i>Chapter Overview & Contributions</i>	3
1.2	<i>Related Work</i>	6
1.3	<i>Mathematical Prerequisites</i>	9
1.4	<i>Some Comments On Notation</i>	9
2	<i>Categories for Computable Analysis</i>	11
2.1	<i>Partial Combinatory Algebras</i>	11
2.2	<i>Kleene's Second Model $\mathcal{K}_2$</i>	14
2.3	<i>Assemblies</i>	16
2.4	<i>Partitioned Modest Sets</i>	21
3	<i>Weihrauch Problems as Containers</i>	27
3.1	<i>Type-2 Computability</i>	27
3.2	<i>Introduction to Weihrauch Reducibility</i>	30
3.3	<i>Alternative Definitions of Weihrauch Reducibility</i>	34
3.4	<i>An Introduction to Containers</i>	36
3.5	<i>Weihrauch Problems are Containers over Partitioned Modest Sets</i>	41
3.6	<i>Extended Weihrauch Degrees and Containers over Partitioned Assemblies</i>	42
4	<i>The Algebraic Structure of Problems</i>	47
4.1	<i>Symmetric Operators on Containers</i>	47
4.2	<i>Composition Product</i>	52
4.3	<i>Embedding the Medvedev Degrees</i>	57

5	<i>Operators Definable by Fixpoints</i>	61
5.1	<i>Extending the Parallel Product</i>	61
5.2	<i>Fixpoints of Polynomial Functors over Containers</i>	62
5.3	<i>Examples of Operators Definable by Fixpoints</i>	70
5.4	<i>Problems Definable as Closed ζ-Expressions</i>	77
6	<i>Strong Reductions as Adaptors</i>	83
6.1	<i>Internal Relations</i>	83
6.2	<i>Adaptors</i>	87
6.3	<i>Strong Reductions are Adaptors</i>	90
6.4	<i>Tensor Products of Adaptors</i>	93
6.5	<i>Cylinders</i>	94
6.6	<i>The Strong Weihrauch Lattice</i>	96
7	<i>Conclusion</i>	105
7.1	<i>Weihrauch Reducibility à la Carte</i>	105
7.2	<i>Other Weihrauch Reducibilities</i>	107
7.3	<i>A Modest Proposal</i>	108
7.4	<i>Directions for Further Work</i>	109
A	<i>Category Theory</i>	111
A.1	<i>Basics</i>	111
A.2	<i>(Co)algebras</i>	112
A.3	<i>Factorisation Systems</i>	115
A.4	<i>Fibrations</i>	116
B	<i>Free Monoid on $\text{Cont}(C)$</i>	121
	<i>Bibliography</i>	129

1 Introduction

WEIHRAUCH REDUCIBILITY is “a framework to classify the uniform computational content of problems and theorems from analysis and other areas of mathematics”¹. In particular, it provides a notion of solvability up to an oracle in type-2 computability, where Turing machines are augmented to take infinite sequences as input and output infinite sequences. It is not immediately apparent how to even give an oracle an infinite input, as one surely cannot write the entire output to a working tape before instructing the machine to use its oracle. Weihrauch reducibility limits access to the oracle, allowing only a single use, and so offers a means of sidestepping this issue entirely: reductions are split into two pieces, corresponding to a type-2 computation generating the problem we wish the oracle to solve, and a type-2 computation deriving the final solution from the oracle’s response. It doesn’t matter what an oracle is; what matters is how you use it.

The notion of problem in Weihrauch reducibility is a partial multi-valued function on Baire space. Since $\mathbb{N}^{\mathbb{N}}$ is uncountable, it is possible to code infinite structures not possible in ordinary Church-Turing (type-1) computability², but type-2 machines themselves are modest extensions of type-1: the ability to do classically noncomputable computations comes from access to noncomputable inputs, not the machines.

Weihrauch reducibility has been successful as a programme, as can be seen by the extensive bibliography maintained by Brattka³, with results comparing the relative difficulty of mathematical theorems such as the Bolzano-Weierstrass, König’s lemma, LPO, Ramsey’s theorem, choice, the intermediate value theorem and solving initial-value problems. We cannot hope to give a full account of the accomplishments of Weihrauch reducibility here; we defer to Brattka’s bibliography and the recent survey of Brattka et al.⁴. Figure 3.4 shows a fragment of the Weihrauch lattice.

As you may expect, Weihrauch reducibility is similar to Reverse Mathematics⁵, as has been pointed out by Gherardi & Marcone⁶ and Dorais et al.⁷ Reverse mathematics studies systems of second-order arithmetic with the aim of determining which axioms are necessary to prove a given theorem. A theorem P reduces to a theorem Q if there is a proof of P in RCA_0 assuming Q . Reverse mathematics also has a common suite of test systems, the “big five”, into which many theorems of mathematics fall. According to Brattka et al.⁸: “most classifications in the Weihrauch lattice can be seen as uniform and resource sensitive refinements of classifications in reverse mathematics”, and the recent textbook of Dzhafarov and Mummert⁹ includes material on Weihrauch reducibility.

¹ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’.

² Most importantly, the real numbers.

³ Brattka, *Bibliography on Weihrauch Complexity*.

⁴ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’.

⁵ Friedman, ‘Some systems of second order arithmetic and their use’; Simpson, *Subsystems of second order arithmetic*.

⁶ Gherardi and Marcone, ‘How Incomputable is the Separable Hahn-Banach Theorem?’

⁷ Dorais, Dzhafarov, Hirst, Mileti and Shafer, ‘On uniform relationships between combinatorial problems’.

⁸ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’.

⁹ Dzhafarov and Mummert, *Reverse Mathematics*.

This thesis is not, however, an attempt to prove new separations and equivalences in the Weihrauch lattice; rather, it starts from the observation that similar kinds of “call and response” structures have appeared elsewhere in mathematics under the name “Containers”.

Containers are an abstraction that gets reinvented time and again; a blog post by Hedges ¹⁰ claims at least 9 independent rediscoveries. The modern theory termed “Containers” comes from the work of Abbott et al. ¹¹, who were attempting to provide a satisfying category-theoretic treatment of datatypes in functional programming languages. The central idea is that polymorphic type families, e.g., labelled binary trees, can be seen as having a “shape” and “positions” within those shapes at which data can be attached. This observation goes back further to the “shapely types” of Jay & Cockett ¹². This factorisation can be exploited in many algorithms, e.g., reflecting a binary tree only shuffles around the positions into a new shape and doesn’t touch the data at those positions. Indeed, under the name “functional references” (or “lenses”¹³), they are part of the popular Haskell `Control.Lens` library ¹⁴. Category theorists are most familiar with containers under the name “Polynomial Functors”¹⁵, and we will explicitly treat that connection in [Section 3.4.1](#).

A central claim of this thesis is that computability theorists can claim another independent discovery. Weihrauch reducibility is an instance of the theory of Containers. It is time for computability theorists and category theorists to talk to one another!

In fact, they can claim two! In Hirsch’s PhD Thesis ¹⁶, Definition 2.1 is a container on topological spaces and Definition 3.4 is a container morphism. We are not aware of this notion being picked up by others in computability; however, this thesis is briefly mentioned by Brattka et al. ¹⁷ as follows: “In a more abstract categorical setting a concept related to Weihrauch reducibility has independently been studied by Hirsch”.

That Weihrauch reductions are container morphisms is no mere coincidence. Not only does this definition carry over, but both communities have invented the same algebraic operations. For example, the diamond operator of Weihrauch reducibility was introduced in 2016 ¹⁸, whereas its corresponding operation in category theory, the free monad on a functor, had been studied since at least 1980 ¹⁹! A table listing the correspondences in this thesis is provided in [Figure 1.1](#).

The second claim of this thesis is that, by working in a category-theoretic setting, we will be able to capture more reducibilities than just Weihrauch reducibility. In order to achieve this, we state theorems in terms of generic categorical properties, rather than the specifics of type-2 computability, and derive results about Weihrauch reducibility as corollaries using a small number of theorems.

The final contention of this thesis is that recognising this relationship generates interesting new category theory for the theory of containers. This will be evidenced by the construction of a weak composition product, the invention of fibred polynomial functors and the use of dependent adaptors to categorify strong reductions.

¹⁰ Hedges, *Lenses for philosophers*.

¹¹ Abbott, Altenkirch and Ghani, ‘Categories of Containers’.

¹² Jay and Cockett, ‘Shapely types and shape polymorphism’.

¹³ As such, the title of this thesis is a play on words.

¹⁴ Kmett and contributors, *lens: Lenses, Folds and Traversals*.

¹⁵ The author first became acquainted with Containers as an undergraduate, when he did his honours project on Joyal’s Combinatorial Species (Bergeron, Labelle and Leroux, *Combinatorial species and tree-like structures*; Joyal, ‘Une Théorie Combinatoire des Séries Formelles’), which is a similar category-theoretic structure to Containers. Suffice to say that they are related to *analytic functors* (Gambino and Kock, ‘Polynomial functors and polynomial monads’, §1.20).

¹⁶ Hirsch, ‘Applications of topology to lower bound estimates in computer science’.

¹⁷ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’.

¹⁸ Neumann and Pauly, ‘A topological view on algebraic computation models’.

¹⁹ Kelly, ‘A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on’.

	Category Theory	Computable Analysis
Categories	$\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})^{\text{op}}$ $\text{Cont}_{\text{Ans}}(\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$ $\text{Cont}_{\text{Ans}}(\text{pMod}(\mathcal{K}_2))$ $\text{Cont}_{\text{Ans}}(\text{pAsm}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$ $\text{DAdt}(\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$	Medvedev Degrees Weihrauch Degrees Continuous Weihrauch Degrees Extended Weihrauch Degrees Strong Weihrauch Degrees
Operators on Weihrauch Degrees	Product $p \times q$ Coproduct $p + q$ Tensor $p \otimes q$ Composition $p \star_p q$ Finite Parallelisation p^* Infinite Parallelisation $\hat{q}$ Free Monad on p Infinite Iteration p^∞	Meet $p \sqcap q$ Join $p \sqcup q$ Parallel Product $p \times q$ Composition $p \star q$ Finite Parallelisation p^* Infinite Parallelisation $\hat{q}$ Diamond operator $p^\diamond$ Inverse Limit p^∞
Operators on Strong Degrees	Tensor $p \otimes q$ Cylinder $\text{Cyl}(p)$ Products $p \times q$ Weak Coproducts $p \boxplus q$	Parallel Product $p \times q$ $p \times \text{id}$ Meet $p \sqcap q$ Join $p \sqcup q$

1.1 Chapter Overview & Contributions

THIS DOCUMENT is comprised of five main content chapters. First, we will recapitulate background information on type-2 computability, realisability, Weihrauch reducibility and containers. We then proceed to flesh out the correspondence between the algebra of degrees and the algebra of containers in some detail. To account for operators on degrees that can be defined via fixpoint equations, we introduce fibred polynomial functors and show that natural degrees can be expressed in a language of ζ -expressions. Finally, we turn to strong reducibility, which will be handled by a container-like construction called a dependent adaptor.

Figure 1.1: Dictionary linking category-theoretic objects on the left to computable analysis objects on the right.

1.1.1 Chapter 2 — Categories for Computable Analysis

SINCE WE CLAIM that the algebra of Weihrauch degrees does not rely on the specifics of type-2 computability, this chapter introduces the more general setting of categorical realisability. We first introduce partial combinatory algebras, which generalise the calculus of combinators familiar from the λ -calculus. We will not be able to avoid type-2 computability, so we treat Kleene's second model $\mathcal{K}_2$ in some detail. We then proceed to introduce categories of assemblies; assemblies are sets augmented with a relation that tells us which objects in the PCA code which elements of the set. We spend some time developing the basic properties of these categories, proving that they are locally cartesian closed, regular and come with useful factorisation systems. We end by investigating the properties of projective modest sets, where codes and elements are in a one-to-one correspondence. Many of the properties of assemblies carry over, but not all do; e.g., projective modest sets are only weakly locally cartesian closed.

This chapter does not contain any new material, but provides the essential background on which the rest of the thesis rests. We draw mainly on the monograph of van Oosten²⁰ and the PhD thesis and notes of Bauer²¹. We are also heavily indebted to the lecture notes of de Jong²², from whom the author first learned this material at the 2024 Midlands Graduate School.

1.1.2 Chapter 3 — *Weihrauch Problems as Containers*

WE BEGIN BY discussing type-2 computability and its relation to Kleene’s second model. We then give an introduction to Weihrauch reducibility, providing all the main definitions and a few basic examples of reductions. Next, we discuss different definitions of Weihrauch reducibility that induce the same order, building up to the definition of a container morphism. We then give a brief introduction to containers, sketching out the relationship between containers and polynomial functors. Finally, we have enough to state and prove the main theorems of this chapter, connecting Weihrauch reductions to container morphisms. We also characterise continuous reducibility and the extended Weihrauch degrees of Bauer²³.

The content of the section on type-2 computability is well-known, with the results mainly coming from Bauer’s thesis²⁴ or Pauly²⁵. The author has never seen an explicit proof of [Proposition 3.1.3](#), but it is implicit in both Bauer and Pauly. The section on Weihrauch reducibility is essentially a précis of Brattka et al.²⁶ [Definition 3.3.2](#) is widely used in the Weihrauch literature, e.g., by Dorais et al.²⁷ [Definition 3.3.7](#) is novel, first appearing in my paper with Pradic²⁸. The section on containers draws heavily on the work of Abbott et al.²⁹, Gambino & Kock³⁰ and Weber³¹. The characterisations of the Weihrauch degrees ([Theorem 3.5.3](#), [Theorem 3.5.4](#), [Theorem 3.6.3](#)) and the definition of answerable containers ([Definition 3.5.1](#)) are due to the author and Pradic³².

1.1.3 Chapter 4 — *The Algebraic Structure of Problems*

THIS CHAPTER EXTENDS the dictionary relating Weihrauch degrees to containers by exploring the algebraic structure of both theories. We first show that containers form an extensive category, which immediately implies that Weihrauch degrees have a lattice structure. Next, we define a monoidal product on containers which corresponds to the parallel product of degrees. We then give the composition product of containers, corresponding to composition of their container functors, but explain that this cannot be applied to categories of partitioned modest sets. However, we can define a *weak* composition product $\star_p$, and this can be shown to give the composition of Weihrauch degrees. Finally, we discuss the Medvedev degrees, recovering their expected embeddings into the Weihrauch degrees by the adjoints to the shape fibration.

The four monoidal products on containers that we discuss are well-known, appearing in Abbott’s thesis³³ and in surveys of polynomial functors³⁴. Pradic and I showed that three of these ($+$, $\times$ and $\otimes$) correspond to important operators on the degrees in our CiE paper³⁵. We also gave

²⁰ van Oosten, *Realizability: An Introduction to its Categorical Side*.

²¹ Bauer, ‘The realizability approach to computable analysis and topology’, *Notes on Realizability*.

²² de Jong, *Categorical Realizability*.

²³ Bauer, ‘Instance reducibility and Weihrauch degrees’.

²⁴ Bauer, ‘The realizability approach to computable analysis and topology’.

²⁵ Pauly, ‘On the topological aspects of the theory of represented spaces’.

²⁶ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’.

²⁷ Dorais, Dzharov, Hirst, Mileti and Shafer, ‘On uniform relationships between combinatorial problems’.

²⁸ Pradic and Price, ‘Weihrauch Problems as Containers’.

²⁹ Abbott, Altenkirch and Ghani, ‘Categories of Containers’.

³⁰ Gambino and Kock, ‘Polynomial functors and polynomial monads’.

³¹ Weber, ‘Polynomials in categories with pullbacks’.

³² Pradic and Price, ‘Weihrauch Problems as Containers’.

³³ Abbott, ‘Categories of Containers’.

³⁴ Gambino and Kock, ‘Polynomial functors and polynomial monads’; Niu and Spivak, *Polynomial Functors: A Mathematical Theory of Interaction*.

³⁵ Pradic and Price, ‘Weihrauch Problems as Containers’.

there the novel construction of $\star_p$. At that time, we did not give the connection to the Medvedev degrees. The adjunctions to the shape fibration were already known to category theorists³⁶, and the left adjoint was known to computable analysts.

1.1.4 Chapter 5 — Operators Definable by Fixpoints

HAVING ESTABLISHED a basic complement of operators on the degrees, we turn to those definable as fixpoints of monotone operators. To generalise this, we introduce fibred polynomial functors. We prove a fixpoint theorem for these functors, showing that they have not just least and greatest fixpoints (Theorem 5.2.8), but an intermediate fixpoint ζ . We give a suite of basic examples of fibred polynomial functors, and these will be enough to set up a grammar of ζ -expressions in which to express operators. We then give important operators from the Weihrauch literature, such as finite and infinite parallelisations, the diamond operator and the inverse limit, as fixpoints of fibred polynomial functors. Using the answerable part of a problem, we are able to express many examples of Weihrauch degrees as closed ζ -expressions.

That P^* is definable by fixpoints is well-known; that $\hat{P}$ is also a fixpoint does not appear to be. The definition of fibred polynomial functor is novel and is due to the author and Pradic, first appearing in our LICS paper³⁷. Fixpoints of polynomial functors have been well-studied³⁸, but we are not aware of any systematic study of fixpoints of endofunctors over $\text{Cont}(\mathcal{C})$. Theorem 5.2.8, establishing the existence of fixpoints, is similar to other fixpoint theorems over total categories of fibrations³⁹; however, they all operate on *posetal* fibrations, whereas we do not need that restriction. We do not believe that the ζ fixpoint exists in the literature on fixpoints. The operators in Figure 5.2 are all known operators on Weihrauch degrees⁴⁰, except for $\tilde{P}$, which is trivial in that setting. That $P^\diamond$ is the free monad on P is well-known to category theorists⁴¹. The observation that Weihrauch problems might be expressible by closed ζ -expressions and an answerable part operator is due to the author, who was convinced that WKL could be expressed in this manner, but the proof that fibred polynomial functors restrict to answerable containers is due to Pradic⁴².

1.1.5 Chapter 6 — Strong Reductions as Adaptors

WE BEGIN BY reconsidering bundles (families) as our representation of problems. We introduce the concept of an internal relation in a category, and develop the basic properties of relations; in particular, we show that internal relations in $\mathcal{C}$ are fibred over $\mathcal{C}$. Then, recalling that the shape fibration is the opposite of the cod fibration, we apply the opposite construction to the dom fibration on cototal relations. The total space of the fibration we construct is the category of dependent adaptors. We then show that strong Weihrauch reductions arise as the posetal reflection of the category of adaptors over represented spaces. We can then develop the algebraic operations on strong degrees, showing that parallel products exist

³⁶ Niu and Spivak, *Polynomial Functors: A Mathematical Theory of Interaction*, §5.1.

³⁷ Pradic and Price, ‘Problems with Fixpoints of Polynomials of Polynomials’.

³⁸ Abbott, Altenkirch and Ghani, ‘Categories of Containers’, ‘Containers: Constructing strictly positive types’; van den Berg and De Marchi, ‘Non-well-founded trees in categories’; Moerdijk and Palmgren, ‘Wellfounded trees in categories’.

³⁹ Bonchi, Petrişan, Pous and Rot, ‘Coinduction up-to in a fibrational setting’; Fiore, Galal and Jafarrahmani, ‘Fixpoint constructions in focused orthogonality models of linear logic’; Hasuo, Kataoka and Cho, ‘Coinductive predicates and final sequences in a fibration’.

⁴⁰ Brattka, ‘Loops, Inverse Limits and Non-Determinism’; Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’; Westrick, ‘A note on the diamond operator’.

⁴¹ Gambino and Kock, ‘Polynomial functors and polynomial monads’; Kelly, ‘A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on’.

⁴² It is for this reason that we have omitted it from this work.

and that we can recover the Galois connection between strong Weihrauch degrees and Weihrauch degrees. Lastly, we show that the category of Dependent adaptors has finite products and weak coproducts, and hence the strong degrees are a lattice. To implement weak coproducts, we use weak partial map classifiers as the completion of a space, allowing us to generalise Dzharov’s proof ⁴³.

This chapter builds on an unpublished follow-up to the preprint “Fibre Optics” ⁴⁴ by Braithwaite et al. Pradic and I learned of it from a talk by Hedges that was uploaded to YouTube ⁴⁵, and fortuitously it turned out to be perfect for strong Weihrauch reducibility. Most of [Sections 6.1](#) and [6.2](#) are based on their work ⁴⁶. In particular, the choice of domain fibration ([Proposition 6.1.9](#)), definition of the category of adaptors ([Definition 6.2.1](#)) and the adjunction between $C^{\rightarrow}$ and $\text{RelTot}(C)$ ([Proposition 6.1.10](#)) are due to them. As far as we are aware, the key role of the opposite of a fibration was first highlighted by Spivak ⁴⁷.

In this thesis, we have expanded on the work of Braithwaite et al., clarifying some of the limitations of various results, filling in proofs, and making a modest generalisation by working up to an arbitrary factorisation system. [Theorem 6.3.5](#) was previewed in a short blog post by the author ⁴⁸, but everything else, from [Section 6.3](#) onward, appears here for the first time. All proofs are my own, but the idea to use weak partial map classifiers for the completion of Dzharov was due to Pradic.

1.2 Related Work

Despite the long tradition of studying realizability and its variants from a categorical perspective, a systematic and in-depth analysis of computability-like reducibilities, such as Turing, Medvedev, Mučnik, and Weihrauch reducibilities, from a categorical perspective is still lacking. (Trotta et al. ⁴⁹)

THERE HAVE BEEN OTHER recent attempts to provide a category-theoretic account of Weihrauch reducibility, so it is important to contrast the present thesis with the three main approaches by Trotta et al., Kihara, and Bauer & Ahman. We stress that in each of these instances, the authors characterise a more general notion of reducibility than Weihrauch reducibility, usually Bauer’s extended Weihrauch reducibility, within which the Weihrauch degrees are an interesting subclass. Another approach was taken by Pauly ⁵⁰, who used enriched p -categories ⁵¹ to capture many-one reducibility. This approach was able to characterise several reducibility theories, including Weihrauch, but is very different in flavour to the others mentioned.

1.2.1 Instance Reducibility

AN INSTANCE REDUCTION is a common pattern in constructive mathematics, whereby a predicate $\phi \subseteq A$ is reducible to another $\psi \subseteq A$, written $\phi \leq_I \psi$, when $\forall x \in A. \exists y \in B. \psi(y) \implies \phi(x)$ holds. This pattern was isolated by Bauer ⁵², who proved this ordering is both a bounded distributive lattice and a Heyting algebra. This work was given in intuitionistic logic without countable choice, and so was suitable for interpretation

⁴³ Dzharov, ‘Joins in the strong Weihrauch degrees’.

⁴⁴ Braithwaite, Capucci, Gavranović, Hedges and Rischel, *Fibre optics*.

⁴⁵ Hedges, *We solved dependent optics!*

⁴⁶ Braithwaite, Capucci, Gavranović, Hedges and Rischel, ‘Fibre optics II’.

⁴⁷ Spivak, *Generalized Lens Categories via functors $C^{\text{op}} \rightarrow \text{Cat}$* .

⁴⁸ Price, *Strong Reducibility as an Adaptor*.

⁴⁹ Trotta, Valenti and de Paiva, ‘Categorifying computable reducibilities’.

⁵⁰ Pauly, ‘Many-one reductions and the category of multivalued functions’.

⁵¹ A p -category has an operation resembling the pairing of partial functions; this is not the same as the categorical product.

⁵² Bauer, ‘Instance reducibility and Weihrauch degrees’.

in the internal language of a topos. In particular, using the Kleene–Vesley topos⁵³, he defined an extended Weihrauch reducibility based on instance reductions over $\text{Asm}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ ⁵⁴. The ordinary Weihrauch degrees are the $\neg\neg$ -dense modest predicates.

By viewing instance reductions as second-order functionals, Ahman and Bauer⁵⁵ began to investigate instance reductions from a category-theoretic perspective using right comodules for a monad. In particular, the tree representations they use are *container morphisms*, and the tree monad they consider is related to the free monad on containers. The difference between their container representation of instance reductions and ours is that they use propositional containers $A \rightarrow \text{Prop}$ ⁵⁶, whereas we use containers $A \rightarrow \text{Type}$. Essentially, we work *proof-relevantly*, where the element witnessing a proposition is as important as whether or not that proposition is true. Both theories are substantially similar; e.g., they both make essential use of a weak exponential⁵⁷.

It seems to be the case that we stumbled upon the idea of using containers for reducibility independently. We characterised Bauer’s extended degrees in terms of containers in our CiE paper, but the original instance reducibility paper does not mention containers. We have no reason to believe that Ahman and Bauer knew of our work prior to Pradic speaking about it at the [2025 Dagstuhl meeting on Weihrauch reducibility](#).

1.2.2 Oracle Modalities

EARLIER, WE STATED THAT “It doesn’t matter what an oracle is; what matters is how you use it”. To that end, our protocol for interacting with an oracle was given by a container morphism. If we instead wished to have a protocol for asking multiple questions, we might specify a type former $\mathcal{O}_P$ for oracle computations, and two operations: $\text{pure} : s \rightarrow \mathcal{O}_P s$, which embeds computations that don’t use the oracle into computations that do; and $\text{ask} : \prod (a : A). (P a \rightarrow \mathcal{O}_P s) \rightarrow \mathcal{O}_P s$, which converts computations that take a query a and a continuation whose domain is the type $P a$ of oracle responses, and returns an oracle computation⁵⁸. In essence, we treat querying an oracle as a kind of effect. This is very familiar to both functional programmers and category theorists; we are using a monad! We refer to such a protocol as an “oracle modality”.

Modalities⁵⁹ on truth values have been studied for a long time; e.g., the necessity and possibility modalities form the core of modal logic. In a recent preprint, Swan⁶⁰ used oracle modalities to characterise Turing reductions in a synthetic setting inspired by modalities on homotopy type theory⁶¹. Ahman and Bauer⁶² build on this, noting that the $\mathcal{O}_P$ is the least modality above a modality for instance reducibility (Proposition 3.12), i.e., that $\mathcal{O}_P$ is the free monad on P . The free monad is precisely how we handle finitely many oracle calls in [Section 5.3](#) and the LICS paper on which it is based⁶³. They go further, showing in Corollary 3.16 that every modality they consider is an oracle modality, and, in particular, this applies to extended Weihrauch reducibility. In computable analysis, oracle modalities were considered by Pauly and de Brecht⁶⁴, although not under that name.

⁵³ van Oosten, *Realizability: An Introduction to its Categorical Side*, §4.5.

⁵⁴ This is our [Theorem 3.6.3](#).

⁵⁵ Ahman and Bauer, ‘Comodule representations of second-order functionals’.

⁵⁶ In homotopy type theory (Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*), a proposition $P \in \text{Prop}$ is a Type with at most one element

⁵⁷ Compare their Proposition 8.1 with our [Definition 4.2.10](#).

⁵⁸ The presentation here is based on Ahman and Bauer (Ahman and Bauer, *Sheaves as oracle computations*).

⁵⁹ These go by many names. Modality tends to be used for the logical concept, whereas closure operator is preferred in order theory. In topos theory, they are known as Lawvere–Tierney topologies or local operators. They are also monads on posetal/thin categories.

⁶⁰ Swan, ‘Oracle modalities’.

⁶¹ Rijke, Shulman and Spitters, ‘Modalities in homotopy type theory’.

⁶² Ahman and Bauer, *Sheaves as oracle computations*.

⁶³ Pradic and Price, ‘Problems with Fixpoints of Polynomials of Polynomials’.

⁶⁴ Pauly and de Brecht, ‘Descriptive Set Theory in the Category of Represented Spaces’.

Nobrega and Pauly ⁶⁵ also showed that every Weihrauch degree can be presented as a modality.

Independently, Lawvere–Tierney topologies in the effective topos have been considered by Kihara ⁶⁶, both to generalise Turing reducibility and later Weihrauch reducibility ⁶⁷. More recent work along this line is due to Yamada ⁶⁸.

Our work differs from the above work by developing a general fixpoint theory of endofunctors on $\text{Cont}(\mathcal{C})$, rather than working with Lawvere–Tierney topologies in a topos-theoretic setting. All the work referred to in this section handles reducibility with finitely many oracle uses in realisability toposes, which in our setting (as in Ahman–Bauer) is handled by the free monad.⁶⁹

Our use of ζ -expressions and the answerable part of a problem in [Section 5.4](#) is somewhat reminiscent of the applications of Kihara ⁷⁰ to characterise particular subtoposes, which we might think of as corresponding to particular classes of Turing degrees. We also make essential use of a game-theoretic model ⁷¹, albeit for very different reasons.

1.2.3 *Dialectica and Realisability Doctrines*

IT HAS BEEN NOTED more than once in the literature that Weihrauch reducibility was formally similar to Gödel’s Dialectica interpretation ^{72,73}. Indeed, given Dialectica’s importance to the related field of applied proof theory ^{74,75}, it would not be very surprising if it made an appearance. It is also familiar to category theorists, having been studied by de Paiva in her PhD thesis ⁷⁶. The relationship between containers and Dialectica is also well-known; essentially, container morphisms are how the Dialectica interpretation handles implication⁷⁷.

Trotta et al. ⁷⁸ study reducibilities as categorical doctrines⁷⁹, giving doctrines for Turing, Medvedev and (generalised) Weihrauch reducibility. Firstly, the doctrines they define are, essentially, neat categorical packages of the reducibility theories, whereas we view reducibility as a notion that can be *derived* from the category-theoretic structure of containers. In particular, we are able to give an account of many interesting algebraic operations on the Weihrauch degrees, rather than just the essential lattice structure, which we believe is more faithful to the spirit of Weihrauch reducibility. Second, they are interested in exploring the relation between these reducibility theories via existential and universal completions of doctrines, for example, obtaining the Dialectica doctrine as a completion of the Medvedev doctrine.

The more recent work of Maschio and Trotta ⁸⁰ continues their program by building a topos in which propositions are *extended* Weihrauch problems. We believe this work is complementary to [Section 5.4](#) of this thesis, as it potentially paves the way for giving synthetic proofs of reductions in the internal language of that topos.

⁶⁵ Nobrega and Pauly, ‘Game characterizations and lower cones in the Weihrauch degrees’.

⁶⁶ Kihara, ‘Lawvere–Tierney topologies for computability theorists’.

⁶⁷ Kihara, *Rethinking the notion of oracle: A prequel to Lawvere–Tierney topologies for computability theorists*.

⁶⁸ Yamada, *Effectful Toposes and Their Lawvere–Tierney Topologies*.

⁶⁹ The author has started thinking of using the free monad on a polynomial for many-use reducibility as the “Ljubljana–Nagoya–Swansea interpretation”.

⁷⁰ Kihara, ‘Lawvere–Tierney topologies for computability theorists’.

⁷¹ Pradic and Price, ‘Problems with Fixpoints of Polynomials of Polynomials’, §5.

⁷² Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’, §9.1.

⁷³ In personal communication, Pauly attributes this to Paolo Oliva.

⁷⁴ Kohlenbach, *Applied Proof Theory*, Chapter 8.

⁷⁵ Or, charmingly, “proof mining”.

⁷⁶ de Paiva, *The Dialectica categories*.

⁷⁷ Indeed, containers (as the opposite of the domain fibration) and Dialectica appear in the same page of Jacobs (Jacobs, *Categorical Logic and Type Theory*, Exercises 1.10.8 and 1.10.11).

⁷⁸ Trotta, Valenti and de Paiva, ‘Categorifying computable reducibilities’.

⁷⁹ Contravariant functors into the category of posets.

⁸⁰ Maschio and Trotta, ‘A topos for extended Weihrauch degrees’.

1.3 Mathematical Prerequisites

The reader is expected to be mathematically mature⁸¹, with knowledge of (Church-Turing) computability up to first-course level (e.g., following Sipser⁸²). Some familiarity with undergraduate real analysis is desirable. Basic category theory is a must, including monads, adjunctions and monoidal categories. For category theory requirements beyond Mac Lane⁸³, we include appendices on (co)algebras (Section A.2), factorisation systems (Section A.3) and fibrations (Section A.4).

1.4 Some Comments On Notation

We use the standard practice of using the pronoun “we” to include the reader, because reading mathematics is an interactive activity requiring your active participation.

This thesis freely makes use of the notation of dependent type theory, albeit in an informal style. In particular, we use $\sum$ for dependent sums and Π for dependent function types. λ -notation is used for anonymous (dependent) functions, and we often define functions by pattern-matching on inputs. We will generally use $:=$ to mean “equal by definition”. Partial application will be denoted by use of $-$, e.g., $f(3, -) := \lambda x. f(3, x)$.

The type of finite sequences (lists) with elements in A is denoted by A^* . Lists are written with square brackets $[x_0, x_1, \dots]$ and $x :: xs$ is used to prepend x to the list xs . For strings in an alphabet, we use superscripts to denote that a symbol is repeated finitely 0^n or infinitely many $0^{\mathbb{N}}$ times. The natural numbers, of course, start at 0. Subscripts will be elided whenever there is no risk of confusion. Partial functions are denoted by $f : \subseteq X \rightarrow Y$ in accordance with practice of computable analysis⁸⁴.

We have generally given diagrammatic proofs of category-theoretic theorems, although sometimes working in the internal language is just too convenient. Colour is used, but is never essential; we believe this document should be accessible to the colour-blind.

⁸¹ Whatever that means.

⁸² Sipser, *Introduction to the Theory of Computation*.

⁸³ Mac Lane, *Categories for the working mathematician*.

⁸⁴ The author generally prefers partial arrows $f : X \rightharpoonup Y$.

2 Categories for Computable Analysis

ONE OF THE MAIN CONTENTIONS of this thesis is that the algebraic structure of Weihrauch reducibility is relatively agnostic to the underlying type of computability. As such, rather than working in type-2 computability, our basic concept will be that of a partial combinatory algebra (Definition 2.1.11). We will define categories of assemblies (Proposition 2.3.4), and assembly maps will be our notion of computable map. Assemblies are rich in structure, in particular, they are finitely (co)complete, regular and locally cartesian closed (Theorem 2.3.9). To handle Weihrauch reducibility, we are interested in a full subcategory of assemblies consisting of partitioned modest sets (Definition 2.4.3). Partitioned modest sets have less structure than assemblies, not having all finite colimits, and only being weakly locally cartesian closed (Corollary 2.4.19).

2.1 Partial Combinatory Algebras

THE GREAT THING about computability is that we can describe regular “Church–Turing” computability in many ways, e.g., Turing machines, the lambda calculus, or grammars. At the heart of each, however, is the idea that “code is data” and so we have a single set of codes representing both computations and the data they operate on. Moreover, the possibility of non-terminating computations requires that we pay attention to the partiality of such operations. Accordingly, the basic definition of this section is suitably austere.

Definition 2.1.1 (PAS). ¹ A partial applicative structure (PAS) is a non-empty set $\mathbb{A}$ equipped with a partial map $\cdot : \subseteq \mathbb{A} \times \mathbb{A} \rightarrow \mathbb{A}$.

¹ van Oosten, *Realizability: An Introduction to its Categorical Side*, Definition 1.1.1.

Remark 2.1.2. We will normally introduce maps on objects like PASs at the same time as the object. However, we will have no reason to study maps between PASs in this thesis. Suffice it to say that applicative morphisms ² between PASs are not homomorphisms in the algebraic sense, but are “implementations” of one PAS in another.

² *ibid.*, Definition 1.5.3.

Fixing a PAS and a countably infinite set of variables, it is possible to build up a set of terms over that PAS.

Definition 2.1.3 (Terms). Let $\mathbb{A}$ be a PAS and V an infinite set of variables. The set of terms $\text{Term}(\mathbb{A}, V)$ is defined inductively by:

- a variable $v \in V$ is a term,
- an element $a \in \mathbb{A}$ of the PAS is a term,
- given terms s and t , $s \cdot t$ is a term.

A term is closed if it contains no variables.

Convention 2.1.4. To reduce notation, it is conventional to juxtapose elements rather than explicitly use the multiplication operator, and to treat $\cdot$ as right associative. In other words, $f x y z$ should be read as $((f \cdot x) \cdot y) \cdot z$.

Given a closed term t , we can define a relation on terms corresponding to the element of the pca it “evaluates” to.

Definition 2.1.5 (Denotes, $\downarrow$). Let t be a closed term over $\mathbb{A}$ and $a \in \mathbb{A}$. The relation³ $t \downarrow a$ is defined inductively by:

³ Pronounced “ t denotes a ”.

- $a \downarrow a$ for all $a \in A$
- $(t s) \downarrow a$ iff there are $b, c \in \mathbb{A}$ such that $t \downarrow b$, $s \downarrow c$ and $b c = a$.

We write $t \downarrow$ as shorthand for $\exists a. t \downarrow a$.

The partiality of multiplication means that not all terms denote. We would like to consider terms equivalent if they denote the same element, or if they don’t denote at all.

Definition 2.1.6 (Kleene Equality, $\simeq$). Let t, s be closed terms over $\mathbb{A}$. We write $s \simeq t$ if either neither s, t denotes, or if both denote the same element.

For a non-closed term t , we will write $t(x_1, \dots, x_n)$ to indicate that $x_1, \dots, x_n$ are (a superset of) the free variables of t , and $t(a_1, \dots, a_n)$ to indicate a substitution of elements $a_1, \dots, a_n$ of the pca for those free variables.

For non-closed terms $t(x_1, \dots, x_n), s(x_1, \dots, x_n)$, write $t \simeq s$ if and only if $t(a_1, \dots, a_n) \simeq s(a_1, \dots, a_n)$ for all $a_1, \dots, a_n \in \mathbb{A}$.

Having established all the requisite definitions, we turn to the property we want if a PAS is to be a useful notion of computation.

Definition 2.1.7 (Combinatory Complete). ⁴ A PAS $\mathbb{A}$ is combinatory complete if for any $n \in \mathbb{N}$ and any term $t(x_1, \dots, x_{n+1})$ there is an element $a \in \mathbb{A}$ such that for all $a_1, \dots, a_{n+1} \in A$, $a a_1 \dots a_n \downarrow$ and $a a_1 \dots a_{n+1} \simeq t(a_1, \dots, a_{n+1})$.

⁴ van Oosten, *Realizability: An Introduction to its Categorical Side*, Definition 1.1.2.

Lemma 2.1.8. ⁵ Let $\mathbb{A}$ be a PAS. A PAS is combinatorially complete if and only if there are elements $K, S \in \mathbb{A}$ such that:

⁵ *ibid.*, Theorem 1.1.3.

- $K a \downarrow$ and $K a b = a$ for all $a, b \in \mathbb{A}$,
- $S a b \downarrow$ for all $a, b \in \mathbb{A}$, and
- $S a b c \simeq a c (b c)$.

This property is the same combinatory completeness property familiar from the S and K combinators of combinatory logic. As such, it is possible to define all the usual basic programming niceties, such as conditionals, arithmetic and recursion, although we will omit doing so here.

Notation 2.1.9. A list of basic programming combinators we will assume include:

- Booleans $\top, \text{F}$ and a conditional operator if satisfying the equations $\text{if } \top \ a \ b \simeq a$ and $\text{if } \text{F} \ a \ b \simeq b$;
- A pairing pair and projections fst, snd satisfying $\text{fst}(\text{pair } a \ b) \simeq a$ and $\text{snd}(\text{pair } a \ b) \simeq b$;
- a copairing case and coprojections $\text{left}, \text{right}$ satisfying $\text{case}(\text{left } a) \ f \ g \simeq f \ a$ and $\text{case}(\text{right } b) \ f \ g \simeq g \ b$; and
- a fixpoint combinator fix satisfying the equation $\text{fix } f \simeq f(\text{fix } f)$.

PASs have an abstraction operator similar to the familiar λ -abstraction.

Definition 2.1.10 (Abstraction in a PAS). Let $\mathbb{A}$ be a PAS, v be a variable and t a term. We define a new term $\langle v \rangle. t$ by induction on t :

$$\begin{aligned} \langle v \rangle. v &= \text{I} = \text{S K K} \\ \langle v \rangle. x &= \text{K } x && \text{if } v \neq x \in V \\ \langle v \rangle. a &= \text{K } a && \text{if } a \in \mathbb{A} \\ \langle v \rangle. t_1 \ t_2 &= \text{S}(\langle v \rangle. t_1)(\langle v \rangle. t_2) \end{aligned}$$

It is easy to check that $(\langle v \rangle. t) \ a \simeq t(a)$.

The class of PASs satisfying completeness is important enough to merit its own name.

Definition 2.1.11 (PCA). A partial combinatory algebra (PCA) is a combinatory complete PAS.

Example 2.1.12 (Trivial PCA). The singleton set $\{\star\}$ is a PCA with the total application $\star \cdot \star = \star$. $\star$ serves as both the K and S combinators.

Example 2.1.13 (Untyped λ -calculus). The set of closed λ expressions (up to $\alpha\beta\eta$ -equivalence) with application forms a PCA, with $\text{K} = \lambda x. \lambda y. x$ and $\text{S} = \lambda f. \lambda g. \lambda x. (f \ x)(g \ x)$ as usual.

Example 2.1.14 (Kleene's 1st Model $\mathcal{K}_1$). Fix a sensible enumeration $(\phi_n)_{n \in \mathbb{N}}$ of all computable functions (Turing machines) $\mathbb{N} \rightarrow \mathbb{N}$. Then we can form a PCA $\mathcal{K}_1$ on $\mathbb{N}$ by taking our application $n \cdot m$ to mean "run machine ϕ_n on (some encoding of) the number m ". That S and K combinators exist follows by Kleene's *snm* theorem.

There are many other interesting PCAs, e.g., Scott's graph model⁶ plays an important role in Bauer's thesis⁷, however, we will content ourselves with one more important example in Section 2.2. The last concept we need at this moment is that of a filtered PCA.

Definition 2.1.15 (sub-PCA, filtered PCA).⁸ Given a PCA $(\mathbb{A}, \cdot)$ with S and K combinators, we call a subset $\mathbb{A}' \subseteq \mathbb{A}$ a sub-PCA if it is closed under application and $\text{S}, \text{K} \in \mathbb{A}'$ ⁹.

A filtered PCA is a PCA $\mathbb{A}$ with a sub-PCA $\mathbb{A}'$.

Convention 2.1.16. We will often introduce a PCA $\mathbb{A}$ and its sub-PCA $\mathbb{A}'$ at the same time by saying something like "let $\mathbb{A}' \subseteq \mathbb{A}$ be a filtered PCA".

⁶ Scott, 'Data Types as Lattices'.

⁷ Bauer, 'The realizability approach to computable analysis and topology'.

⁸ van Oosten and Zou, 'Classical and Relative Realizability', Definition 1.5.

⁹ One useful consequence of this is that terms in the sub-PCA are closed under abstraction.

2.2 Kleene's Second Model $\mathcal{K}_2$

WE NOW DEVELOP a PCA structure on $\mathbb{N}^{\mathbb{N}}$ which will be very important in the rest of this thesis, as it gives a PCA for type-2 computability¹⁰. We will take time to explore it in some depth, but before giving a formal treatment, it makes sense to first give some intuition for the PCA¹¹.

First, we consider the case $\mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}^{\perp}$, where we have added an additional “error value”¹² $\perp$. Partial continuous functions of this form are functions that either return $\perp$, or will output a number after reading only a *finite* prefix of the input. We can extend this to the general case $\mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}^{\mathbb{N}}$ by uncurrying and prepending the natural to the front of the sequence. Then, as partial continuous functions $\mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}^{\mathbb{N}}$ are in a one-to-one correspondence with elements of $\mathbb{N}^{\mathbb{N}}$ itself, multiplication $f \cdot g$ in the PCA consists of decoding the function f and applying it to g .

Definition 2.2.1 (Baire Space (Topological Space)). *The topological space $\mathbb{B}$ is the set $\mathbb{N}^{\mathbb{N}}$ equipped with the product topology, where the natural numbers $\mathbb{N}$ are given the discrete topology¹³.*

A basis for this topology is given by sets of the form

$$U_{(x_0, \dots, x_{n-1})} = \{x_0\} \times \dots \times \{x_{n-1}\} \times \mathbb{N} \times \mathbb{N} \times \dots$$

where $n, x_0, \dots, x_{n-1} \in \mathbb{N}$.

Convention 2.2.2. *We will use Greek letters $\alpha, \beta, \dots$ for elements of $\mathbb{B}$.*

Consequently, if m is in the image of a partial continuous function $f : \subseteq \mathbb{B} \rightarrow \mathbb{N}$, say $f(\alpha) = m$, then there exists some $(x_0, \dots, x_{n-1})$, such that $\alpha \in U_{(x_0, \dots, x_{n-1})} \subseteq f^{-1}(m)$. Moreover, for any $\beta \in \text{dom}(f) \cap U_{(x_0, \dots, x_{n-1})}$, $f(\beta) = m$.

We will assume the existence¹⁴ of a bijective coding function $\langle \cdot \rangle : \text{List}(\mathbb{N}) \rightarrow \mathbb{N}$, and set $\bar{\alpha}(n) = \langle \alpha(0), \dots, \alpha(n-1) \rangle$ for $\alpha \in \mathbb{B}$, $n \in \mathbb{N}$. Every $\alpha \in \mathbb{B}$ gives rise to a partial map in the following way.

Definition 2.2.3 (F_α). *Let $\alpha, \beta \in \mathbb{B}$. Define $F_\alpha : \subseteq \mathbb{B} \rightarrow \mathbb{N}$ by $F_\alpha(\beta) = k$ if there is an n such that $\alpha(\bar{\beta}(n)) = k+1$ and $\alpha(\bar{\beta}(m)) = 0$ for all $m < n$.*

Remark 2.2.4. *The intuition behind F_α is that a computation is that you use the input β as the description of a path in an infinite tree represented by α , where leaves are labelled with outputs. You ask for larger and larger prefixes $\bar{\beta}(m)$ until you reach the end of a finite path, signalled by $\alpha(\bar{\beta}(m)) \neq 0$, at which point you output the value at the leaf.*

Proposition 2.2.5. *Let $f : \subseteq \mathbb{B} \rightarrow \mathbb{N}$ be a partial function. Then $f = F_\alpha$, for some $\alpha \in \mathbb{B}$, if and only if f is continuous on $\text{dom}(f)$ and $\text{dom}(f)$ is open.*

Proof. If $\beta \in f^{-1}(\{k\})$ for some $k \in \mathbb{N}$, then there is a least $n_\beta \in \mathbb{N}$ such that $\alpha(\bar{\beta}(n_\beta)) = k+1$, and hence $\beta \in U_{(\beta(0), \dots, \beta(n_\beta-1))}$. Moreover, f is defined, and equal to k , for every γ in $U_{(\beta(0), \dots, \beta(n_\beta-1))}$. As such, we can see that $f^{-1}(\{k\})$ is the open set $\bigcup_{\beta \in f^{-1}(k)} U_{(\beta(0), \dots, \beta(n_\beta-1))}$. By taking

unions, the preimage of any set is open, so f is continuous, and $\text{dom}(f) = \bigcup_{\beta \in \text{dom}(f)} U_{(\beta(0), \dots, \beta(n_\beta-1))}$ is open.

¹⁰ The relationship will be made clear in Chapter 3.

¹¹ This explanation is essentially same as one given by Swan (Swan, *Intuition behind Kleene's “second algebra” $\mathcal{K}_2$*).

¹² This additional value can be added without issue.

¹³ In particular, $\mathbb{B}$ is a Baire space in the sense of the Baire category theorem. In this thesis, we will never use Baire space to mean anything other than $\mathbb{B}$.

¹⁴ For example, we could define $\langle x_0, x_1, \dots, x_n \rangle := 2^{x_0} \cdot 3^{x_1} \dots p_n^{x_n} - 1$, using the prime factorisation theorem.

For the converse, if f is continuous, then the subset $f^{-1}(k)$ is open in $\text{dom}(f)$, which means it is also open in $\mathbb{B}$. In particular, $f^{-1}(k)$ is a union of basic opens, and so if we have $f(\beta) = k$, then there is some finite sequence $x = (x_0, \dots, x_{n-1})$ of minimum length for which $\beta \in U_x \subseteq \text{dom}(f)$. We then set $\alpha(\langle x_0, \dots, x_{n-1} \rangle) = k + 1$ and $\alpha(\langle x_0, \dots, x_{m-1} \rangle) = 0$ for $m < n$. Taking $\alpha(i) = 0$ for all other i , we see that $f = F_\alpha$. $\square$

The following results are useful for characterising realised maps (those equal to F_α for some $\alpha \in \mathbb{B}$).

Theorem 2.2.6. ¹⁵ Every partial continuous map $f : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ can be extended to a map F_α for some $\alpha \in \mathbb{B}$.

Theorem 2.2.7. ¹⁶ A partial function $f : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ is realised if, and only if, f is continuous and its domain is a G_δ -set¹⁷.

We now turn to defining the PCA structure on Baire space.

Definition 2.2.8 (Baire Space (PAS)). The PAS $\mathbb{B}$ consists of the set $\mathbb{N}^{\mathbb{N}}$ with application given by $\alpha \cdot \beta := \lambda n. F_\alpha(n :: \beta)$, which is defined whenever the right side is a total function.

To define a PCA on Baire space, we will need a preliminary lemma about continuous functions of two variables.

Lemma 2.2.9. ¹⁸ Let $f : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$ be a continuous function, then there is $\phi \in \mathbb{B}$ such that for all $\alpha, \beta \in \mathbb{B}$, $\phi \alpha \beta = f(\alpha, \beta)$.

Proof. For $n, k \in \mathbb{N}$, set $U_n^k = \{(\alpha, \beta) \mid f(\alpha, \beta)(n) = k\}$ open. ϕ is defined by case analysis:

$$\begin{aligned} \phi(\langle \rangle) &= 0 \\ \phi(\langle \langle \rangle \rangle :: \tau) &= 1 \\ \phi(\langle \langle n \rangle :: \sigma \rangle :: \tau) &= 1 \quad \text{if there is no } k \text{ such that } U_\tau \times U_\sigma \subseteq U_n^k \\ \phi(\langle \langle n \rangle :: \sigma \rangle :: \tau) &= k + 2 \quad \text{if there is a } k \text{ such that } U_\tau \times U_\sigma \subseteq U_n^k. \end{aligned}$$

Note that if such a k exists, it is necessarily unique. $\square$

Proposition 2.2.10. There are $K, S \in \mathbb{B}$ satisfying the conditions of [Lemma 2.1.8](#).

Proof. For K , apply [Lemma 2.2.9](#) to the first projection $\pi(x, y) = x$.

For S , we first extend our notations $F_\alpha(\beta)$ and $\alpha \cdot \beta$ to names of finite sequences¹⁹. Then we can define a function $s : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$ by $s(\alpha, \beta)(\langle \rangle) = 0$ and $s(\alpha, \beta)(\langle n \rangle :: c) = m + 1$ if and only if there is a $k < \text{len}(c)$ such that:

- $(\beta c)(0), \dots, (\beta c)(k - 1)$ are all defined;
- $F_\alpha(\langle \langle b \rangle \rangle :: c)$ for all prefixes b of $[n, (\beta c)(0), \dots, (\beta c)(i)]$;
- $F_\alpha(\langle \langle n, (\beta c)(0), \dots, (\beta c)(k - 1) \rangle \rangle :: c) = m + 1$; and
- $F_\alpha(\langle \langle n, (\beta c)(0), \dots, (\beta c)(i) \rangle \rangle :: c) = 0$ for all $i < k - 1$.

and $s(\alpha, \beta)(\langle n \rangle :: c) = 0$ if no such k exists. By unrolling definitions, we see that $s(\alpha, \beta)(\langle n \rangle :: c) = m + 1$ for some γ, c such that $\gamma \in U_c$ if and only if $(\alpha \gamma(\beta \gamma))(n) = m$. Consequently, $s(\alpha, \beta)\gamma \simeq \alpha \gamma(\beta \gamma)$ for all γ . Then applying [Lemma 2.2.9](#) to s gives our S . $\square$

¹⁵ Bauer, ‘The realizability approach to computable analysis and topology’, Theorem 1.1.6.

¹⁶ *ibid.*, Theorem 1.1.9.

¹⁷ A G_δ -set is a countable intersection of open sets.

$(n :: \beta)(0) = n$ and $(n :: \beta)(m + 1) = \beta(m)$ for all $m \in \mathbb{N}$.

¹⁸ van Oosten, *Realizability: An Introduction to its Categorical Side*, Lemma 1.4.1.

¹⁹ $F_\alpha(b) = k$ if for some name of an initial segment b' of b , we have $\alpha(b') = k + 1$ and $\alpha(b'') = 0$ for every proper initial segment b'' of b' . $(\alpha b)(n) = k$ if $F_\alpha(\langle n \rangle :: b) = k$.

We can now conclude with the following definition for our PCA structure on Baire space.

Definition 2.2.11 ($\mathcal{K}_2$). *The PAS $\mathbb{B}$ from Definition 2.2.8 is a PCA, which we will call $\mathcal{K}_2$ for Kleene’s second model. The elements S and K are given by Proposition 2.2.10.*

We will also consider the filtered PCA $\mathcal{K}_2^{\text{rec}} \subseteq \mathcal{K}_2$, where we take the subset of $\mathbb{N}^{\mathbb{N}}$ consisting of the computable maps (in the Church-Turing sense).

2.3 Assemblies

HAVING ESTABLISHED (filtered) PCAs as our generic notion of computation, we can turn to defining our categories of computable maps. The main idea of this section is that every mathematical object we care about is represented by at least one “name” and every map is also represented by a function that sends names for inputs to names for outputs.

Definition 2.3.1 (Assembly). *An assembly X over a filtered PCA $\mathbb{A}' \subseteq \mathbb{A}$ is a set $|X|$ (called the carrier) equipped with a relation²⁰ $\Vdash_X \subseteq \mathbb{A} \times |X|$ such that for every $x \in |X|$, there exists an $a \in \mathbb{A}$ with $a \Vdash_X x$.*

If $a \Vdash_X x$ we refer to a as a realiser (or name) for x .

Example 2.3.2 (Real Numbers). *We can give an assembly over $\mathcal{K}_2^{\text{rec}} \subseteq \mathcal{K}_2$ to $\mathbb{R}$ by defining the realisability relation as $(x_n)_{n \in \mathbb{N}} \Vdash_{\mathbb{R}} r$ if and only if $(x_n)_{n \in \mathbb{N}}$ is a fast Cauchy sequence²¹ converging to r .*

Note that in that example we did not require our sequence $(x_n)_{n \in \mathbb{N}}$ to belong to $\mathcal{K}_2^{\text{rec}}$; in fact, we did not seem to use the sub-PCA $\mathbb{A}'$ in the definition of assemblies at all. This was not an oversight; we do make use of it when defining assembly maps.

Definition 2.3.3 (Tracked, Assembly Map). *Fix a sub-PCA $\mathbb{A}' \subseteq \mathbb{A}$ and assemblies X and Y . A function $f : |X| \rightarrow |Y|$ is tracked by an element $t \in \mathbb{A}'$ when, for all $a \in \mathbb{A}$ and $x \in |X|$, if $a \Vdash_X x$, then $t a \Downarrow$ and $t a \Vdash_Y f(x)$. An assembly map is a tracked function between assemblies.*

The importance of using a filtered PCA $\mathbb{A}' \subseteq \mathbb{A}$ is to divide our names into those that we should treat as “computable” ($\mathbb{A}'$) and those which we shouldn’t ($\mathbb{A} - \mathbb{A}'$). In the case of Example 2.3.2, not all real numbers are computable since they not all reals are limits of (Church-Turing) computable functions²³. It is important to stress that while we restrict to computable maps²⁴, those maps may manipulate *uncomputable* inputs. In doing so, they may produce uncomputable outputs.

Proposition 2.3.4. *Assemblies over a filtered PCA $\mathbb{A}' \subseteq \mathbb{A}$ form a category $\text{Asm}(\mathbb{A}, \mathbb{A}')$.*

Proof. The identity map is tracked by the combinator I . If $f : X \rightarrow Y$ is tracked by t and $g : Y \rightarrow Z$ is tracked by s , then $g \circ f$ is tracked by $\langle x \rangle. s \cdot (t \cdot x)$. $\square$

In the case that $\mathbb{A} = \mathbb{A}'$, we will call the category $\text{Asm}(\mathbb{A})$.

²⁰ $\Vdash$ is pronounced “realises”.

²¹ Recall from undergraduate mathematics that a sequence of rationals²² $(x_n)_{n \in \mathbb{N}}$ is fast Cauchy if $|x_m - x_n| < 2^{-\min(n,m)}$, for all $m, n \in \mathbb{N}$. All fast Cauchy sequences converge to some real number and all reals are the limit of some fast Cauchy sequence.

²² We also assume here some computable bijection $\mathbb{N} \rightarrow \mathbb{Q}$.

²³ As there are only countably many such sequences.

²⁴ In the sense that their trackers belong to the computable subset.

Example 2.3.5 (Set). *The category $\mathbf{Set}$ is equivalent to the category $\mathbf{Asm}(\{\star\})$, where $\{\star\}$ is the trivial PCA. In this sense, ordinary mathematics is computable mathematics where everything has trivial computational content.*

If the previous example is privileged by ordinary mathematicians, then those of us in computing favour a different category.

Example 2.3.6 (Church–Turing Computability). *The category $\mathbf{Asm}(\mathcal{K}_1)$ of assemblies over $\mathcal{K}_1$ constitutes the category of functions which are computable in the sense of Church & Turing.*

An important feature of this category is that not all functions $\mathbb{N} \rightarrow \mathbb{N}$ are computable (tracked by an element of $\mathcal{K}_1$).

If you will permit a small bit of poetry²⁵...

Look again at the category $\mathbf{Asm}(\mathcal{K}_1)$. That’s here. That’s home. That’s us.
On it every function you ever wrote, every datum you ever structured, every process that ever was, lived out its runtime.

However, as we’ve already mentioned, Church–Turing computability is not sufficient for computable *analysis*, as it does not have any way of talking about almost all real numbers. Even if we were to accept this restriction²⁶, say by working in the effective topos²⁷ $\mathbf{Eff}$, the analysis that we developed would have some very unfamiliar properties. For example, it is possible to prove that the closed interval (Cantor space $2^{\mathbb{N}}$) is (computably) isomorphic²⁸ to the reals (Baire space $\mathbb{B}$), and that the intermediate value theorem fails^{29,30}. Fortunately, things will be much more familiar if we work with the category of assemblies $\mathbf{Asm}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$.

2.3.1 Structure of $\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$

CATEGORIES OF ASSEMBLIES have a lot of useful structure. This isn’t very surprising because we have already seen that $\mathbf{Set}$ is an example of such a category (Example 2.3.5). More generally, assemblies are sets with computational content, and it transpires that standard constructions on sets can be adapted to track this information. As such, we will omit giving complete proofs of universal properties and simply describe the relevant assemblies.

Theorem 2.3.7. *The category of assemblies $\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$ has finite limits.*

Proof. It suffices to show $\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$ has a terminal object and pullbacks.

For a terminal assembly 1 , we can take any one element set $\{\star\}$ as our carrier with the realisability relation $a \Vdash_1 \star$ for some³¹ $a \in \mathbb{A}'$. The unique map $! : X \rightarrow 1$ is tracked by $\langle x \rangle . a$.

Let X, Y, Z be assemblies and $f : X \rightarrow Z, g : Y \rightarrow Z$ be assembly maps. We can take the pullback P to be $\{(x, y) \mid f(x) = g(y)\}$ with the realisability relation pair $a b \Vdash_P (x, y)$ iff $a \Vdash_X x$ and $b \Vdash_Y y$. The projections from P to X and Y are tracked by fst and snd respectively.

Note that $\langle x \rangle . a, \text{fst}$ and snd are all in the sub-PCA $\mathbb{A}'$. $\square$

²⁵ With apologies to Carl Sagan.

²⁶ Anecdotally, not “believing in” uncomputable reals seems to be not that uncommon amongst computer scientists.

²⁷ A magical land where every definable function is computable.

²⁸ van Oosten, *Realizability: An Introduction to its Categorical Side*, Proposition 3.2.26.

²⁹ *ibid.*, Proposition 3.3.6 (ii).

³⁰ Failure of IVT particularly smarts because finding the root is just a binary search. In reverse mathematics, IVT is provable in the base system $\mathbf{RCA}_0$ (Simpson, *Subsystems of second order arithmetic*, Theorem I.8.3). The intermediate value theorem also fails in the Kleene–Vesley topos because it has a Weihrauch degree above id (Brattka and Gherardi, ‘Effective Choice and Boundedness Principles in Computable Analysis’).

³¹ It is common to allow every $a \in \mathbb{A}$ to realise $\star$, but this is equivalent.

Theorem 2.3.8. *The category of assemblies $\text{Asm}(\mathbb{A}, \mathbb{A}')$ has finite colimits.*

Proof. Again, it will suffice to show that $\text{Asm}(\mathbb{A}, \mathbb{A}')$ has an initial object and pushouts.

The initial assembly 0 is simply by the empty set with the trivial realisability relation. The unique map from 0 to any assembly is trivially tracked.

For pushouts, let X, Y, Z be assemblies and $f : Z \rightarrow X, g : Z \rightarrow Y$ be assembly maps, and we will take the usual pushout construction on sets for the carrier $|P|$. More specifically, take the disjoint union of X and Y quotiented by the finest equivalence $\sim$ such that $f(z) \sim g(z)$ for all $z \in Z$. Then we can define the realisability relation as follows:

$$\begin{array}{ll} \text{left } a \Vdash_P [f(z)]_{\sim} & \text{if } a \Vdash_X f(z) \\ \text{right } b \Vdash_P [g(z)]_{\sim} & \text{if } b \Vdash_Y g(z) \end{array}$$

The coprojections from X and Y to P are tracked by left and right respectively, and both belong to the sub-PCA. $\square$

Theorem 2.3.9. *The category of assemblies $\text{Asm}(\mathbb{A}, \mathbb{A}')$ is (locally) cartesian closed.*

Proof. Let X, Y be assemblies, then we define $|Y^X|$ to be the set of assembly maps $X \rightarrow Y$ that are tracked by elements of $\mathbb{A}$ (not just $\mathbb{A}'$). $|Y^X|$ is given the realisability relation $a \Vdash_{Y^X} f$ if and only if $a \in \mathbb{A}$ tracks f . The evaluation map $\text{ev} : Y^X \times X \rightarrow Y$ is defined by $\text{ev}(f, x) = f(x)$ and is tracked by fst and snd . Hence, $\text{Asm}(\mathbb{A}, \mathbb{A}')$ is cartesian closed.

Let $f : X \rightarrow Y$ be an assembly map. We need to show that the pullback functor $f^* : \text{Asm}(\mathbb{A}, \mathbb{A}')/Y \rightarrow \text{Asm}(\mathbb{A}, \mathbb{A}')/X$ has a right adjoint³². For a map $g : Z \rightarrow X$, define the assembly of “fibrewise maps” by

$$|P| = \{(y, s : f^{-1}(y) \rightarrow Z) \mid s(x) \in |g^{-1}(x)| \text{ for all } x \in |f^{-1}(y)|\}$$

where $|f^{-1}(y)| = \{x \mid f(x) = y\}$ and $a \Vdash_{f^{-1}(y)} x$ iff $a \Vdash_X x$. For our realisability relation on P , we take pair $b \Vdash_P (y, s)$ if $b \Vdash_Y y$ and t tracks s . We obtain a functor $\prod_f : \text{Asm}(\mathbb{A}, \mathbb{A}')/X \rightarrow \text{Asm}(\mathbb{A}, \mathbb{A}')/Y$ by setting $\prod_f(g)$ to be the first projection $\pi_1 : P \rightarrow Y$ (tracked by fst). Given a morphism $k : Z \rightarrow W$ from $g : Z \rightarrow X$ to $h : W \rightarrow X$, the morphism $\prod_f(k) : \prod_f(g) \rightarrow \prod_f(h)$ is given by $(y, s) \mapsto (y, k \circ s)$. It is easy to see that is tracked by an element of $\mathbb{A}'$. $\square$

The left (resp. right) adjoint to pullback allows us to write *dependent pairs* (resp. *functions*).

Definition 2.3.10 (Dependent Pairs, Functions). *Let I be a set (assembly) and $(X_i)_{i \in I}$ be a family of sets (assemblies). A dependent pair $\sum_{i \in I} X_i$ is the set (assembly) of all pairs (i, x) such that $i \in I$ and $x \in X_i$. A dependent function $\prod_{i \in I} X_i$ is a function (assembly map) $f : I \rightarrow \prod_{i \in I} X_i$ such that $f(i) \in X_i$. For $f \in \prod_{i \in I} X_i$, we will sometimes write $f : (i : I) \rightarrow X_i$.*

Aside from being locally cartesian closed, assemblies also have a well-behaved notion of image factorisation.

³² The construction of the right adjoint may seem daunting, but it is really the translation of the same proof for Set (Mac Lane and Moerdijk, *Sheaves in Geometry and Logic*, §I.9, Theorem 3).

Theorem 2.3.11. *The category of assemblies $\text{Asm}(\mathbb{A}, \mathbb{A}')$ is regular.*

Proof. We already saw in [Theorem 2.3.7](#) that $\text{Asm}(\mathbb{A}, \mathbb{A}')$ has finite limits. Coequalisers of kernel pairs exist by [Theorem 2.3.8](#). All it remains to check is that regular epimorphisms are stable under pullback.

Let $f : X \twoheadrightarrow Z$ be a regular epimorphism and $g : Y \rightarrow Z$ an assembly map tracked by t . As f is a regular epi, there exists an $s \in \mathbb{A}'$ such that for all $b \Vdash_Z z$ we have $sb \Vdash_X x$ for some x such that $f(x) = z$. Now, the pullback of f along g is the projection $X \times_Z Y \rightarrow Z$. Suppose $a \Vdash_Y y$, then $ta \Vdash_Z g(y)$ and hence $s(ta) \Vdash_X x$. It follows by [Proposition 2.3.16](#) that the projection is a regular epimorphism. $\square$

Theorem 2.3.12. *There exist regular functors $\Gamma : \text{Asm}(\mathbb{A}, \mathbb{A}') \rightarrow \text{Set}$ and $\nabla : \text{Set} \rightarrow \text{Asm}(\mathbb{A}, \mathbb{A}')$ such that $\Gamma \dashv \nabla$ and $\Gamma \circ \nabla = \text{id}_{\text{Set}}$.*

Proof. Γ is the forgetful functor from assemblies to sets. It evidently preserves finite limits and regular epimorphisms³³.

³³ In set, every epimorphism is regular.

∇ is the cofree functor sending sets X to assemblies X with the maximal realisability relation: $a \Vdash_X x$ for all $a \in \mathbb{A}$, $x \in X$. Any function on sets $f : X \rightarrow Y$ becomes an assembly map tracked by id .

It is easy to check that see $\Gamma \dashv \nabla$ (with the unit tracked by id) and that $\Gamma \circ \nabla = \text{id}_{\text{Set}}$. As a right adjoint, ∇ preserves limits and it follows by [Proposition 2.3.16](#) that ∇ preserves regular epimorphisms. $\square$

Remark 2.3.13. *Functions of the form $f : X \rightarrow \nabla(Y)$ are always tracked (by id) and are, essentially, black holes where computational content loses all meaning.*

2.3.2 Factorisation Systems

WE ALREADY HINTED at the use of factorisation systems when discussing that $\text{Asm}(\mathbb{A}, \mathbb{A}')$ is a regular category, but it is worth spending a little more time on this since factorisation systems will play a role when we come to characterising strong reducibility in [Chapter 6](#). First, we need to characterise those morphisms which are surjective / injective on the carriers.

Proposition 2.3.14. *For an assembly map $f : X \rightarrow Y$, we have*

- *f is an epimorphism iff $f : |X| \rightarrow |Y|$ is surjective, and*
- *f is a monomorphism iff $f : |X| \rightarrow |Y|$ is injective.*

Proof. It is clear that if f is surjective on carriers, then it is an epimorphism. For the other direction, we can use that Γ is a left adjoint and so preserves colimits. Since a morphism f is an epimorphism if only if $\text{id} \circ f = f \circ \text{id}$ is a pushout, Γ preserves epimorphisms.

A similar argument establishes the result for monomorphisms. $\square$

Epimorphisms in Asm have an additional useful property: they're stable under pullback.

Proposition 2.3.15. *Let $f : X \rightarrow Z$, $g : Y \rightarrow Z$ be assembly maps. If f is an epimorphism, then its pullback along g is also an epimorphism.*

Proof. By how we constructed pullbacks in [Theorem 2.3.7](#), we can see that Γ preserves pullbacks. The result then follows from [Proposition 2.3.14](#) and from surjections being stable under pullbacks in $\mathbf{Set}$. $\square$

An important subset of epimorphisms are the *regular* epimorphisms, i.e., those which are the coequaliser of a parallel pair of morphisms.

Proposition 2.3.16. *An assembly map $f : X \rightarrow Y$ is a regular epimorphism iff $|f| : |X| \rightarrow |Y|$ is surjective and there exists $s \in \mathbb{A}'$ such that for all $y \in |Y|$, $b \Vdash_Y y$, we have $s \cdot b \Vdash_X x$ for some $x \in |X|$ with $f(x) = y$.*

Similarly important is the class of regular monomorphisms, i.e., those which are the equaliser of some pair of parallel morphisms.

Proposition 2.3.17. *An assembly map $f : X \rightarrow Y$ is a regular monomorphism iff $|f| : |X| \rightarrow |Y|$ is injective and there exists $i \in \mathbb{A}'$ such that $i \cdot b \Vdash_X x$ for all $x \in |X|$ and $b \Vdash_Y f(x)$.*

Corollary 2.3.18. *If $f : X \rightarrow Y$ is a regular monomorphism, then f corestricts to an isomorphism $X \cong f(X) \subseteq Y$.*

$f(X)$ is a subset of Y with the maximal realisability relation $a \Vdash_{f(X)} y$ if and only if $a \Vdash_Y y$.

Remark 2.3.19. *As a consequence of this corollary, we can feel free to replace regular monomorphisms with inclusions whenever convenient.*

Equipped with all these characterisations, we can show that $\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$ is has two useful factorisation systems.

Theorem 2.3.20. *$\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$ has a stable factorisation system $(\mathcal{L}, \mathcal{R})$ where $\mathcal{L}$ is the class of regular epimorphisms and $\mathcal{R}$ is the class of monomorphisms.*

Proof. Let $f : X \rightarrow Y$ be an assembly map tracked by $t \in \mathbb{A}$. In $\mathbf{Set}$, the carrier function $|f| : |X| \rightarrow |Y|$ factors uniquely through its image $\text{Im}(f)$. We will call this factorisation $|f| = m \circ e$ where $e : |X| \rightarrow \text{Im}(f)$ is surjective and $m : \text{Im}(f) \rightarrow |Y|$ is injective. We will give the set $\text{Im}(f)$ the realisability relation $a \Vdash_{\text{Im}(f)} e(x)$ iff $a \Vdash_X x$. Then e is surjective, tracked by the identity element, and the identity element also realises preimages, so e is a regular epimorphism. m is injective, hence a monomorphism, and is tracked by t .

It is easy to see that the classes of regular epimorphisms and monomorphisms are closed under composition, and that all isomorphisms are epimorphisms and regular monomorphisms.

Now, suppose we have a commuting rectangle $g \circ m \circ e = m' \circ e' \circ f$ where e, e' are epimorphisms and m, m' are regular monos. Since $\mathbf{Set}$ has a surjection-injection factorisation system, we already know there is a unique morphism h such that $h \circ e = e' \circ f$ and $g \circ m = m' \circ h$, so all that remains is to show that h is tracked. Since e is a regular epimorphism, there is an element s that sends realisers $a \Vdash_{\text{dom}(h)} y$ to realisers $s \cdot a \Vdash_{\text{dom}(f)} x$ for some $x \in e^{-1}(y)$. We can then use the trackers for f and e' to obtain a realiser for $h(y)$. $\square$

Remark 2.3.21. *In fact, any regular category has a stable factorisation system given by regular epimorphisms and monomorphisms ³⁴.*

³⁴ Johnstone, *Sketches of an Elephant: A Topos Theory Compendium*, Proposition A1.3.4.

Theorem 2.3.22. $\text{Asm}(\mathbb{A}, \mathbb{A}')$ has a stable factorisation system $(\mathcal{L}, \mathcal{R})$ where $\mathcal{L}$ is the class of (necessarily pullback-stable) epimorphisms and $\mathcal{R}$ is the class of regular monomorphisms.

Proof. Let $f : X \rightarrow Y$ be an assembly map tracked by $t \in \mathbb{A}$. As in [Theorem 2.3.20](#), we use the image-factorisation of $|f| = m \circ e$, where $e : |X| \rightarrow \text{Im}(f)$ is surjective and $m : \text{Im}(f) \rightarrow |Y|$ is injective. However, this time we will give the set $\text{Im}(f)$ the realisability relation $a \Vdash_{\text{Im}(f)} e(x)$ iff $a \Vdash_Y f(x)$. The map e is surjective, hence an epimorphism, and is tracked by t . m is injective, tracked by the identity, and left cancellativity is realised by the identity element, therefore it is a regular monomorphism.

Again, it is not hard to check that $\mathcal{L}$ and $\mathcal{R}$ are closed under composition and contain all isomorphisms. And similarly, we can use the surjective-injective factorisation for Set to guarantee the existence of a unique map h in the commuting diagram below.

$$\begin{array}{ccccc} X & \xrightarrow{e} & Y & \xhookrightarrow{m} & Z \\ f \downarrow & & \downarrow h & & \downarrow g \\ X' & \xrightarrow{e'} & Y' & \xhookrightarrow{m'} & Z' \end{array}$$

The tracker for h is obtained from a straightforward composition of the trackers for g and m and the witness to the left-cancellability of m' . $\square$

2.4 Partitioned Modest Sets

THE CATEGORY OF ASSEMBLIES might have an abundance of structure, but it will be necessary to consider its less well-endowed sibling: the category of partitioned modest sets. This category will be the one we use when characterising Weihrauch reducibility in [Section 3.5](#), and the notion of canonical cover we introduce here ([Definition 2.4.5](#)) will be essential when handling the composition of Weihrauch degrees in [Section 4.2.2](#).

Definition 2.4.1 (Modest Sets). *Given a filtered PCA $(\mathbb{A}, \mathbb{A}')$, the category of modest sets³⁵ $\text{Mod}(\mathbb{A}, \mathbb{A}')$ is the full subcategory of assemblies $(X, \Vdash_X)$ such that any element of $\mathbb{A}$ realises at most one element (i.e., $a \Vdash_X x$ and $a \Vdash_X x'$ imply $x = x'$).*

Restricting ourselves to modest sets does not change the structure very much. The following theorem reiterates the relevant properties of assemblies, all of which follow by checking that the constructions introduced for assemblies preserve modesty.

Theorem 2.4.2. *The category of modest sets $\text{Mod}(\mathbb{A}, \mathbb{A}')$*

- *has finite limits,*
- *has finite colimits,*
- *is (locally) cartesian closed,*
- *is (regular), and*
- *has (regular-epi, mono) and (epi, regular mono) stable factorisation systems.*

³⁵ According to Bauer (Bauer, [Notes on Realizability](#), §3.2.1), this terminology is due to Dana Scott, and refers to the cardinality of a modest set S being no greater than the cardinality of $\mathbb{A}S$.

We will, however, want to work with a further restriction on assemblies: that our modest sets are partitioned. This condition will limit the structure that carries over from modest sets.

Definition 2.4.3 (Partitioned Assemblies). *Call an assembly partitioned when each element has at most one realiser. We write $\mathbf{pMod}(\mathbb{A}, \mathbb{A}')$ and $\mathbf{pAsm}(\mathbb{A}, \mathbb{A}')$ for the respective full subcategories of partitioned modest sets $\mathbf{Mod}(\mathbb{A}, \mathbb{A}')$ and partitioned assemblies $\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$.*

From a category-theoretic perspective, partitioned assemblies are the regular projectives.

Definition 2.4.4 ((Regular) Projective). *An object X in a category $\mathcal{C}$ is called (regular³⁶) projective when it has the left lifting property against regular epimorphisms, that is, whenever we have a regular epimorphism $e : Z \twoheadrightarrow Y$ and a map $f : X \rightarrow Y$, there exists some $g : X \rightarrow Z$ such that $f = e \circ g$.*

$$\begin{array}{ccc} & & Z \\ & \nearrow g & \downarrow e \\ X & \xrightarrow{f} & Y \end{array}$$

For a category $\mathcal{C}$, we call $\mathbf{pC}$ its full subcategory consisting of regular projective objects.

A (regular) projective cover of an object X is a projective object $\tilde{X}$ together with a regular epimorphism $\tilde{X} \twoheadrightarrow X$.

A category $\mathcal{C}$ has enough (regular) projectives if for every object X there exists a regular projective cover $\tilde{X} \twoheadrightarrow X$.

For each assembly, it is possible to define a projective cover.

Definition 2.4.5 (Canonical Cover). *For an assembly $X \in \mathbf{Asm}(\mathbb{A}, \mathbb{A}')$, the canonical cover $\mathbf{Canon}(X)$ is the assembly with carrier $\Vdash_X \subseteq \mathbb{A} \times |X|$, i.e., the set of all pairs (x, e) such that e realises x , equipped with the realisability relation $e \Vdash_{\mathbf{Canon}(X)} (x, e)$.*

There is a regular epimorphism $\pi : \mathbf{Canon}(X) \twoheadrightarrow X$ defined by $\pi(x, e) := x$.

The same construction works for modest sets; in fact, we can simplify things further by simply restricting the carrier to its first component, since modesty means that a name uniquely determines the elements we want to code. So, all in all, we have the following.

Lemma 2.4.6. *The projective objects of modest sets (assemblies) are, up to isomorphism, the partitioned modest sets (assemblies). Furthermore, the categories of modest sets and assemblies always have enough projectives.*

Proof. For modest sets, this is Theorem 3.1 of Bauer³⁷. For assemblies, we can deduce that from the characterization of partitioned assemblies as the projectives of the realisability topos $\mathbf{RT}(\mathbb{A}, \mathbb{A}')$ ³⁸ by noting that coequalizers are preserved by the embedding $\mathbf{Asm}(\mathbb{A}, \mathbb{A}') \rightarrow \mathbf{RT}(\mathbb{A}, \mathbb{A}')$. $\square$

The upshot of the previous lemma is that we can treat projectives as subsets of the assembly.

Convention 2.4.7. *Let $\mathbf{pMod}(\mathbb{A}, \mathbb{A}')$ be a category of projective modest sets. We will identify assemblies X with their set of realisers $\{a \in \mathbb{A} \mid a \Vdash_X x \text{ for some } x\}$ and assume the trivial (identity) realisability relation.*

³⁶ In this thesis we will only be concerned with regular projectives, so projective will always mean regular projective.

³⁷ Bauer, ‘The realizability approach to computable analysis and topology’.

³⁸ van Oosten, *Realizability: An Introduction to its Categorical Side*, Proposition 3.2.7.

2.4.1 Structure of $\mathbf{pMod}(\mathbb{A}, \mathbb{A}')$

ALTHOUGH MUCH of the structure of assemblies does carry over to partitioned modest sets, not all of it does. In particular, we still have finite limits and finite coproducts (rather than finite colimits).

Theorem 2.4.8. *The category of projectives $\mathbf{pMod}(\mathbb{A}, \mathbb{A}')$ (resp. $\mathbf{pAsm}(\mathbb{A}, \mathbb{A}')$) has finite limits and finite coproducts, and these are preserved by the inclusion into $\mathbf{Mod}(\mathbb{A}, \mathbb{A}')$ (resp. $\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$).*

Proof. For finite limits, the proof of [Theorem 2.3.7](#) carries over as is.

For finite coproducts, we can keep the same initial object from [Theorem 2.3.8](#), but we need to give an explicit construction of coproducts, since we will no longer have pushouts. Given assemblies X and Y , the coproduct $X + Y$ is given by the disjoint union of those sets and is realised by left $a \Vdash_{X+Y} \iota_X(x)$ iff $a \Vdash_X x$ and right $b \Vdash_{X+Y} \iota_Y(y)$ iff $b \Vdash_Y y$.

It is clear that the inclusions preserve limits and coproducts. $\square$

The above construction of coproducts also works for $\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$, and with it the following proposition is very easily checked.

Proposition 2.4.9. *The categories $\mathbf{Asm}(\mathbb{A}, \mathbb{A}')$, $\mathbf{pAsm}(\mathbb{A}, \mathbb{A}')$, $\mathbf{Mod}(\mathbb{A}, \mathbb{A}')$ and $\mathbf{pMod}(\mathbb{A}, \mathbb{A}')$ are distributive.*

As every assembly has a projective cover, we can see that all assemblies are quotients of projectives. Since there are assemblies that are not partitioned modest sets, $\mathbf{pMod}(\mathbb{A}, \mathbb{A}')$ does not have all finite colimits. Similar reasoning tells us that $\mathbf{pMod}(\mathbb{A}, \mathbb{A}')$ is not regular. However, the most consequential structure that we lack is cartesian closure. In particular, they are not locally cartesian closed by [Remark A.1.4](#).

Proposition 2.4.10. *The categories $\mathbf{pMod}(\mathcal{K}_2)$, $\mathbf{pAsm}(\mathcal{K}_2)$, $\mathbf{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ and $\mathbf{pAsm}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ are not cartesian closed.*

Proof. Let us focus on $\mathbf{pAsm}(\mathcal{K}_2)$ as proofs along the same lines work for the other categories. We are going to show there is no exponential object for $2^{\mathbb{B}}$; let us call E such an exponential object, which comes with an evaluation map $E \times \mathbb{B} \rightarrow 2$ and a universal property. First, because of the uniqueness part of the universal property, we must have that E is modest. We can consider the partitioned modest set T of well-founded trees with countable branching whose leaves are labeled in 2 . There is a computable map $T \times \mathbb{B} \rightarrow 2$ that assigns each pair (t, p) to the unique Boolean that the path p traverses in t , so, using the exponential structure, we have a fortiori a surjective map $r : T \rightarrow E$ in $\mathbf{pMod}(\mathcal{K}_2)$. Using the evaluation map, we can continuously map a name of E into a name tracking the underlying function $f : \mathbb{B} \rightarrow 2$, and from such a name is up to encoding details an element of T encoding f . Hence we have a section s of r .

Now consider some t which is in the image of s , and replace one of its leaf l by an infinite well-founded tree whose leaves are labelled the same way to obtain t' . We have $s(r(t')) = t$. But since $s \circ r$ is continuous, it must only inspect a finite part of t' to determine the label of l . Then considering t'' obtained from t' by flipping one of the unexamined leaves that did not belong to t leads to a contradiction. $\square$

Remark 2.4.11. *Certain exponentials do exist extensionally in $p\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$: for instance, for every regular projective represented space A and subspace N of $\mathbb{N}$, the exponential A^N can be built. We also have an analogue local statement: call a map $f : X \rightarrow Y$ in $p\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ locally countable if there exists some $s : X \rightarrow \mathbb{N}$ such that $\langle f, s \rangle : X \rightarrow Y \times \mathbb{N}$ is a subspace embedding. For f locally countable, f^* does have a right adjoint Π_f in $p\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$.*

The failure of cartesian closure is not due to a lack of power, as we can build modest sets of names for function spaces, but the lack of quotients means we cannot guarantee that a function has a unique name. More formally, we will have a *weakly* locally cartesian closed category, but to show this we must discuss weak right adjoints.

Definition 2.4.12 (Weak Right Adjoint). ³⁹ A weak right adjoint of a functor $F : \mathcal{C} \rightarrow \mathcal{D}$ consists of

- a map G from objects of $\mathcal{D}$ to objects of $\mathcal{C}$, and
- natural surjections $[-, G(A)]_{\mathcal{C}} \twoheadrightarrow [F(-), A]_{\mathcal{D}}$.

Example 2.4.13. For any small category⁴⁰ $\mathcal{C}$, we can define a preorder on $\text{Obj}(\mathcal{C})$ by $X \leq Y$ iff there is a morphism $f : X \rightarrow Y$. The posetal reflection $\mathcal{C} / \leftrightarrow$ of $\mathcal{C}$ is the quotient of that preorder by $X \equiv Y$ iff $X \leq Y$ and $Y \leq X$.

Assuming the axiom of choice, the quotient $\mathcal{C} \rightarrow \mathcal{C} / \leftrightarrow$ has an obvious weak right adjoint.

Weak right adjoints need not be functors, however, they are still *quasi-functors* in the following sense.

Definition 2.4.14 (Quasi-Functor). ⁴¹ A quasi-functor $F : \mathcal{C} \rightsquigarrow \mathcal{D}$ maps objects of $\mathcal{C}$ to objects of $\mathcal{D}$ and morphisms $X \rightarrow Y$ of $\mathcal{C}$ to non-empty sets of morphisms $F(X) \rightarrow F(Y)$ such that

- $\text{id}_{F(X)} \in F(\text{id}_X)$ for every object X of $\mathcal{C}$
- for composable arrows $f : X \rightarrow Y, g : Y \rightarrow Z$, if $f' \in F(f)$ and $g' \in F(g)$, then $f' \circ g' \in F(f \circ g)$.

Remark 2.4.15. While quasi-functors are weaker than functors, the two notions coincide when the codomain is a preorder. Quasi-functors over $\mathcal{C}$, much like functors, also induce monotonous operations over the poset $\mathcal{C} / \leftrightarrow$.

Categories $\mathcal{C}$ with enough projectives *almost* have a coreflection on their projectives, which we can characterise via a weak right adjoint.

Lemma 2.4.16. If $\mathcal{C}$ has enough regular projectives, the full and faithful embedding induced by the inclusion $p\mathcal{C} \subseteq \mathcal{C}$ has a weak right adjoint $X \mapsto \widetilde{X}$ with $\widetilde{\widetilde{X}} = X$ for any object X of $p\mathcal{C}$.

Proof. For any object X , fix a regular projective cover $\varepsilon_X : \widetilde{X} \twoheadrightarrow X$, taking the identity when X was already regular projective. This gives a counit for our weak adjoint, whose morphism part is then the postcomposition

$$\varepsilon_X \circ - : [A, \widetilde{X}]_{p\mathcal{C}} \twoheadrightarrow [A, X]_{\mathcal{C}}$$

It is obviously natural in A . It is easily seen to be surjective since A is regular projective and ε_X is a regular epimorphism. $\square$

³⁹ Kainen, ‘Weak adjoint functors’, §1.

⁴⁰ A category is small when both its classes of objects and morphisms are sets.

⁴¹ *ibid.*, §0.

A weakly locally cartesian closed category is similar to a locally cartesian closed category, but where the usual requirement that pullback functors have a right adjoint is loosened using [Definition 2.4.12](#). That $\text{pMod}(\mathbb{A}, \mathbb{A}')$ is a weak LCCC follows from [Theorem 2.3.9](#).

Definition 2.4.17 (Weak LCCC). ⁴² *A category is weakly locally cartesian closed⁴³ when all pullback functors $f^*(-)$ have weak right adjoints.*

⁴² Carboni and Rosolini, ‘Locally cartesian closed exact completions’, Remark 3.2.

⁴³ Usually shortened to WLCCC or weak LCCC.

⁴⁴ *ibid.*, Theorem 3.3.

Theorem 2.4.18. ⁴⁴ *If $\mathcal{C}$ has enough projectives, then $\text{p}\mathcal{C}$ is weakly locally cartesian closed if and only if $\mathcal{C}$ is locally cartesian closed.*

Corollary 2.4.19. $\text{pMod}(\mathbb{A}, \mathbb{A}')$ is weakly locally cartesian closed.

2.4.2 Factorisation Systems

UNLIKE WITH ASSEMBLIES, it is no longer the case that the epimorphisms are automatically pullback-stable, and we need to modify [Proposition 2.3.14](#) accordingly.

Proposition 2.4.20. *A map $f : X \rightarrow Y$ in $\text{pMod}(\mathbb{A}, \mathbb{A}')$ is surjective if and only if it is a pullback-stable epimorphism.*

Proof. It is still the case that if f is surjective then it is a pullback-stable epimorphism. For the other direction, let $y \in Y$ and take a pullback of f along a constantly y map $!_y : 1 \rightarrow Y$. Then $\text{dom}(!_y^* f)$ cannot be empty⁴⁵, otherwise $!_y^*$ would equalise the two maps $\text{true}, \text{false} : 1 \rightarrow 2$. It follows that y is in the image of f . $\square$

⁴⁵ Hence, isomorphic to 0.

There are also epimorphisms in $\text{pMod}(\mathbb{A}, \mathbb{A}')$ which are not pullback-stable/surjective.

Counterexample 2.4.21. *We will work in $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ and show there is a non-surjective epimorphism $\mathbb{N} \rightarrow 2^{\mathbb{N}}$. Both of these can be viewed as projective modest sets, e.g., represent a natural n by the constant function $\lambda x. n$, and use the trivial reflexive realisability relation on $2^{\mathbb{N}} \subseteq \mathbb{B}$. Using any computable isomorphism $l : \mathbb{N} \rightarrow \text{List}(\mathbb{N})$, we can define a map $i : \mathbb{N} \rightarrow 2^{\mathbb{N}}$ sending n to the sequence $b_0 b_1 \dots$, where $b_i = 1$ if $i \in l(n)$ and $b_i = 0$ otherwise. The function i cannot be surjective for cardinality reasons, but it is an epimorphism.*

Let $f, g : 2^{\mathbb{N}} \rightarrow 2$ be continuous functions. On any input $\alpha \in 2^{\mathbb{N}}$, both f and g will output their answer after reading only a finite prefix $\alpha(0) \dots \alpha(n)$ of the input, and will therefore give the same answer on the input $\alpha' = \alpha(0) \dots \alpha(n) 00^{\mathbb{N}}$. However, α' is clearly in the image of i . Hence, if $f \circ i = g \circ i$, then $f = g$.

More generally, any assembly in $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ is isomorphic to a represented subspace⁴⁶ of $2^{\mathbb{N}}$, so we only need to check that $f \circ i = g \circ i$ implies $f = g$ for $f, g : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$. But if $f \circ i = g \circ i$, then $(\lambda \alpha. \alpha(n)) \circ f \circ i = (\lambda \alpha. \alpha(n)) \circ g \circ i$ for every $n \in \mathbb{N}$, meaning the functions f, g are equal.

⁴⁶ We already know this is true for $\mathbb{B}$ using [Lemma 2.4.6](#), and there is a computable isomorphism $\mathbb{B} \rightarrow 2^{\mathbb{N}}$ sending the sequence $n_0, n_1, \dots$ to $0^{n_0} 10^{n_1} 1 \dots$.

As such, the qualifier “pullback-stable” in [Theorem 2.3.22](#) is essential for giving a factorisation system for $\text{pMod}(\mathbb{A}, \mathbb{A}')$.

Theorem 2.4.22. $\text{pMod}(\mathbb{A}, \mathbb{A}')$ has a factorisation system $(\mathcal{L}, \mathcal{R})$ where $\mathcal{L}$ is the class of pullback-stable epimorphisms and $\mathcal{R}$ is the class of regular monomorphisms.

Proof. The proof of [Theorem 2.3.22](#) remains good. The only additional thing that needs checked is that $\text{Im}(f)$ is a partitioned modest set, but this is true as $\text{cod}(f)$ is a partitioned modest set. $\square$

3 Weihrauch Problems as Containers

WE BEGIN THIS CHAPTER by describing type-2 computability, which provides a framework for computing on $\mathbb{B}$ and discuss how it relates to Kleene’s second model introduced in [Section 2.2](#). We then give an introduction to Weihrauch reducibility, which is the main topic of this thesis. Next, we lead the reader through a series of alternative definitions for Weihrauch reducibility, culminating in a description of reducibility that can be interpreted in any category with pullbacks. Containers are then formally introduced, before we give characterisations of reducibility and continuous reducibility in terms of answerable containers over the category of projective modest sets.

3.1 Type-2 Computability

THE TYPE TWO THEORY OF EFFECTIVITY (TTE) was introduced by Weihrauch as a framework for computable analysis. It achieves this by studying partial functions on $\mathbb{B}$, which is a space with cardinality of the continuum. This section will be somewhat brief, aiming primarily to link TTE to assemblies on $\mathcal{K}_2$ that we have already studied¹. Interested readers are recommended to consult Weihrauch’s book² for more details.

In TTE, we don’t need to change the basic function of Turing machines, except to require an infinite input tape and a *one way* infinite output tape. The alphabet used on each tape will always be $\mathbb{N}$ ³.

Definition 3.1.1 (Type-2 Machine, Computable).⁴ *A type-2 machine is a Turing machine with an infinite input tape, finitely many working tapes, and a single output tape whose head can only be moved to the right and each cell can only be written to once.*

A function $f : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ is (type-2) computable if and only if there is a machine M such that, for each $\alpha \in \text{dom}(f)$, M runs forever on input α and writes $f(\alpha)$ to the output tape, and M diverges⁵ for $\alpha \notin \text{dom}(f)$.

Remark 3.1.2. *It does not change the computational power if require the input tape to be read only, or if we allow finitely many input tapes. By comparison, these restrictions on the output tape are essential. Extending machines to allow going back and modifying earlier outputs, referred to as “mind changes”, is strictly more powerful. For example,⁶ multiplying an infinite decimal fraction by 3 is not computable by type-2 machines, but can be computed with a two-way output tape.*

If $f(\alpha) = \beta$, then every cell β_n of the output tape is written to after some finite number of steps, and hence after only reading a finite prefix of

¹ This reflects the authors bias towards assemblies, which he studied first, and his conviction that realisability relations are more readable than representations.

² Weihrauch, *Computable Analysis: An Introduction*.

³ The choice of $\mathbb{N}$ for our alphabet rather than $\{0,1\}$ is convenient, but inessential; we could always code $n \in \mathbb{N}$ by 0^n1 .

⁴ *ibid.*, Definitions 2.1.1 and 2.1.2.

⁵ A machine M can diverge in two ways: it can terminate, or it can run forever but only write to a finite number of cells of the output tape.

⁶ Note that this is because infinite decimals are a poor representation of real numbers, and multiplication is computable with a better representation, such as the one in [Example 3.1.5](#).

the input tape. In particular, this means that f is continuous with respect to the topology on Baire space (Definition 2.2.1). Moreover, this definition of type-2 machine computes the same functions as the realised partial functions on $\mathbb{B}$ discussed in Section 2.2.

Proposition 3.1.3. *A partial function $f : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ is equal to F_α for some computable ⁷ $\alpha \in \mathbb{B}$ if and only if there is some type-2 machine that computes it.*

Proof. It is immediate from the definition of F_α (Definition 2.2.3) that if α is type-1 computable then F_α is type-2 computable.

For the other direction, we already know that if f is type-2 computable, then it is continuous. For any finite prefix of the output tape $\beta_0, \dots, \beta_n$, any preimage $\alpha \in f^{-1}(U_{(\beta_0, \dots, \beta_n)})$ will output that after reading only some finite prefix $\alpha_0, \dots, \alpha_m$ of α , hence f will output $\beta_0, \dots, \beta_n$ on any element of the basic open $U_{(\alpha_0, \dots, \alpha_m)}$. By taking unions over all prefixes of length n , and all such basic opens, then $O_n = \{\alpha \mid f(\alpha) \text{ writes to the first } n \text{ cells}\}$ is an open set. It follows that we have $\text{dom}(f) = \bigcap_{n \in \mathbb{N}} O_n$, which is G_δ by definition. Hence, there is an α such that $f = F_\alpha$ by Theorem 2.2.7.

In fact, a suitable α can be defined by $\alpha(\langle n \rangle :: \beta) = k + 1$ for $n \in \mathbb{N}$, $\beta \in \mathbb{N}^*$, if $f(\beta)$ writes k to the n -th cell in $\text{len}(\beta)$ steps, and $\alpha(x) = 0$ otherwise. This is clearly computable. $\square$

While $\mathbb{B}$ is a sufficient setting for type-2 computability, it is convenient to be able to transfer computability to other kinds of structures. In type-1 computability, this role is performed by numberings; for type-2, this is achieved by representations. The key idea is that each object x of a set X , will have one or more “names” in $\mathbb{B}$, and computable functions $X \rightarrow Y$ on sets will be those that have a corresponding partial function $\mathbb{B} \rightarrow \mathbb{B}$ on names. This section will not explore representations thoroughly and the reader is referred to the work of Pauly ⁸ for a more detailed introduction.

Definition 3.1.4 (Represented Spaces). *A represented space (X, δ_X) is a set X with a surjective partial function $\delta_X : \subseteq \mathbb{B} \rightarrow X$.*

When $\delta_X(p) = x$, we call p a name for x .

Example 3.1.5.

- $(\mathbb{B}, \text{id}_{\mathbb{B}})$ is a represented space.
- The natural numbers $\mathbb{N}$ can be represented using $\delta_X(\lambda x. n) := x_n$.
- There exists a representation for the real numbers $\mathbb{R}$ sending fast Cauchy sequences to their limit ⁹ (c.f. Example 2.3.2). An equivalent representation sends nested sequences of closed intervals $[a_0, b_0] \supseteq [a_1, b_1] \supseteq \dots [a_n, b_n] \supseteq \dots$ ($a_n, b_n \in \mathbb{Q}$, $|a_n - b_n| \leq 2^{-n}$) to their unique common point $\bigcap_{n \in \mathbb{N}} [a_n, b_n]$.
- The previous example can be generalised to give a standard representation for Polish spaces¹⁰, by replacing the use of rationals by members of the canonical dense subset.
- Non-polish spaces can also be represented, e.g. the Sierpiński space¹¹ S can be given the representation:

$$\delta_S(p) := \begin{cases} \perp & \text{if } p = 0^{\mathbb{N}}, \\ \top & \text{otherwise.} \end{cases}$$

⁷ Computable in the sense of Church-Turing, i.e., type-1 computable.

⁸ Pauly, ‘On the topological aspects of the theory of represented spaces’.

⁹ Weihrauch, *Computable Analysis: An Introduction*, Definition 4.1.5.

¹⁰ A Polish space is one that is homeomorphic to a complete metric space with a countable dense subset.

¹¹ The topological space with two elements $\{\perp, \top\}$ and three open sets $\{\emptyset, \{\top\}, \{\perp, \top\}\}$.

The notion of morphism between represented spaces is that of a partial multi-valued function that is realised, i.e., where the operation on spaces is mirrored by a corresponding function on names.

Definition 3.1.6 (Realiser). Let $(X, \delta_X), (Y, \delta_Y)$ be represented spaces, $f : \subseteq X \rightrightarrows Y$ a partial multi-valued function, and $F : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ a partial function. We say F is a realiser for f , denoted $F \vdash f$, if $\delta_Y(F(p)) \in f(\delta_X(p))$ for all $p \in \text{dom}(f \circ \delta_X)$. In other words, F maps names for x to names for $f(x)$.

A partial multi-valued function f is called computable (continuous) if it has a computable (continuous) realiser.

Remark 3.1.7. The reason for the terminology “partial multi-valued function” rather than “relation” comes from them having a different notion of composition than relations. Two relations are composed by $x(R \circ S)y$ if and only if there exists a z such that xRz and zRy . Two partial multi-valued functions are composed by the same formula, however, the domain is restricted:

$$\text{dom}(g \circ f) := \{x \in \text{dom}(f) \mid f(x) \subseteq \text{dom}(g)\}.$$

A useful reference on the differences between the category of multi-valued functions Mult and the category of relations Rel was written by Pauly¹².

By placing restrictions on partial multi-valued functions and realisers, we can form different categories of represented spaces, of which two are of central importance to this thesis.

Definition 3.1.8 (ReprSp, ctsReprSp). The category ReprSp has represented spaces as objects and computable functions¹³ as maps. ctsReprSp is the category of represented spaces and continuous functions¹⁴.

Represented spaces are closed under various operations, including subspaces, products, coproducts and exponentials¹⁵. Indeed, that represented spaces are locally cartesian closed will follow from [Proposition 3.1.10](#) and [Theorem 2.4.2](#). Represented spaces induce a topological space by using the quotient topology of the representation¹⁶.

Remark 3.1.9. Cartesian closure of ReprSp means that there exists a standard representation for function spaces like $C[0, 1]$. In fact, there are multiple interesting equivalent representations, from which it can be shown that $C(X, Y)$ has the expected compact-open topology, and, in particular, $C[0, 1]$ is a metric space with the usual uniform metric¹⁷.

Now, we can turn to showing that ReprSp and ctsReprSp are categories of modest sets.

Proposition 3.1.10. ReprSp is isomorphic to $\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ and ctsReprSp is isomorphic to $\text{Mod}(\mathcal{K}_2)$.

Proof. Given a modest set $(X, \Vdash_X)$, we can define a representation for X by $\delta_X(a) := x$ iff $a \Vdash_X x$. The functors are identities on morphisms, as functions being tracked and having a realiser are the same thing. Note we use [Proposition 3.1.3](#) and [Proposition 2.2.5](#) to guarantee our realisers belong to the appropriate PCA (and vice versa). $\square$

¹² Pauly, ‘Many-one reductions and the category of multivalued functions’.

¹³ Functions with computable realisers.

¹⁴ Functions with continuous realisers.

¹⁵ Pauly, ‘On the topological aspects of the theory of represented spaces’, Propositions 1 and 3.

¹⁶ *ibid.*, §4.

¹⁷ Weihrauch, *Computable Analysis: An Introduction*, §6.1, especially Lemmas 6.1.5, 6.1.7 and 6.1.10.

A very useful subcategory of represented spaces are those for which the representation is injective. It is easily seen that these are the (regular) projective modest sets, and so we will refer to such spaces as projective represented spaces.

Definition 3.1.11 (rpReprSp). *The category rpReprSp is the full subcategory of ReprSp where the objects are projective represented spaces.*

Corollary 3.1.12. *rpReprSp is equivalent to $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$.*

Proof. This follows by [Lemma 2.4.6](#) and [Proposition 3.1.10](#) □

The categories of assemblies we have covered in this chapter and their counterparts in computable analysis are recorded in [Figure 3.1](#).

Realisability	Computable Analysis
Assemblies (Asm)	Multi-Represented Spaces
Modest Sets (Mod)	Represented Spaces
Partitioned Modest Sets (pMod)	Represented Subspaces of $\mathbb{B}$ (Projective Represented Spaces)

Finally, we record a convention that is occasionally useful.

Convention 3.1.13. *When working with projective represented spaces, we will treat them as subspaces of $\mathbb{B}$ with the identity map as their representation.*

3.2 Introduction to Weihrauch Reducibility

IN COMPUTABILITY THEORY, we are not just interested in the basic questions of whether a given problem is definable or type-2 computable, but also in determining the relative difficulty of problems. This section gives a brief introduction to one method of classifying problems in computable analysis, namely Weihrauch reducibility. It covers only the basic definitions and examples of reductions, leaving the ample algebraic structure to other chapters. For additional information on the Weihrauch degrees, we refer the reader to the detailed survey of the literature by Brattka et al.¹⁸, and to Brattka’s history of the development of Weihrauch reducibility¹⁹.

The basic notion of problem in computable analysis is that of a partial multi-valued function, rather than a decision procedure. Aside from their convenience in representing $\forall\exists$ statements, multi-valued maps are considered essential as, for natural problems like root-finding, it is possible that the function version of a problem is not computable, but the multi-valued function is computable²⁰.

Definition 3.2.1 (Problem). *A Problem is a partial multi-valued function $f : \subseteq X \rightrightarrows Y$ on represented spaces.*

If $x \in \text{dom}(f)$ we sometimes refer to it as an f -question and an element $y \in f(x)$ as an f -answer for x .

Convention 3.2.2. *We will use sans-serif for the names of problems.*

While any function on $\mathbb{B}$, such as the identity id , can be viewed as a (single valued) problem, many of the most interesting problems are *non-computable* relations. We will explore a few examples of these.

Figure 3.1: Categories used in realizability and their counterpart, once they are specialised to the filtered PCA $(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$, in computable analysis.

¹⁸ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’.

¹⁹ Brattka, ‘Weihrauch Complexity and the Hagen School of Computable Analysis’.

²⁰ Weihrauch, *Computable Analysis: An Introduction*, Example 6.3.1.

The limited principle of omniscience (LPO) decides if a sequence is the zero sequence; the lesser limited principle of omniscience (LLPO) is weaker still, telling us only if the sequence is zero on odd or even indices. These were originally studied in type-2 computability by Weihrauch ²¹.

²¹ Weihrauch, *The TTE-interpretation of three hierarchies of omniscience principles*.

Example 3.2.3 (LPO, LLPO).

$$\text{LPO} : \mathbb{B} \rightarrow \mathbb{N}$$

$$\text{LPO}(p) = \begin{cases} 0 & \text{if } p(n) = 0 \text{ for all } n \in \mathbb{N} \\ 1 & \text{if } p(n) \neq 0 \text{ for some } n \in \mathbb{N} \end{cases}$$

$$\text{LLPO} : \subseteq \mathbb{B} \Rightarrow \mathbb{N}$$

$$\text{dom}(\text{LLPO}) := \{p \in \mathbb{B} \mid p(k) \neq 0 \text{ for at most one } k\}$$

$$\text{LLPO}(p) \ni \begin{cases} 0 & \text{if } p(2n) = 0 \text{ for all } n \in \mathbb{N} \\ 1 & \text{if } p(2n+1) = 0 \text{ for all } n \in \mathbb{N} \end{cases}$$

Proposition 3.2.4. *Neither LPO nor LLPO is type-2 computable.*

Proof. Suppose LPO was type-2 computable, then on the zero sequence it would return 0 after only inspecting a finite prefix 0^n of the argument. As such, it would return 0 for non-zero sequences such as $0^n 10^\omega$. A similar argument establishes the result for LLPO. $\square$

Two of the author's personal favourites involve finding infinite paths in infinite (binary) trees. A tree is a set of words closed under prefixes. We omit the details of coding up such trees as represented spaces.

Definition 3.2.5 (KL, WKL).

- König's lemma (KL) is the problem whose questions are given a finitely branching trees $t \subseteq 2^*$, and t is answered by an infinite path $p \in \mathbb{N}^{\mathbb{N}}$ in t .
- Weak König's lemma (WKL) is the problem whose questions are infinite binary trees $t \subseteq 2^*$, and where such a question is answered by an infinite path $p \in 2^{\mathbb{N}}$ in t .

A problem that takes advantage of representations in an essential way²² is provided by the intermediate value theorem from calculus.

Definition 3.2.6 (Intermediate Value Theorem). ²³

$$\text{IVT} : \subseteq \mathcal{C}[0, 1] \Rightarrow [0, 1]$$

$$\text{dom}(\text{IVT}) := \{f \in \mathcal{C}[0, 1] \mid f(0) \cdot f(1) < 0\}$$

$$\text{IVT}(f) := \{x \in [0, 1] \mid f(x) = 0\}$$

²² In the sense that $\mathcal{C}[0, 1]$ is not a subspace of $\mathbb{B}$: $\mathbb{B}$ and all its subspaces are totally disconnected, but $\mathcal{C}[0, 1]$ (with the usual uniform metric) is not.

²³ Brattka and Gherardi, 'Effective Choice and Boundedness Principles in Computable Analysis', Definition 6.1.

Problems have a natural partial ordering according to whether a problem f contains another g .

Definition 3.2.7 (Solves). *Let $f, g : \subseteq X \Rightarrow Y$ be problems. Then $f \sqsubseteq g$ if and only if $\text{dom}(g) \subseteq \text{dom}(f)$ and $f(x) \subseteq g(x)$ for all $x \in \text{dom}(g)$. In such a case, we say f solves g .*

As already stated, we are interested in determining the relative difficulty of problems, which we can rephrase as “can a problem A be solved (computed) given access to an oracle for B ?”. It is not immediately apparent how one can do this when our basic objects (sequences of natural numbers) are already infinite. The key insight of Weihrauch reducibility is that, when we restrict to only allowing a single oracle call, our solution should factor into two type-2 computable maps: computing the argument for the oracle, and computing the solution to the original input, given access to both the input itself and the output of the oracle on that input.

Definition 3.2.8 (Weihrauch Reducible). *Let f and g be problems.*

- f is Weihrauch reducible to g , written $f \leq_W g$, if there exist computable $\Phi, \Psi : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ such that, for all realisers $G \vdash g$, $\Psi \langle \text{id}, G \circ \Phi \rangle \vdash f$ (see Figure 3.2a).
- f is continuously reducible to g if there exist continuous Φ, Ψ satisfying such that $\Psi \langle \text{id}, G \circ \Phi \rangle$ continuously realises f for all continuous realisers $G \vdash g$.
- f is strongly²⁴ Weihrauch reducible to g , written $f \leq_{sW} g$, if there exist computable $\Phi, \Psi : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ such that, $\Psi \circ G \circ \Phi \vdash f$ for all realisers $G \vdash g$ (see Figure 3.2b).

In all three cases we will refer to Φ as the forward map and Ψ as the backward map.

²⁴ Strong reducibility will not play a major role in this thesis until Chapter 6.

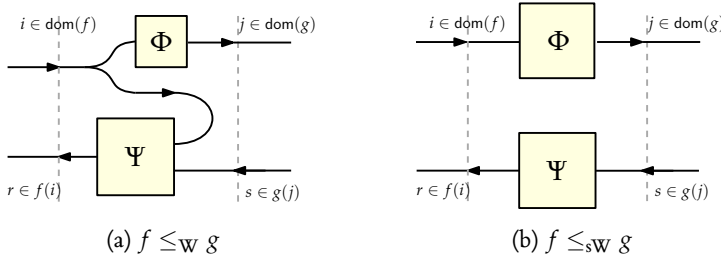


Figure 3.2: Diagrammatic representation of Weihrauch reductions.

Example 3.2.9. Any type-2 computable function $f : \subseteq X \rightrightarrows Y$ is reducible to the identity problem $\text{id} : \subseteq \mathbb{B} \rightrightarrows \mathbb{B}$, and vice versa if $\text{dom}(f) \neq \emptyset$. For $f \leq_W \text{id}$, there is a strong reduction with f as the forward map and id as the backward map. For $\text{id} \leq_W f$, take any map to be the forward map and the first projection $\pi_1 : \mathbb{B} \times Y \rightarrow \mathbb{B}$ as the backward map.

Example 3.2.10. We have that $\text{LLPO} \leq_W \text{LPO}$. For the forward map Φ we take the identity. For the backward map $\Psi(p, b)$, we have as arguments the original sequence $p = (p_n)_{n \in \mathbb{N}}$, as well as an additional bit b of information telling us whether or not p is the zero sequence. If $b = 0$, then $\{0, 1\} = \text{LLPO}(p)$, so we can return either value, say $\Psi(p, 0) = 0$. If $b = 1$, then we know some p_n is non-zero, so we simply go through the sequence in order until we find it. If that n is even we return 1, otherwise return 0.

An alternative proof, which shows that $\text{LLPO} \leq_{sW} \text{LPO}$, is as follows. We take the backwards map Ψ to be the identity. For the forward map Φ , we go through the input $(p_n)_{n \in \mathbb{N}}$ in order, copying p_n to the output sequence if it is 0. If we ever have $p_n = 1$, then we output $0^{\mathbb{N}}$ if n is odd and $10^{\mathbb{N}}$ if n is even.

Counterexample 3.2.11. It is not the case that $\text{LPO} \leq_W \text{LLPO}$. Suppose we had such a reduction (Φ, Ψ) and let l be a realiser for LLPO . Since we are using the identity representation on $\mathbb{B}$, there are precisely two realisers l_b of LLPO , where $l_b(0^\omega) = b \in \{0, 1\}$.

If $\Phi(0^\omega) = 0^n 10^\omega$, then it outputs the 1 after inspecting only $m \in \mathbb{N}$ many characters. Consequently, $\Phi(0^m p) = 0^n 10^\omega$ for any binary sequence p . It follows that, for any sufficiently large $m' \geq m$, $0 = \text{LPO}(0^\omega) = \Psi(0^\omega, l(\Phi(0^\omega))) = \Psi(0^{m'} 10^\omega, l(\Phi(0^{m'} 10^\omega))) = \text{LPO}(0^{m'} 10^\omega) = 1$. It follows by contradiction that $\Phi(0^\omega) = 0^\omega$.

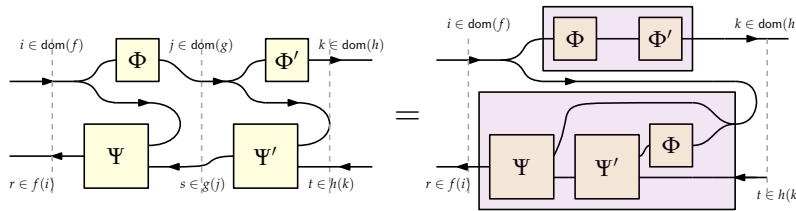
Similarly, $0 = \text{LPO}(0^\omega) = \Psi(0^\omega, l_b(\Phi(0^\omega)))$ will answer after inspecting the bit b and a finite prefix 0^{n_b} of the input. This means that, on any non-zero sequence $0^{\max(n_0, n_1)} p$, LPO must return 0, which contradicts that we have a valid reduction of LPO to LLPO .

Example 3.2.12. There is a reduction $\text{LPO} \leq_W \text{KL}$ defined as follows: the forward map takes a sequence $p \in 2^\mathbb{N}$ to the least tree t such that $t(0^n) = 1$ if and only if 0^n is a prefix of p , $t((k+1)0^n) = 1$ for some $k, n \in \mathbb{N}$ if and only if $p_k = 1$ and $p_{k'} = 0$ for every $k' < k$. Then the backwards functional ignores its first argument $p \in 2^\mathbb{N}$ and returns 1 if and only if we have $q_0 \neq 0$ where $q \in \mathbb{B}$ is the path through t .

Conversely, there is no reduction $\text{KL} \leq_W \text{LPO}$ for continuity reasons.

Example 3.2.13. The degree of the intermediate value theorem was classified by Brattka & Gherardi²⁵, where it was shown to be equivalent to choice on the interval (Theorem 6.2). In particular, $\text{LLPO} \leq_W \text{IVT}$, but IVT and LPO are incomparable (Corollary 4.12).

It is easily checked that $\leq_W$ and $\leq_{sW}$ form preorders (see Figure 3.3 for composition) and that $f \leq_{sW} g$ implies $f \leq_W g$. The converse is not true.



²⁵ Brattka and Gherardi, ‘Effective Choice and Boundedness Principles in Computable Analysis’.

Figure 3.3: Composition of Weihrauch reductions $f \leq_W g$ and $g \leq_W h$.

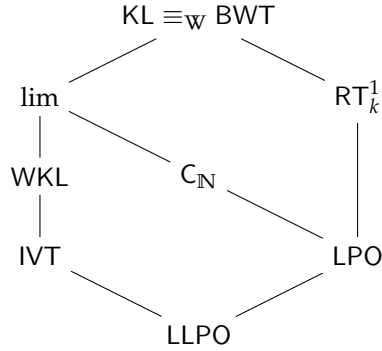
Counterexample 3.2.14. Let id_2 be the identity function on the Booleans 2 and id_1 be the identity on a space 1 with a single point. Then $\text{id}_2 \leq_W \text{id}_1$ by taking the forward map to be the unique such function, and the backwards map to be the projection $\langle n, \alpha \rangle \mapsto n$. This reduction cannot be strong because any candidate for the backwards map must be constant.

Definition 3.2.15 (Degrees). We can induce a partial order on $\leq_W$ by taking $p \equiv_W q$ iff $p \leq_W q$ and $q \leq_W p$. We refer to the equivalence classes as the Weihrauch degrees.

In a similar way, the equivalence classes induced on $\leq_{sW}$ are referred to as the strong Weihrauch degrees

Much of the work in Weihrauch reducibility goes into determining where various problems fit into the poset of Weihrauch degrees by proving

the existence or, or lack of, reductions between degrees. In particular, it is typical to compare new problems against a suite of well-studied (classes of) problems such as LPO, WKL, KL, RT²⁶, and C_X ²⁷. To take a recent example, Brattka and Smischlaew²⁸ show that solving initial value problems is equivalent to weak König's lemma (WKL) in the strong Weihrauch degrees. A small fragment of the Weihrauch degrees containing all these problems appears in Figure 3.4; we refer readers to the work of Brattka²⁹ for proofs of the existence of the relevant reductions and separations.



²⁶ RT stands in for any version of Ramsey's theorem studied in the Weihrauch reducibility literature.

²⁷ C_X is the problem of choosing an element of a set X .

²⁸ Brattka and Smischlaew, 'Computability of Initial Value Problems'.

²⁹ Brattka and Gherardi, 'Effective Choice and Boundedness Principles in Computable Analysis'; Brattka, Gherardi and Marcone, 'The Bolzano–Weierstrass Theorem is the jump of Weak König's Lemma'; Brattka and Rakotoniaina, 'On the Uniform Computational Content of Ramsey's Theorem'.

Figure 3.4: Small fragment of the Weihrauch lattice.

Remark 3.2.16. *It is known that the strong degrees are strictly finer than the Weihrauch degrees. For example, the Baire category theorem BCT is Weihrauch equivalent to the problem of finding the limit of eventually constant sequences $\lim_\Delta$, but they are not strongly equivalent*³⁰.

3.3 Alternative Definitions of Weihrauch Reducibility

IN THIS SECTION, we will chart a course from the usual definition of Weihrauch reducibility (Definition 3.2.8) to an isomorphic definition that is more amenable to category-theoretic treatment.

The first change we will make comes from noticing that we don't actually need to talk about represented spaces at all. Represented spaces make arguments about some spaces, as IVT, more intuitive, but we can work directly on $\mathbb{B}$ and only treat subspaces of that. This following definition of Weihrauch reducibility is already widely used in the literature, e.g., by Dorais et al³¹.

Definition 3.3.1 (Problem (2)). *A Problem is a partial multi-valued function on $\mathbb{B}$.*

Definition 3.3.2 (Weihrauch Reducible (2)). *Let f and g be problems in the sense of Definition 3.3.1. Then f is Weihrauch reducible to g , written $f \leq_W^l g$, if there exist computable $\Phi, \Psi : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ such that $\Psi \langle \text{id}, g \circ \Phi \rangle \subseteq f$.*

To show that $\leq_W$ is isomorphic to $\leq_W^l$, it is convenient to use the following definition and lemma.

Definition 3.3.3 (Realiser Part of a Problem, f^r). *Let $f : \subseteq X \rightrightarrows Y$ be a problem. The realiser part of f , $f^r : \subseteq \mathbb{B} \rightrightarrows \mathbb{B}$, is defined by*

$$\beta \in f^r(\alpha) \iff \delta_Y(\beta) \in f(\delta_X(\alpha)).$$

³⁰ Brattka, Gherardi and Marcone, 'The Bolzano–Weierstrass Theorem is the jump of Weak König's Lemma', Fact 3.6 and the discussion following Fact 3.7.

³¹ Dorais, Dzhafarov, Hirst, Mileti and Shafer, 'On uniform relationships between combinatorial problems'.

Lemma 3.3.4. *If $f : \subseteq X \rightrightarrows Y$ is a problem, then $f =_{sW} f^r$.*

Proof. A partial function is a realiser for f if and only if it is a realiser for f^r . From this, we can simply take our forward and backward maps to be the identity. $\square$

Corollary 3.3.5. $f \leq_W g \iff f^r \leq_W g^r \iff f^r \leq'_W g^r$.

Proof. This is a simple consequence of the definitions and the axiom of choice. $\square$

For our next definition, the key idea is to treat problems as families of non-empty sets, indexed by their domains.

Definition 3.3.6 (Problem (3)). *A Problem is a family $(F_i)_{i \in I}$ of non-empty subsets of $\mathbb{B}$ indexed by $I \subseteq \mathbb{B}$.*

Definition 3.3.7 (Weihrauch Reducible (3)). *Let $(F_i)_{i \in I}$ and $(G_j)_{j \in J}$ be problems in the sense of the previous definition.*

Then f is Weihrauch reducible to g , written $f \leq''_W g$, if there exist a computable map $\varphi : I \rightarrow J$, and a family of computable maps $(\psi_i : G_{\varphi(i)} \rightarrow F_i)_{i \in I}$.

Proposition 3.3.8. *The partial orders on the degrees induced by the orders $\leq'_W$ and $\leq''_W$ are isomorphic.*

Proof. Given a problem f , we can define a family $(f(\alpha))_{\alpha \in \text{dom}(f)}$. This process is clearly bijective.

Suppose $f \leq'_W g$ with forward map Φ and backward map Ψ . A moment's thought tells us that Φ is necessarily defined on $\text{dom}(f)$, so we set $\varphi := \Phi|_{\text{dom}(f)}$. Similarly, Ψ must be defined on the set $\sum_{\alpha \in \text{dom}(f)} g(\varphi(\alpha))$, so we can define $\psi_\alpha(\beta) := \Psi(\alpha, \beta)$. Then $(\varphi, (\psi_\alpha)_{\alpha \in \text{dom}(f)})$ is the desired reduction $(f(\alpha))_{\alpha \in \text{dom}(f)} \leq''_W (g(\beta))_{\beta \in \text{dom}(g)}$.

For the converse, suppose that $(f(\alpha))_{\alpha \in \text{dom}(f)} \leq''_W (g(\beta))_{\beta \in \text{dom}(g)}$, is witnessed by $(\varphi : \text{dom}(f) \rightarrow \text{dom}(g), (\psi_\alpha : g(\varphi(\alpha)) \rightarrow f(\alpha))_{\alpha \in \text{dom}(f)})$. We can construct a reduction $f \leq'_W g$ by $(\Phi := \varphi, \Psi(\alpha, \beta) := \psi_\alpha(\beta))$. Note that for Ψ to be type-2 computable, it must be computable to “look up” the map ψ_α . This is straightforward if we represent the family of maps as a type-2 function with domain $\text{dom}(f)$ in the first place.

Since we have a monotone order-embedding, the orders are isomorphic as preorders, and this induces an isomorphism on degrees. $\square$

Definitions 3.3.6 and 3.3.7 only really refer to type-2 computability in two instances: that our families are defined in terms of subsets of $\mathbb{B}$, and in the requirement that our maps be (type-2) computable. This makes this definition very amenable to generalisation. Given the categorical perspective of this thesis, we desire a definition paramaterised by a category $\mathcal{C}$ which we will take to be our types and computable maps. The means to perform this is well-known in the category theory literature: we generalise the equivalence of sets indexed by I and functions into I .

Definition 3.3.9 (Fam(I)). *Let I be a set. The category Fam(I) of families indexed by I consists of families of sets $(X_i)_{i \in I}$. Arrows $(X_i)_{i \in I} \rightarrow (Y_i)_{i \in I}$ are given by collections of functions $(f_i : X_i \rightarrow Y_i)_{i \in I}$; composition and identities are defined component-wise.*

Definition 3.3.10 (Slice Category). *Let $\mathcal{C}$ be a category and $I \in \mathcal{C}$. The slice category $\mathcal{C}/I$ has arrows $f : X \rightarrow I$ as objects and commuting triangles as morphisms.*

$$\begin{array}{ccc} X & \xrightarrow{h} & Y \\ & \searrow f & \swarrow g \\ & I & \end{array}$$

Proposition 3.3.11. ³² *For any set I , Set/I and $\text{Fam}(I)$ are equivalent categories.*

Proof. There is a functor $\text{Set}/I \rightarrow \text{Fam}(I)$ sending functions $f : X \rightarrow I$ to the family of preimages $(f^{-1}(i))_{i \in I}$. We can also define a functor $\text{Fam}(I) \rightarrow \text{Set}/I$ sending families $(F_i)_{i \in I}$ to projections from the disjoint union $\pi_1 : \sum_{i \in I} F_i \rightarrow I$. It is easy to check this an equivalence. $\square$

Representing families by collections $(X_i)_{i \in I}$ will be referred to as *point-wise indexing* and representing them by functions will be referred to as *display indexing*. This suggests that we consider the slice category $\mathcal{C}/I$ to be families in a category $\mathcal{C}$. Indeed, using the internal languages of categories, a morphism $f : X \rightarrow I$ is denoted by the family $(X_i)_{i \in I}$. This notation is used extensively, e.g., in Abbott’s thesis ³³, which we are building upon.

The operation of reindexing a family, i.e., forming $(X_{\varphi(i)})_{i \in I}$ for some map $\varphi : I \rightarrow J$, corresponds to taking a pullback of the corresponding map $\pi_1 : X \rightarrow I$ along φ . Thus we can formulate general definitions of problems and reductions in categories with pullbacks.

Definition 3.3.12 (Problem). *Let $\mathcal{C}$ be a category with pullbacks. A problem is simply a morphism in $\mathcal{C}$.*

Definition 3.3.13 (Reduction). *Let $\mathcal{C}$ be a category with pullbacks and let $f : X \rightarrow I, g : Y \rightarrow J$ be problems. A reduction $f \rightarrow g$ is an (equivalence class of) pairs of morphisms $(\varphi : I \rightarrow J, \psi : Y \times_J I \rightarrow X)$ such that the following diagram commutes.*

$$\begin{array}{ccccc} X & \xleftarrow{\psi} & Y \times_J I & \xrightarrow{\quad} & Y \\ f \downarrow & & \downarrow & \lrcorner & \downarrow g \\ I & \xlongequal{\quad} & I & \xrightarrow{\varphi} & J \end{array}$$

Remark 3.3.14. *In the internal language, the morphism $\psi : Y \times_J I \rightarrow X$ in the slice $\mathcal{C}/I$ is denoted by a family of maps $(\psi_i : Y_{\varphi(i)} \rightarrow X_i)_{i \in I}$.*

Such reductions are known to category theorists as “container morphisms” and the category of containers will be the main topic of [Section 3.4](#). We will defer proving that Weihrauch reductions are container morphisms (in an appropriate base category) until [Theorem 3.5.3](#).

3.4 An Introduction to Containers

CONTAINERS WERE INTRODUCED BY Abbott et al. ³⁴ as “a new approach to the theory of datatypes based on separating data and storage”. The idea is that elements of a type, e.g., of binary trees, can be viewed as consisting of a

³² E.g., Jacobs, *Categorical Logic and Type Theory*, Proposition 1.2.2.

³³ Abbott, ‘Categories of Containers’.

³⁴ Abbott, Altenkirch and Ghani, ‘Categories of Containers’.

“shape” and a function assigning “positions” in that shape to concrete data. More formally, a container $S \triangleright P$ consists of a type S and a family of types $(P(s))_{s:S}$. Morphisms $S \triangleright P \rightarrow T \triangleright Q$ between containers then naturally factor into two parts: a map $\varphi : S \rightarrow T$ on shapes, and a collection of morphisms $(Q(\varphi(s)) \rightarrow P(s))_{s:S}$ sending positions to positions.

As an example, consider the type of sequences $\text{List}(X)$ with entries in X . Any list can be represented by a length $n : \mathbb{N}$ and function $\{1, \dots, n\} \rightarrow X$ whose domain is a fixed type with n elements. So we could define $\text{List}(X) := (\{1, \dots, n\} \rightarrow X)_{n:\mathbb{N}}$. Now if we want to define the function $\text{reverse} : \text{List}(X) \rightarrow \text{List}(X)$ that reverses a list. Reversing a list preserves lengths, and for each length n , we send the element in the i -th position to the $(n + 1 - i)$ -th position. That is, $\text{reverse}(n, f) := (n, \lambda i. f(n + 1 - i))$.

As previously discussed in [Section 3.3](#), the category-theoretic definition involves treating containers in a category $\mathcal{C}$ as internal families, i.e., as $\mathcal{C}$ -morphisms, and container morphisms will be represented by commuting diagrams.

Definition 3.4.1 (Container). *Let $\mathcal{C}$ be a category with pullbacks. A container $f : I \triangleright X$ over $\mathcal{C}$ is a morphism $f : X \rightarrow I$ in $\mathcal{C}$. The shape of f is the codomain $\text{shape}(f) := \text{cod}(f)$. The positions of f is the domain $f_{\text{pos}} := \text{dom}(f)$.*

Convention 3.4.2. *As containers are maps in a category, we will inherit the usual convention of lower-case $f, g, h, \dots$ for the names of containers, and capital letters for their shapes $I, J, K, \dots$ and positions $X, Y, Z, \dots$.*

Definition 3.4.3 (Morphism Representatives). *Let $\mathcal{C}$ be a category with pullbacks, and $f : X \rightarrow I, g : Y \rightarrow J$ be containers over $\mathcal{C}$. A representative for a morphism $f \rightarrow g$ is a pair of morphisms $(\varphi : I \rightarrow J, \psi : Y \times_J I \rightarrow X)$ ³⁵ such that the following diagram commutes.*

$$\begin{array}{ccccc} X & \xleftarrow{\psi} & Y \times_J I & \xrightarrow{\quad} & Y \\ f \downarrow & & \downarrow & \lrcorner & \downarrow g \\ I & \xlongequal{\quad} & I & \xrightarrow{\varphi} & J \end{array}$$

Since pullbacks are only defined up to unique isomorphism, two morphism representatives are equivalent, $(\varphi, \psi) \sim (\varphi', \psi')$, whenever $\varphi = \varphi'$ and $\psi = \psi' \circ \chi$, where $\chi : \text{dom}(\psi) \rightarrow \text{dom}(\psi')$ is the unique isomorphism between those pullbacks.

Definition 3.4.4 (Container Morphisms). *Let f, g be containers over $\mathcal{C}$. A container morphism $f \rightarrow g$ is an equivalence class $[(\varphi, \psi)]_{\sim}$ of representatives of morphisms $f \rightarrow g$.*

As expected, containers over $\mathcal{C}$ form a category³⁶.

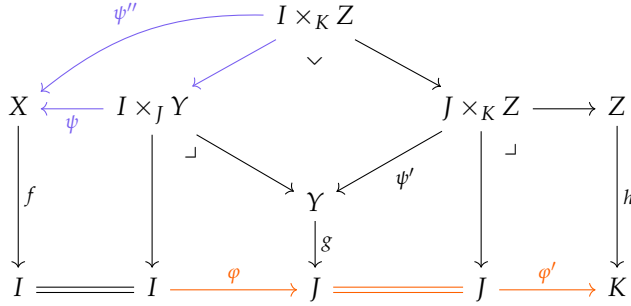
Proposition 3.4.5. *Let $\mathcal{C}$ be a category with pullbacks. The category $\text{Cont}(\mathcal{C})$ of containers over $\mathcal{C}$ has containers as objects and container morphisms as arrows.*

Proof. For a given container $f : X \rightarrow I$, the identity container id_f is the (equivalence class) of morphism representative $(\text{id}_I, \text{id}_X)$.

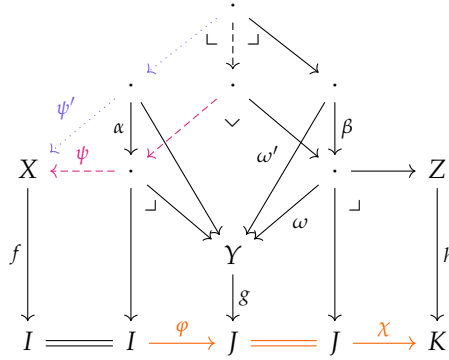
³⁵ We will use orange for forward maps and blue for backwards maps. This is meant to be a visual aid only and isn't normative.

³⁶ An alternative proof of this fact is that $\text{Cont}(\mathcal{C})$ is the total space of the shape fibration, as discussed in [Section 3.4.2](#).

For containers $f : X \rightarrow I$, $g : Y \rightarrow J$, $h : Z \rightarrow K$, and container morphisms $[(\varphi, \psi)]_{\sim} : f \rightarrow g$, $[(\varphi', \psi')]_{\sim} : g \rightarrow h$, the composition is given by the (equivalence class) of the representative $(\varphi' \circ \varphi, \psi'')$, as given by the following commuting diagram.

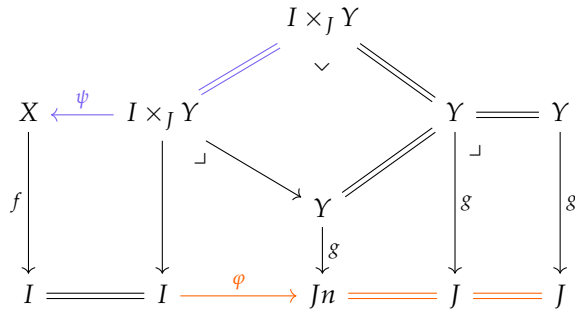


Composition is well-defined Suppose we have two representatives (φ, ψ) , (φ, ψ') of a morphism $f \rightarrow g$, and two representatives (χ, ω) , (χ, ω') of a morphism $g \rightarrow h$. Then we have isomorphisms α, β such that the following diagram commutes.

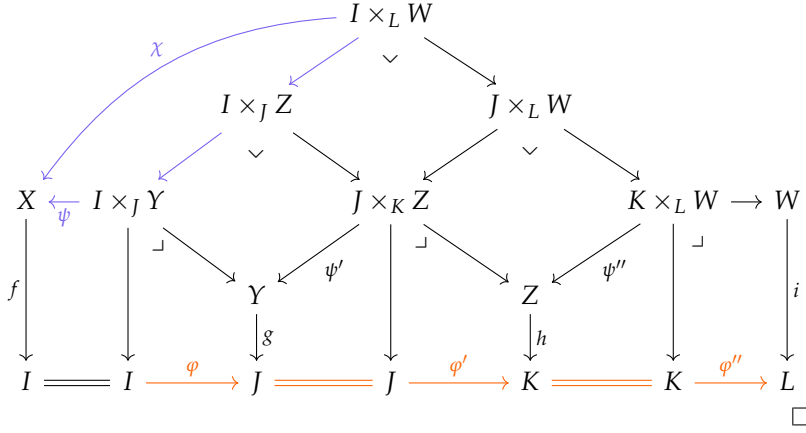


Note the dashed arrow is an isomorphism as it is the pullback of an isomorphism. Setting ψ'' to be the composition of **dotted arrows** and ω'' to be the composition of **dashed arrows**, we see that $(\chi, \omega) \circ (\varphi, \psi) = (\chi \circ \varphi, \psi'') \sim (\chi \circ \varphi, \omega'') = (\chi, \omega') \circ (\varphi, \psi')$.

Identity laws Let $[(\varphi, \psi)]_{\sim} : f \rightarrow g$ be a container morphism. Then the following diagram shows that (φ, ψ) is morphism representative for both $[(\varphi, \psi)]_{\sim}$ and $[(\varphi, \psi)]_{\sim} \circ \text{id}$, and hence we have equality. A similar diagram establishes that $\text{id} \circ [(\varphi, \psi)]_{\sim} = [(\varphi, \psi)]_{\sim}$.



Associativity of Composition Suppose we have three container morphisms $[(\varphi, \psi)]_{\sim} : f \rightarrow g$, $[(\varphi', \psi')]_{\sim} : g \rightarrow h$, and $[(\varphi'', \psi'')]_{\sim} : h \rightarrow i$. To see that composition is associative, it is enough to verify that the commuting diagram below gives a representative $(\varphi'' \circ \varphi' \circ \varphi, \chi)$ for both sides of the identity.



It will frequently be convenient to work with representatives directly in our proofs rather than equivalence classes, and to leave out proofs that certain constructions don't depend on the choice of representatives.

The theory of containers as set out by Abbott et al. in their original paper³⁷, and in Abbott's thesis³⁸, is mainly studied over base categories which are locally cartesian closed. This makes sense, as they were developed in a dependent type theory, and it is well-known that such type theories are the internal languages of locally cartesian closed categories³⁹. The use of such type theories can simplify proofs over working with diagrams.

However, as we have just seen, the category of containers $\text{Cont}(\mathcal{C})$ can be defined whenever $\mathcal{C}$ has pullbacks⁴⁰. Assuming more gives you a much richer theory, of course, e.g., Abbott et al. showed that $\text{Cont}(\mathcal{C})$ will inherit completeness and cocompleteness from $\mathcal{C}$; but to a certain extent, you can pay only for what you need to use. We will defer discussion of the algebraic structure of containers until Chapter 4.

3.4.1 Connection to Polynomial Functors

IT WILL BE USEFUL to make explicit the connection to the theory of polynomial functors hinted at in the introduction. Throughout this section, we will assume our base category $\mathcal{C}$ is locally cartesian closed, and we write the adjunctions to pullback along f by $\Sigma_f \dashv f^* \dashv \Pi_f$.

Definition 3.4.6 (Polynomial, Polynomial Functor).⁴¹ A polynomial over $\mathcal{C}$ is the same thing as an indexed container^{42 43}, i.e., a diagram F in $\mathcal{C}$ of shape

$$I \xleftarrow{s} B \xrightarrow{f} A \xrightarrow{t} J.$$

To each polynomial f , we associate a polynomial functor $P_f : \mathcal{C}/I \rightarrow \mathcal{C}/J$ defined by $P_f = \Sigma_t \circ \Pi_f \circ s^*$.

³⁷ Abbott, Altenkirch and Ghani, 'Categories of Containers'.

³⁸ Abbott, 'Categories of Containers'.

³⁹ This observation goes back to Seely's work in the 80s (Seely, 'Locally cartesian closed categories and type theory').

⁴⁰ The theory of containers in this restricted setting was developed by Weber (Weber, 'Polynomials in categories with pullbacks').

⁴¹ Gambino and Kock, 'Polynomial functors and polynomial monads', ¶1.4.

⁴² *ibid.*, ¶2.18.

⁴³ Altenkirch, Ghani, Hancock, McBride and Morris, 'Indexed containers'.

In the specific case of $I = J = 1$, a polynomial is a container. In this case, the polynomial functor associated to a polynomial $f : B \rightarrow A$ is of the form $P_F(X) = \sum_{a \in A} X^{B_a}$, justifying the terminology.

Polynomial functors have a canonical strength ⁴⁴, and form a category $\text{Poly}_{\mathcal{C}}(I, J)$, where the morphisms are strong natural transformations. Polynomial functors factorise in a nice way.

Proposition 3.4.7. ⁴⁵ *Every strong natural transformation $P_G \Rightarrow P_F$ between polynomial functors is represented in an essentially unique way by a diagram of the form:*

$$\begin{array}{ccccc}
 G : & I & \xleftarrow{u} & D & \xrightarrow{g} & C & \xrightarrow{v} & J \\
 & \parallel & & \uparrow & & \uparrow & & \parallel \\
 & & & B' & \xrightarrow{\quad} & C & & \\
 & & & \downarrow & \lrcorner & \downarrow & & \\
 F : & I & \xleftarrow{s} & B & \xrightarrow{f} & A & \xrightarrow{t} & J
 \end{array}$$

In the case $I = J = 1$, we can see that the diagram is that of (a representative of) a container morphism. More specifically, we have an equivalence of categories.

Lemma 3.4.8. ⁴⁶ *For any I, J , the functor $F \mapsto P_F$ is an equivalence of categories $\text{Cont}(\mathcal{C})(I, J) \cong \text{Poly}_{\mathcal{C}}(I, J)$ where $\text{Cont}(\mathcal{C})(I, J)$ denotes the category of indexed containers.*

The category of polynomial functors is very well-endowed, particularly in the case of $\mathcal{C} = \text{Set}$, and a useful reference is provided by Spivak ⁴⁷. In the sequel, we will prefer to use the terminology of containers for general $\mathcal{C}$, reserving the terminology of polynomial functors for the case where $\mathcal{C}$ is locally cartesian closed.

3.4.2 Shape Fibration

IT IS EASY TO SEE that there is a functor mapping containers to their shapes. This functor has the special property of being a fibration⁴⁸.

Proposition 3.4.9. *Let $\mathcal{C}$ be a category with pullbacks. There is a fibration shape : $\text{Cont}(\mathcal{C}) \rightarrow \mathcal{C}$ sending $f : X \rightarrow I$ to I .*

Proof. Let $\varphi : I \rightarrow J$ be a map in $\mathcal{C}$ and $g : Y \rightarrow J$ be a container. Then a cartesian container morphism above φ can be represented by (φ, id) , i.e., cartesian morphisms are essentially pullback squares. That this morphism is cartesian follows by the pullback lemma. More generally, (φ, ψ) is a cartesian morphism whenever ψ is an isomorphism. $\square$

Vertical container morphisms are those represented by (id, ψ) for some map ψ , i.e., $(\mathcal{C}/I)^{\text{op}}$ is equivalent to the category of vertical morphisms over $I \in \mathcal{C}$.

From the description of cartesian and vertical morphisms above, and the definition of the category of containers, it is easy to see that the shape

⁴⁴ Gambino and Kock, ‘Polynomial functors and polynomial monads’, Proposition 1.15.

⁴⁵ *ibid.*, Theorem 2.12.

⁴⁶ *ibid.*, Lemma 2.15.

⁴⁷ Spivak, *A reference for categorical structures on Poly*.

⁴⁸ Basic facts about fibrations are collected in Section A.4.

fibration is equivalent to the opposite of the codomain fibration (see [Definition A.4.11](#) and [Example A.4.3](#)). Indeed, this is the content of [Proposition 3.4.7](#). This is not mere abstract nonsense; Spivak has recently argued⁴⁹ that taking the opposite of a fibration is the key to generalised forms of lenses⁵⁰. In fact, we will see this again in [Section 6.2](#) when we tackle strong Weihrauch reducibility.

3.5 Weihrauch Problems are Containers over Partitioned Modest Sets

WE ARE NOW READY to return to the thread of [Section 3.3](#), and to explain in what sense Weihrauch reducibility is an instance of the theory of containers. The base category we work with is the category of partitioned modest sets over Baire space, specifically $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$, which we have already seen to be equivalent to rpReprSp ([Corollary 3.1.12](#)).

There is a small detail to take care of before proving the main theorem: some of the containers in $\text{Cont}(\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$ may represent families of subsets of $\mathbb{B}$ that have the empty set as a member. Weihrauch problems do not allow that, so we need to restrict the class of objects to rule out those containers.

Definition 3.5.1 (Answerable). *A container is called answerable if the underlying map is a pullback-stable epimorphism. The full subcategory of answerable containers is denoted $\text{Cont}_{\text{Ans}}(\mathcal{C})$.*

As already proven in [Proposition 2.3.14](#), pullback-stable epimorphisms in Asm and pMod are those which are surjective; this ensures each fibre is non-empty. If we wish to specifically include non-answerable containers we will refer to objects of (the posetal reflection of) $\text{Cont}(\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$ as *partial Weihrauch problems (degrees)*⁵¹.

As a shorthand, we will give names to two particular categories of containers.

Definition 3.5.2 (Wei, CtsWei). *The category of Weihrauch reductions, Wei, is the category of answerable containers over the category of projective represented spaces and (type 2) computable functions $\text{Cont}_{\text{Ans}}(\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$.*

Similarly, the category of continuous Weihrauch reductions, CtsWei, is the category of answerable containers over projective represented spaces and continuous functions $\text{Cont}_{\text{Ans}}(\text{pMod}(\mathcal{K}_2))$.

The following theorem states that we should view Wei as the “proof relevant” version of the poset of Weihrauch degrees.

Theorem 3.5.3. *The Weihrauch degrees are isomorphic to the posetal reflection of Wei.*

Proof. To each answerable container $P : X \rightarrow U$ over $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$, we can map the P to the Weihrauch problem $w(P) : \subseteq \mathbb{B} \rightrightarrows \mathbb{B}$ defined by

$$w(P)(p) := \{q \in \mathbb{B} \mid \exists x \in |X|. p \Vdash_U P(x) \wedge q \Vdash_X x\}.$$

For any other $Q : Y \rightarrow V$ and reduction representative $(\varphi, \psi) : P \rightarrow Q$, we can find an equivalent one (φ, ψ') where the domain of ψ' is actually

⁴⁹ Spivak, *Generalized Lens Categories via functors $\mathcal{C}^{\text{op}} \rightarrow \text{Cat}$* .

⁵⁰ Lenses are a common alternative name for container morphisms. We will refrain from using this terminology in this thesis, despite preferring it in conversation.

⁵¹ Previously, these were referred to as *slightly extended problems (degrees)* (Pradic and Price, ‘*Weihrauch Problems as Containers*’), however, the original name was felt to be clunky, and “partial” was settled on after consultation with Eike Neumann, Arno Pauly, and Manlio Valenti.

the set of pairs (u, y) with $Q(y) = \varphi(u)$ as it is a pullback. Then (φ, ψ') is literally a Weihrauch reduction from $w(P)$ to $w(Q)$.

In the other direction, given a Weihrauch problem $f : \subseteq \mathbb{B} \rightrightarrows \mathbb{B}$, we have a map $c(f) : \sum_{u \in \text{dom}(f)} f(u) \xrightarrow{\pi_1} \text{dom}(f)$; we can regard its domain and codomain as carriers of partitioned modest sets by giving them trivial realisability relations $\langle e, e' \rangle \Vdash_{\text{dom}(c(f))} (e, e')$ and $e \Vdash_{\text{dom}(f)} e$. By definition, if a reduction $f \leq_W g$ is represented by computable maps (φ, ψ) , we have $\varphi : \text{dom}(f) \rightarrow \text{dom}(g)$ and $\psi : \sum_{u \in \text{dom}(f)} g(\varphi(u)) \rightarrow \bigcup_{u \in \text{dom}(f)} f(u)$. Since $\psi(u, y)$ is meant to give an answer to u , by combining ψ with the projection $\sum_{u \in \text{dom}(f)} g(\varphi(u)) \rightarrow \text{dom}(f)$, we can form a map $\tilde{\psi} : \sum_{u \in \text{dom}(f)} g(\varphi(u)) \rightarrow \sum_{u \in \text{dom}(f)} f(u)$ so that $(\varphi, \tilde{\psi})$ is a morphism representative $c(f) \rightarrow c(g)$.

To conclude, it suffices to show $c(w(P)) \cong P$ and $w(c(f)) =_W f$ for all P and f .

For the first part, for $P : X \rightarrow U$, because P is answerable, we do also have $\text{dom}(w(P)) = \{p \in \mathbb{B} \mid \exists u \in |U|. p \Vdash_U u\}$; this is enough to derive an isomorphism $\theta : \text{dom}(w(P)) \cong U$ in $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ which is tracked by the code for the identity. Then we can easily check that $c(w(P))$ is a pullback of P along θ , hence (θ, id) and (θ^{-1}, id) represent an isomorphism $c(w(P)) \cong P$ as expected.

For the second part, we have that $w(c(f))(p) = \{\langle p, x \rangle \mid x \in f(p)\}$ for every $p \in \mathbb{B}$. So the forward component of both reductions we want can be taken to be the identity, the backward component of the reduction $f \leq_W w(c(f))$ can project away the first component of the output of $w(c(f))$ while the backward part of the converse can be the identity. $\square$

It is important to note that we did not use computability theory in the proof, except insofar as we required our morphisms to be type-2 computable. As partial continuous maps are all representable in $\mathcal{K}_2$, the following theorem can be proven in the same way.

Theorem 3.5.4. *The continuous Weihrauch degrees are isomorphic to the posetal reflection of CtsWei .*

3.6 Extended Weihrauch Degrees and Containers over Partitioned Assemblies

WE WILL END THIS CHAPTER with a discussion of Bauer's notion of extended Weihrauch predicates, and how they fit into the current program⁵².

Bauer's extended Weihrauch predicates⁵³ are a specialisation of his notion of instance reducibility to realisability toposes built from filter-PCAs. The rationale is that, in the Kleene-Vesley topos⁵⁴, a Weihrauch problem P can be translated to predicates $P'(i)$ (over some object of the topos, which corresponds to the inputs of P) such that $\forall i. \exists j. Q'(j) \Rightarrow P'(i)$ holds if and only if P is Weihrauch-reducible to Q . This is essentially because we are in a realisability topos, and the truth of this statement is necessarily witnessed by a program which encodes the forward part of a reduction due to the $\forall\exists$ and a backwards part due to the $\Rightarrow$ part.

⁵² While these degrees can be accommodated by our container framework, they will play no essential role in the rest of this thesis.

⁵³ Bauer, 'Instance reducibility and Weihrauch degrees'.

⁵⁴ van Oosten, *Realizability: An Introduction to its Categorical Side*, §4.5.

By unwinding the definition, we can state extended reducibility more elementarily from the definition of a filter-PCA; let us do that, assuming an arbitrary filter-PCA $(\mathbb{A}', \mathbb{A})$ is fixed for the rest of the subsection.

Definition 3.6.1 (Extended Weihrauch Predicate, Reducibility). ^{55,56} An extended Weihrauch predicate is a map $p : \mathbb{A} \rightarrow \mathcal{P}(\mathcal{P}(\mathbb{A}))$. Its support is the set $\|p\| = \{r \in \mathbb{A} \mid p(r) \neq \emptyset\}$. p is said to be reducible to another extended Weihrauch predicate q when there exists codes $e_{\text{fwd}}, e_{\text{bwd}} \in \mathbb{A}'$ such that, for every $r \in \|p\|$:

1. $e_{\text{fwd}} \cdot r$ is defined and belongs to $\|q\|$
2. for every $\theta \in p(r)$ there is $\xi \in q(e_{\text{fwd}} \cdot r)$ such that $e_{\text{bwd}} \cdot r \cdot s$ is defined and belongs to θ for any $s \in \xi$.

For the sequel, let us restate 2, using the axiom of choice, as follows:

- 2' we have a function $f_r : p(r) \rightarrow q(e_{\text{fwd}} \cdot r)$ such that $e_{\text{bwd}} \cdot r \cdot s$ is defined and belongs to θ for any $s \in f_r(\theta)$.

With this version, it is helpful to think of $(e_{\text{fwd}}, (f_r)_{r \in \|p\|})$ as a code for a forward reduction and e_{bwd} as a code for a backward reduction. In the case of extended Weihrauch predicate p coming from Weihrauch problems (or partial Weihrauch problems), $p(r)$ will be always a (sub)singleton. In the general case, p corresponds to a binary partial multi-valued function $\mathbb{A} \times \mathbb{A} \rightrightarrows \mathbb{A}$ where the first input corresponds to r , the second to an element of $p(r)$ and the forward part of the reduction is allowed to be an arbitrary set-theoretic map while the backward reduction cannot compute at all with the second input.

Example 3.6.2. ⁵⁷ An example of an extended Weihrauch predicate which does not match any Weihrauch degree is WLEM which is formally defined as the constant function mapping any $r \in \mathbb{A}$ to $\{\underline{0}\}, \{\underline{1}\}\}$, where $\underline{i}$ is the canonical code for the number i in $\mathbb{A}'$.

Assuming the axiom of choice, it is maximal for reducibility among all extended Weihrauch predicates p with $\bigcup p(r) \subseteq \{\underline{0}, \underline{1}\}$; clearly there is no such maximal Weihrauch problem.

The following theorem justifies calling objects of $\text{pAsm}(\mathbb{A}, \mathbb{A}')$ extended Weihrauch problems.

Theorem 3.6.3. Assuming the axiom of choice, the extended Weihrauch degrees over a filtered PCA $(\mathbb{A}', \mathbb{A})$ are isomorphic to the posetal reflection of the category of containers over partitioned assemblies $\text{pAsm}(\mathbb{A}, \mathbb{A}')$.

Note that the proof, as well as the rest of the section, uses the axiom of choice liberally.

Proof. First let us explain how to turn an extended Weihrauch predicate p into a container $\hat{p}$. We first define the assemblies I_p and X_p of shapes and positions of $\hat{p}$ as follows:

⁵⁵ Bauer, ‘Instance reducibility and Weihrauch degrees’, Definition 3.7.

⁵⁶ See also Kihara, *Rethinking the notion of oracle: A prequel to Lawvere-Tierney topologies for computability theorists*, Definition 5.7.

⁵⁷ Bauer, ‘Instance reducibility and Weihrauch degrees’.

$$|X_p| = \{(r, \theta, s) \in \mathbb{A} \times \mathcal{P}(\mathbb{A}) \times \mathbb{A} \mid r \in \|p\|, \theta \in p(r), s \in \theta\}$$

$$\langle r, s \rangle \Vdash_{X_p} (r, \theta, s)$$

$$|I_p| = \{(r, \theta) \in \mathbb{A} \times \mathcal{P}(\mathbb{A}) \mid r \in \|p\|, \theta \in p(r)\}$$

$$r \Vdash_{I_p} (r, \theta)$$

It is obvious those assemblies are partitioned. $\widehat{p}$ is then defined by the projection $(r, \theta, s) \mapsto (r, \theta)$, which is tracked by code of the first projection in $\mathbb{A}'$. Furthermore, a reduction $((e_{\text{fwd}}, f), e_{\text{bwd}})$ between extended Weihrauch predicates p and q induces a map $(\Phi, \Psi) : \widehat{p} \rightarrow \widehat{q}$ in $\text{Cont}(\text{pAsm}(\mathbb{A}, \mathbb{A}'))$:

$$\begin{aligned} \Phi(r, \theta) &:= (e_{\text{fwd}} \cdot r, f(\theta)) \\ &\text{tracked by } e_{\text{fwd}} \\ \Psi((r, \theta), (e_{\text{fwd}} \cdot r, f(\theta), s)) &:= (r, \theta, e_{\text{bwd}} \cdot r \cdot s) \\ &\text{tracked by } \lambda \langle r, \langle t, s \rangle \rangle. e_{\text{bwd}} \cdot r \cdot s. \end{aligned}$$

Conversely, given a container $P : X \rightarrow I$ in $\text{Cont}(\text{pAsm}(\mathbb{A}, \mathbb{A}'))$, define the extended Weihrauch predicate $\phi_P(r) = \{\Phi_P(i) \mid r \Vdash_I i\}$, where $\Phi_P(i) = \{e \in \mathbb{A} \mid \exists s \in P^{-1}(i). e \Vdash_X s\}$.

Using the axiom of choice, let us fix a section $\zeta_{P,r}$ of the restriction of Φ_P to the $i \in I$ with $r \Vdash_I i$.

If we have a map (φ, ψ) from P to $Q : Y \rightarrow J$ in $\text{Cont}(\text{pAsm}(\mathbb{A}, \mathbb{A}'))$, where $e_\varphi \Vdash_{I \rightarrow J} \varphi$ and e_ψ tracks ψ , we can turn this into a reduction between extended predicates by taking $(e_\varphi, (\varphi'_r)_{r \in \|\phi_P\|})$ to be the forward map, where $\varphi'_r(X)$ is defined as follows.

$$\begin{aligned} \varphi'_r : \quad & \|\phi_P\|(r) \longrightarrow \|\phi_Q\|(e_\varphi \cdot r) \\ & X \longmapsto \Phi_Q(\varphi(\zeta_{P,r}(X))) \end{aligned}$$

Assuming without loss of generality that the domain of ψ is the assembly where codes are pairs of codes for I and Y , we take $\lambda x. \lambda y. e_\psi \cdot \langle x, y \rangle$ as the code for the backwards part and check that this gives us a reduction from ϕ_P to ϕ_Q .

Now let us show that $\phi_{\widehat{p}}$ is always equivalent to p by exhibiting reductions both ways. First, note that $\|\phi_{\widehat{p}}\| = \{r \mid \exists \theta \in p(r). r \Vdash_{I_p} (r, \theta)\} = \|p\|$ and that $\Phi_{\widehat{p}}(r, \theta) = \{\langle r, s \rangle \in \mathbb{A} \mid s \in \theta\}$. For the reduction from $\phi_{\widehat{p}}$ to p , we can take the code for the identity together with the map $\xi \mapsto \{\pi_2 \cdot e \mid e \in \xi\}$ and the code $\lambda r. \lambda s. \langle r, s \rangle$ for the backward part. For the other direction, the forward part is obtained by taking the code for the identity together with the r -indexed family of maps $s \mapsto \langle r, s \rangle$, and the backward part by $\lambda x. \lambda y. \pi_2 y$.

Finally, we also show that there exists morphisms both ways between $P : X \rightarrow I$ and $\widehat{\phi_P}$ in $\text{Cont}(\text{pAsm}(\mathbb{A}, \mathbb{A}'))$. First, note that the carrier of the object I' of positions of $\widehat{\phi_P}$ is the set of all pairs $(r, \Phi_P(i))$ with $r \Vdash_I i$. There is an obvious (surjective) map $I \rightarrow I'$ taking i to the unique pair

$(r, \Phi_P(i))$ such that $r \Vdash_I i$, which is tracked by $\lambda x. x$. Thanks to the axiom of choice, this map has a section $s : I' \rightarrow I$ which is also tracked by $\lambda x. x$. These two maps will be the forward part of the reductions we want.

For the backward part, first note the carrier X' of the object of directions of $\widehat{\phi_P}$ is the set of all triples $(r, \Phi_P(i), e)$ with $r \Vdash_I i$ and $e \in \Phi_P(i)$, that is those e tracking some element of $x \in X$ such $P(x) = i$, and such a triple is tracked by the pair $\langle r, e \rangle$. Since P is a map in $\mathbf{pAsm}(\mathbb{A}, \mathbb{A}')$, we have that e determines the value of all such triples. Further, P is tracked by some $e_P \in \mathbb{A}'$, so we have that $e_P \cdot e = r$ for every $(r, \Phi_P(i), e) \in |X'|$. So $\lambda x. \langle e_P \cdot x, x \rangle$ tracks (and determines) a map $X \rightarrow X'$ which can be used to complete the reduction from $\widehat{\phi_P}$ to P .

In the other direction, we appeal to the axiom of choice to get functions $\epsilon_{P,r,i}$ such that $e \Vdash_X \epsilon_{P,r,i}(e)$ and $P(x) = i$ for every $(r, \Phi_P(i), e) \in |X'|$. We can use that to define the map

$$\begin{array}{ccc} X' & \longrightarrow & X \\ (r, \theta, e) & \longmapsto & \epsilon_{P,r,s(r,\theta)}(e) \end{array}$$

which is a morphism $X' \rightarrow X$ in $\mathbf{pAsm}(\mathbb{A}, \mathbb{A}')$ tracked by $\lambda x. x$ and can similarly be used to complete the reduction from P to $\widehat{\phi_P}$. $\square$

Example 3.6.4. *The Weihrauch predicate WLEM translates to a container isomorphic to $\mathbb{2} \rightarrow \nabla(\mathbb{2})$, where $\nabla : \mathbf{Set} \rightarrow \mathbf{pAsm}(\mathbb{A}, \mathbb{A}')$ is the right adjoint to the forgetful functor $\mathbf{pAsm}(\mathbb{A}, \mathbb{A}') \rightarrow \mathbf{Set}$ sending a partitioned assembly to its carrier. It sends a set X to the partitioned assembly with carrier X and realisability relation induced by $\underline{0} \Vdash_{\nabla(X)} x$ (so all elements share the same code). The underlying map $\mathbb{2} \rightarrow \nabla(\mathbb{2})$ between carriers is the identity and is tracked by $\lambda x. \underline{0}$.*

Remark 3.6.5. *One may note that the definition of extended Weihrauch predicates and reduction between those comports a certain asymmetry, where the input as well as the forward reduction is split between a computable and non-computable part while the output and backward reduction do not contain any non-computable component. This contrasts the notion of extended Weihrauch problem, where inputs and outputs are treated more uniformly as partitioned assemblies. In this setting, the computational part consist of valid codes for the partitioned assemblies, while the non-computational part consists of the elements of the carrier tracked by the codes, so in principle we could have extended Weihrauch problems with “non-computational outputs”.*

This does not contradict [Theorem 3.6.3](#) because we can essentially turn any extended Weihrauch problem $P : X \rightarrow U$ into an equivalent (but not isomorphic!) one where the non-computational is suppressed, assuming the axiom of choice: one can quotient the carrier of X by taking x and x' to be equivalent if $P(x) = P(x')$ and they are tracked by the same code. Then we obtain a partitioned assembly X' and a problem $X' \rightarrow U$ which turns out to be equivalent provided that we have a section to the quotient map $X \rightarrow X'$; this section always exists if we assume the axiom of choice.

4 The Algebraic Structure of Problems

IN THIS CHAPTER, we will explore how the algebraic structure of the Weihrauch degrees already exists in containers. Operators will typically correspond to (quasi-)functors over containers, implying they are monotone on degrees. Those functors can also be defined *generically* without reference to the specifics of type-2 computability; as such, we will define them over $\text{Cont}(\mathcal{C})$, making only minimal assumptions¹ about the base category $\mathcal{C}$.

In particular, we will show that the Weihrauch lattice reflects that the category of represented spaces ReprSp is distributive (Corollary 4.1.9), that the parallel product on Weihrauch degrees comes from a monoidal product on containers (Proposition 4.1.15), and that the composition of degrees is *almost, but not quite*, the composition of containers (Proposition 4.2.12). Composition is trickier to define, since ReprSp is not cartesian closed (Proposition 2.4.10), however, a weak form of container composition will be sufficient to recover the composition of Weihrauch degrees (Proposition 4.2.12).

The chapter ends with a brief digression on the Medvedev degrees, another notion of reducibility which is related to the Weihrauch degrees. To connect the two, we will take advantage of the shape fibration having both left and right adjoints.

4.1 Symmetric Operators on Containers

OF THE BINARY OPERATORS on the Weihrauch degrees, the easiest to handle are those which are commutative. In this section, we will prove that the degrees form a lattice and there is a parallel product.

4.1.1 The Weihrauch Lattice

THE FIRST IMPORTANT PROPERTY of the Weihrauch degrees is that they form a distributive lattice (Corollary 4.1.9). As we will see, this is a consequence of the fact that $\text{Cont}(\mathcal{C})$ will be a distributive category with only weak assumptions on the base category $\mathcal{C}$. More specifically, while the theory of containers usually assumes that the base category is *locally cartesian closed category*², for this subsection we will only require that sums “behave sensibly” with respect to pullbacks, i.e., that sums are disjoint and universal.

Definition 4.1.1 (Disjoint Sums). ³ Let $\mathcal{C}$ be category with sums and pullbacks

¹ Of course, these assumptions will be consistent with both $\text{pMod}(\mathbb{A}, \mathbb{A}')$ and $\text{pAsm}(\mathbb{A}, \mathbb{A}')$.

² Abbott, Altenkirch and Ghani, ‘Categories of Containers’.

³ Carboni, Lack and Walters, ‘Introduction to extensive and distributive categories’, Definition 2.5.

along their coprojections, then sums are disjoint if all coprojections are monic and the for any sum $X + Y$ the following diagram is a pullback:

$$\begin{array}{ccc} 0 & \xrightarrow{\quad} & Y \\ \downarrow & \lrcorner & \downarrow \iota_Y \\ X & \xrightarrow{\iota_X} & X + Y \end{array}$$

Definition 4.1.2 (Universal Sums). ⁴ Let $\mathcal{C}$ be a category with finite sums and pullbacks along their coprojections. A coproduct diagram⁵

$$X \xrightarrow{\iota_X} X + Y \xleftarrow{\iota_Y} Y$$

is universal if pulling it back along a morphism into $X + Y$ gives a coproduct diagram.

In addition to sums behaving nicely, it will be useful to have the initial object be strict.

Definition 4.1.3 (Strict Initial). ⁶ An initial object 0 is strict if any arrow into it $X \rightarrow 0$ is invertible.

The categories we are interested in that have these nice properties are the extensive categories.

Definition 4.1.4 ((L)extensive). ⁷ A category $\mathcal{C}$ with finite sums and products is called extensive when the canonical functor $+/ : \mathcal{C}/A \times \mathcal{C}/B \rightarrow \mathcal{C}/(A + B)$ is an isomorphism. $\mathcal{C}$ is lextensive⁸ if it has all finite limits, all finite coproducts and is extensive.

Proposition 4.1.5. Let $\mathcal{C}$ be a category with finite sums and pullbacks along coprojections. If $\mathcal{C}$ is an extensive category, then sums are universal, disjoint and initial objects are strict. Moreover, $\mathcal{C}$ is a lextensive category if and only if it has finite limits, and finite sums which are disjoint and universal.

Proof. All these properties are proven by Carboni et al. ⁹, specifically, Propositions 2.6, 2.8, 2.14 and Remark 4.10. $\square$

With these facts established, we can now prove that whenever the base category $\mathcal{C}$ is lextensive, $\text{Cont}(\mathcal{C})$ is a distributive category.

Proposition 4.1.6. ¹⁰ If $\mathcal{C}$ is lextensive, $\text{Cont}(\mathcal{C})$ has finite products, finite coproducts, and products distribute over coproducts.

Proof. It is easy to check that the maps $\text{id}_0 : 0 \rightarrow 0$ and $!_0 : 0 \rightarrow 1$ in $\mathcal{C}$ are respectively initial and terminal objects in $\text{Cont}(\mathcal{C})$; for the latter, this is because any pullback along a map $0 \rightarrow U$ has an initial object as domain¹¹.

Given two containers $f : X \rightarrow I$ and $g : Y \rightarrow J$, a coproduct $f + g : X + Y \rightarrow I + J$ is obtained by functoriality of $+/ : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$. The coprojections are horizontal morphisms represented (ι_I, id_X) and (ι_J, id_Y) ; note that X and Y are valid domains for the backward maps because sums are universal. The copairing $[(\varphi, \psi), (\varphi', \psi')] : f + g \rightarrow h$ of two reductions, $(\varphi, \psi) : f \rightarrow h$ and $(\varphi', \psi') : g \rightarrow h$, is represented by $([\varphi, \varphi'], \psi + \psi')$. This backwards map has the correct domain since, in an extensive category,

⁴ Carboni, Lack and Walters, ‘Introduction to extensive and distributive categories’, Definition 2.10.

⁵ A coproduct diagram is a cospan $X \xrightarrow{f} A \xleftarrow{g} Y$ where A is a coproduct of X and Y , and f, g are the coprojections.

⁶ *ibid.*, Definition 2.7.

⁷ *ibid.*, Definitions 2.1, 4.7.

⁸ The author considers “extensive and finitely complete” to be better terminology, but “lextensive” is standard.

⁹ *ibid.*

¹⁰ See also Spivak, *A reference for categorical structures on Poly*, (3), (6) and (7), for the Set case.

¹¹ Since initial objects are strict.

the pullback of a coproduct is the coproduct of the respective pullbacks along the same morphism.

A product of f and g can be defined as the map $[f \times \text{id}_J, \text{id}_I \times g] : X \times J + I \times Y \rightarrow I \times J$. The projection $f \times g \rightarrow f$ in $\text{Cont}(\mathcal{C})$ is represented by $(\pi_I, \iota_{X \times J})$; again, this is only a valid morphism because $\mathcal{C}$ is lexensive. Similarly, $(\pi_J, \iota_{I \times Y})$ represents the projection $f \times g \rightarrow g$. The pairing $\langle (\varphi, \psi), (\varphi', \psi') \rangle : f \rightarrow g \times h$ of two reductions $(\varphi, \psi) : f \rightarrow g$ and $(\varphi', \psi') : f \rightarrow h$, is represented by $(\langle \varphi, \varphi' \rangle, [\psi, \psi'])$. To see that the domain of the backwards map works, we first note if you pull a map f back along φ , the same object (with different projections) is a pullback of $f \times \text{id}_J$ along $\langle \varphi, \varphi' \rangle$. We can then invoke lexensivity to handle the coproduct of pullbacks.

To see that $\text{Cont}(\mathcal{C})$ is distributive, it is enough to notice that the canonical morphism $(f_1 \times g) + (f_2 \times g) \rightarrow (f_1 + f_2) \times g$ is represented by a horizontal morphism (c, id) where c is the canonical isomorphism in $\mathcal{C}$ witnessing distributivity for the base. $\square$

Corollary 4.1.7. ¹² *The Extended Weihrauch degrees form a bounded lattice, i.e., a distributive lattice with a top and bottom element.*

¹² Bauer, ‘Instance reducibility and Weihrauch degrees’, Propositions 2.3, 3.8.

Proof. Immediate by Theorem 3.6.3. $\square$

With the exception of the terminal object, this structure carries over to answerable containers.

Proposition 4.1.8. *Assuming $\mathcal{C}$ is lexensive, answerable containers of $\text{Cont}(\mathcal{C})$ are stable under binary products and coproducts. The initial object is always an answerable container, but the terminal object is not always answerable.*

Proof. Let $f : X \rightarrow I, g : Y \rightarrow J$ be answerable containers.

For binary products, we first note that $f \times g$ in $\text{Cont}(\mathcal{C})$ is obtained by copairing two pullback-stable epimorphisms: we can show that $f \times \text{id}_J$ is a pullback-stable epimorphism since it is the pullback of f along the projection $\pi_I : I \times J \rightarrow I$, and similarly for $\text{id}_I \times g$. Now, in extensive categories, the pullback of a copairing is the copairing of pullbacks, and a copairing is an epimorphism if one of its components is an epi, so we can conclude.

$f + g$ being represented by a pullback-stable epi follows from the fact that coproducts are disjoint in an extensive category. Since our initial map is strict, is a pullback-stable epimorphism.

For the last claim, the terminal object in $\text{Cont}(\mathcal{C})$ is the unique map $0 \rightarrow 1$, which is clearly not always¹³ a (pullback-stable) epimorphism, e.g., in Set or $\text{Asm}(\mathbb{A}, \mathbb{A}')$. $\square$

Corollary 4.1.9. ¹⁴ *The Weihrauch degrees form a distributive lattice which has a bottom element.*

¹³ If you have an extensive category with a zero object, such as the one object category 1 , or the category with two objects and two mutually inverse non-identity morphisms $\square \rightrightarrows \triangle$, then it can be answerable.

¹⁴ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’, Theorem 3.9.

Proof. Immediate by Theorem 3.5.3. Let $f : \subseteq X \rightrightarrows Y, g : \subseteq Z \rightrightarrows W$ be Weihrauch problems. We can give explicit definitions of these operations as follows:

$$\begin{aligned}
\sqcap &: \subseteq X \sqcup Z \Rightarrow Y \times W & \sqcap &: \subseteq X \sqcup Z \Rightarrow Y \sqcup W \\
\text{dom}(p \sqcap q) &:= \text{dom}(f) \times \text{dom}(g) & \text{dom}(f \sqcup g) &:= \text{dom}(f) \sqcup \text{dom}(g) \\
(f \sqcap g)(x, z) &:= f(x) \sqcup g(z) & (f \sqcup g)(0, x) &:= \{0\} \times f(x) \\
(f \sqcup g)(1, z) &:= \{1\} \times g(z) & 0_W &: \emptyset \rightarrow \emptyset \\
0_W &:= w(\text{id}_0) = \emptyset & & \square
\end{aligned}$$

Intuitively, the Weihrauch degree 0_W is the class of nowhere defined problems; the meet $p \sqcap q$ of problems allows you to ask a p -question and a q -question, and get an answer to *one* of them¹⁵; and $p \sqcup q$ allows you to ask either a p -question or a q -question and get an answer to that question.

Remark 4.1.10. *One difference between the Weihrauch degrees and the extended degrees is that the extended degrees have a top element. In the extended degrees, this part can be played by $w(!_0)$. For the Weihrauch lattice, Brattka & Pauly¹⁶ point out that the natural top element would be the class of problems without realisers, which is inhabited only if the axiom of choice does not hold.¹⁷*

To be more concrete, if the Weihrauch lattice had a top element, then there would need to be a weakly terminal¹⁸ container $f : X \rightarrow I$ over represented spaces. In particular, there is a container morphism, $(\varphi, \psi) : \mathcal{I} \rightarrow f$, whose forward map $\varphi : 1 \rightarrow I$ witnesses that I contains computable points. Let I^{rec} be the set of computable elements of I . Then for any family $(x_i \in f^{-1}(i))_{i \in I^{\text{rec}}}$, there is an α such that $x_i \leq_{\mathcal{I}} \alpha$ for all i ; simply take the Turing jump of a problem obtained by combining the countably many sequences x_i into a single sequence. But this contradicts the existence of a container morphism (φ, ψ) from the unique container $\{\alpha\} \rightarrow 1$ to f , as ψ would not have a computable realiser.

4.1.2 Parallel Product

ANOTHER NATURAL MONOIDAL PRODUCT ON CONTAINERS is given by the “parallel product”. Regarded as a Weihrauch problem, this allows us to combine the act of asking a question to f and a question to g , into asking one compound question to both f and g simultaneously, and get back answers for both.

Definition 4.1.11 (Parallel Product).¹⁹ *Let $\mathcal{C}$ be a category with products and pullbacks, and let $f : X \rightarrow I, g : Y \rightarrow J$ be maps. The parallel product of f and g , denoted²⁰ $f \otimes g$, is the map $f \times g$ in $\mathcal{C}$.*

$\otimes$ is clearly a bifunctor. The following proposition is straightforward and its proof is omitted.

Proposition 4.1.12. *Let $\mathcal{C}$ be a category with finite products and pullbacks. Then $\text{Cont}(\mathcal{C})$ is a symmetric monoidal category with product $\otimes$ and unit $\mathcal{I} := \text{id}_1$. It is strictly monoidal whenever $(\mathcal{C}, \times, 1)$ is. Moreover, answerable containers are stable under $\otimes$ and $\mathcal{I}$ is answerable.*

If we further assume that $\mathcal{C}$ is lex extensive, we can show that $\otimes$ satisfies distributive laws.

¹⁵ You know which question was answered.

¹⁶ Brattka and Pauly, ‘On the algebraic structure of Weihrauch degrees’, p. 5.

¹⁷ Note that they use Definition 3.2.8 as their definition of reducibility, which is not necessarily equivalent to the others in Section 3.3 in the absence of the axiom of choice.

¹⁸ A weakly terminal object in a category $\mathcal{C}$ is an object $1 \in \mathcal{C}$ such that for any object X there exists a map $X \rightarrow 1$.

¹⁹ See Spivak, *A reference for categorical structures on Poly*, (6) for the Set case.

²⁰ As we already saw in Proposition 4.1.6, the cartesian product of f and g in $\text{Cont}(\mathcal{C})$ is $[f \times \text{id}_J, \text{id}_I \times g]$, not $f \times g$.

Proposition 4.1.13. ²¹ *If $\mathcal{C}$ is lexensive, then we have natural transformations $f \otimes g + f \otimes h \cong f \otimes (g + h)$ and $(f \otimes g) \times h \rightarrow f \otimes (g \times h)$ in $\text{Cont}(\mathcal{C})$.*

²¹ See [ibid.](#), (6) & (28) for the Set case.

Proof. The proof that $\otimes$ distributes over $+$ is essentially the same as the proof that $\times$ distributes, so we omit it.

For the second claim, assuming that we have $f : X \rightarrow I$, $g : Y \rightarrow J$ and $h : Z \rightarrow K$, the natural transformation is represented by the vertical map (id, c) .

$$\begin{aligned} c : (X \times Y \times K) + (X \times J \times Z) &\rightarrow (X \times Y \times K) + (I \times J \times Z) \\ c &= \text{id}_{X \times Y \times K} + (f \times \text{id}_{J \times Z}) \end{aligned}$$

Note that (id, c) is a valid morphism representative only because the domain of c is isomorphic to $X \times (Y \times K + J \times Z)$; this is true because $\mathcal{C}$ is distributive. $\square$

Now that we have established the categorical properties of $\otimes$, we can relate it to the corresponding product on Weihrauch Degrees.

Definition 4.1.14 (Product of Problems). *Let $f : \subseteq X \rightrightarrows Y$, $g : \subseteq Z \rightrightarrows W$ be Weihrauch problems. The parallel product²² $f \times g$ is the Weihrauch problem defined as follows:*

$$\begin{aligned} f \times g &: \subseteq X \times Z \rightrightarrows Y \times W \\ \text{dom}(f \times g) &:= \text{dom}(f) \times \text{dom}(g) \\ (f \times g)(x, z) &:= f(x) \times g(z) \end{aligned}$$

²² The notation $f \times g$ is standard in the Weihrauch literature for parallel product of problems and corresponds to the categorical tensor. It is important not to confuse it with the categorical product $f \times g$ which corresponds to the meet $f \sqcap g$ of problems.

Proposition 4.1.15. *Let $f : X \rightarrow I$, $g : Y \rightarrow J \in \text{Wei}$. Then*

$$\mathbf{w}(f) \times \mathbf{w}(g) \equiv_W \mathbf{w}(f \otimes g).$$

Proof. Recall that by [Convention 2.4.7](#), we can assume that all projectives are subsets of $\mathbb{B}$ with trivial realisability relations.

$$\begin{aligned} &((p, p'), (q, q')) \in \mathbf{w}(f) \times \mathbf{w}(g) \\ \iff &(p, q) \in \mathbf{w}(f) \text{ and } (p', q') \in \mathbf{w}(g) \\ \iff &\exists x, x' \in \mathbb{B}. p \Vdash f(x), p' \Vdash g(x'), q \Vdash x, q' \Vdash x' \\ \iff &((p, p'), (q, q')) \in \mathbf{w}(f \otimes g). \end{aligned} \quad \square$$

With this proposition established, it is easy to establish that $\times$ is monotone in both arguments and is a commutative monoid on degrees.

Proposition 4.1.16. *Let f, f', g, g', h be Weihrauch problems. Then*

- $f \leq_W f'$ and $g \leq_W g'$ implies $f \times g \leq_W f' \times g'$;
- $f \times (g \times h) \equiv_W (f \times g) \times h$;
- $f \times g \equiv_W g \times f$;
- $f \times \text{id} \equiv_W f$; and
- $(f \times g) \sqcap h \leq_W f \times (g \sqcap h)$.

Proof. We will only prove monotonicity, as the method is the same for each: pass from problems to containers, invoke the appropriate property of symmetric monoidal categories, then return to problems.

From [Theorem 3.5.3](#), if $f \leq_{\mathbb{W}} f'$ (resp $g \leq_{\mathbb{W}} g'$) then there is a container morphism $c(f) \rightarrow c(f')$ (resp $c(g) \rightarrow c(g')$). Since $\otimes$ is a bifunctor, we obtain a container morphism $c(f) \otimes c(g) \rightarrow c(f') \otimes c(g')$. Invoking [Theorem 3.5.3](#) again, we obtain a Weihrauch reduction $w(c(f) \otimes c(g)) \leq_{\mathbb{W}} w(c(f') \otimes c(g'))$. But $w(c(f) \otimes c(g)) \equiv_{\mathbb{W}} w(c(f)) \times w(c(g)) \equiv_{\mathbb{W}} f \times g$ by [Proposition 4.1.15](#), and it follows that $f \times g \leq_{\mathbb{W}} f' \times g'$. $\square$

Remark 4.1.17. Combining [Proposition 4.1.16](#) with [Corollary 4.1.9](#), we recover all the axioms listed in Neumann et al²³.

²³ Neumann, Pauly and Pradic, ‘The equational theory of the Weihrauch lattice with multiplication’.

4.2 Composition Product

THE LAST MONOIDAL PRODUCT we consider is the *composition product*, which differs from the previous products by being non-symmetric. The composition product is mostly easily defined when working over locally cartesian closed categories, where it corresponds to composition of polynomial functors. We will go through the construction in this setting before turning to the category of projectives, which we have already seen is only a *weak* LCC ([Corollary 2.4.19](#)). The weak composition $\star_p$ we define will correspond to the composition of Weihrauch degrees.

4.2.1 Composition in LCCCs

FOR THIS SUBSECTION, we will assume $\mathcal{C}$ is locally cartesian closed, i.e., for any map $f : X \rightarrow Y \in \mathcal{C}$, the pullback functor $f^*(-) : \mathcal{C}/Y \rightarrow \mathcal{C}/X$ has a left adjoint Σ_f and a right adjoint Π_f . The first important consequence of this assumption is that each container gives rise to a container functor.

Definition 4.2.1 (Container functor). ²⁴ Let $f : X \rightarrow I$ be a container. The corresponding container functor $\llbracket f \rrbracket : \mathcal{C} \rightarrow \mathcal{C}$ is defined by

²⁴ This just specialises [Definition 3.4.6](#) with $I = J = 1$.

$$\llbracket f \rrbracket = \text{dom} \circ \sum_{!I} \circ \prod_f \circ !X^* \circ \text{dom}^{-1},$$

where $!I : I \rightarrow 1$, $!X : X \rightarrow 1$ are the unique such morphisms, and $\text{dom} : \mathcal{C}/1 \cong \mathcal{C}$ is the isomorphism sending such maps to their domain.

Remark 4.2.2. In the internal language of $\mathcal{C}$, containers $f : X \rightarrow I$ are written as $(X_i)_{i \in I}$, in which case, the container functor $\llbracket f \rrbracket$ is given by $\llbracket f \rrbracket(Y) = \sum_{i \in I} Y^{X_i}$.

It is clear that any two container functors compose, but what is not so obvious is that their composition is again a container functor. To see what the composition should be, it helps to calculate a little in the internal language. Let’s fix containers $f : X \rightarrow I$ and $g : Y \rightarrow J$, then:

$$\begin{aligned}
\llbracket g \rrbracket (\llbracket f \rrbracket (Z)) &= \llbracket g \rrbracket \left(\sum_{i \in I} Z^{X_i} \right) \\
&= \sum_{j \in J} \left(\sum_{i \in I} Z^{X_i} \right)^{Y_j} \\
&= \sum_{j \in J} \sum_{h: Y_j \rightarrow I} \prod_{y: Y_j} Z^{X_{h(y)}} \\
&= \sum_{(j, h): \sum_{j \in J} I^{Y_j}} Z^{\sum_{y \in Y_j} X_{h(y)}}
\end{aligned}$$

We can see that this is also a polynomial endofunctor, and we can define the composition of f and g to be the container corresponding to the family $\left(\sum_{y \in Y_j} X_{h(y)} \right)_{(j, h): \sum_{j \in J} I^{Y_j}}$ in the internal language.

Definition 4.2.3 (Composition of Containers).²⁵ Suppose we have containers $f : X \rightarrow I$ and $g : Y \rightarrow J$. We define the composition of containers, $f \star g$, as the composition of the *dashed* morphisms in the diagram below²⁶:

$$\begin{array}{ccccc}
& & f \star g & & \\
& \swarrow & & \searrow & \\
\sum_{j \in J} \sum_{y \in Y_j} \sum_{h \in I^{Y_j}} X_{h(y)} & \dashrightarrow & \sum_{j \in J} Y_j \times I^{Y_j} & \equiv & \sum_{j \in J} Y_j \times I^{Y_j} \dashrightarrow \sum_{j \in J} I^{Y_j} \\
\downarrow & \lrcorner & \downarrow \epsilon_{\pi_Y} & & \downarrow \lrcorner \\
X \times Y & \xrightarrow{f \times id_Y} & I \times Y & & \\
\downarrow \pi_X & & \downarrow \pi_I & \searrow \pi_Y & \downarrow \Pi_g(\pi_Y) \\
X & \xrightarrow{f} & I & & Y \xrightarrow{g} J
\end{array}$$

where ϵ_{π_Y} is the π_Y -component of the counit of the adjunction $g^* \dashv \Pi_g$.

Proposition 4.2.4.^{27,28} If $f : X \rightarrow I$ and $g : Y \rightarrow J$ are containers, then there is a natural isomorphism $\llbracket f \star g \rrbracket \cong \llbracket g \rrbracket \circ \llbracket f \rrbracket$.

Corollary 4.2.5. $(\text{Cont}(\mathcal{C}), \star, \mathcal{I})$ is a (non-symmetric) monoidal category.

To continue our dictionary relating containers to Weihrauch reducibility, we need a similar notion of composition for Weihrauch degrees²⁹.

Definition 4.2.6 (Composition of Problems).³⁰ Let f, g be Weihrauch problems. We will define the compositional (or sequential) product, $f \star g$, of problems by

$$\begin{aligned}
f \star g &: \subseteq \mathbb{B} \Rightarrow \mathbb{B} \\
\text{dom}(f \star g) &:= \{ \langle x, e \rangle \mid x \in \text{dom}(g), \Phi(e, y) \in \text{dom}(f) \text{ for all } y \in g(x) \} \\
(f \star g) \langle x, e \rangle &:= \{ \langle y, z \rangle \mid y \in g(x) \text{ and } z \in f(\Phi(e, y)) \}
\end{aligned}$$

where $\Phi(e, y)$ runs the universal type-2 machine Φ on code e with argument y .

²⁵ Gambino and Kock, ‘Polynomial functors and polynomial monads’, ¶1.11.

²⁶ The names of objects on the top row are chosen according to their definition in type theory.

²⁷ *ibid.*, Proposition 1.12.

²⁸ Niu and Spivak, *Polynomial Functors: A Mathematical Theory of Interaction*, Proposition 6.2.

²⁹ The following definition is not the original given by Brattka & Pauly (Brattka and Pauly, ‘On the algebraic structure of Weihrauch degrees’), but it does correspond to the same Weihrauch degree. See the discussion in Westrick (Westrick, ‘A note on the diamond operator’).

³⁰ *ibid.*, Definition 3.

Intuitively, inputs to the sequential product $f \star g$ are f -questions and a strategy for asking g -questions based on f -answers, and outputs are an answer f' to the f -question and an answer to the g -question your strategy would ask in response to f' .

For example, I might ask you “What is your favourite fruit, and what is the best cultivar of that fruit?”. The first question is “What is your favourite fruit?”, and the second question “What is the best cultivar of that fruit?” is dependent upon the answer to the first question for its meaning. Any fruit, e.g., an apple, is an appropriate answer to the first question, however valid answers to the second question depend on what fruit is given as the answer to the first question: “Granny Smith”³¹, “Honeycrisp” and “McIntosh” are appropriate answers in this case, whereas “Beth”, “Concorde”, and “Red Bartlett” are not (being types of pear).

The result we would like to prove is that $w(f \star g) \equiv_W w(f) \star w(g)$ for any answerable containers f and g . However, we have already seen in [Proposition 2.4.10](#) that the category of projectives $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ is not locally cartesian closed, so [Definition 4.2.3](#) cannot be used. In the next section, we will instead define a weak form of composition of containers using the weak LCCC structure from [Corollary 2.4.19](#). This will be sufficient for us to recover the composition of Weihrauch degrees.

³¹ The author’s favourite kind of apple.

4.2.2 Composition for Containers on Projectives

WHILE $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ IS NOT locally cartesian closed, [Lemma 2.4.6](#) tells us that we are working with the full subcategory of projectives of a locally cartesian closed category, and that category has enough projectives. Our aim in this section is to show that these two properties are sufficient to define a sequential product $\star_p$ on $\text{Cont}(p\mathcal{C})$. We start by making the following definition.

Definition 4.2.7 (Fibrewise Projective). *A morphism $f : Y \rightarrow J$ is fibrewise projective if, for every $g : I \rightarrow J$ with I regular projective, then the domain of the pullback g^*f is also regular projective.*

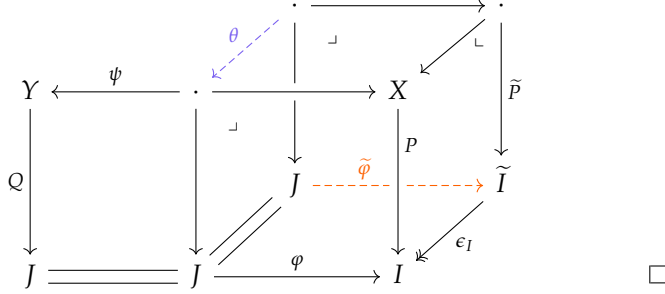
It is easy to see that fibrewise projective morphisms are closed under composition and arbitrary pullbacks³². Let us call $\text{Cont}_{\text{fp}}(\mathcal{C})$ the full subcategory of $\text{Cont}(\mathcal{C})$ whose objects are those containers that are fibrewise projective when regarded as morphisms of $\mathcal{C}$.

Lemma 4.2.8. *Suppose that $p\mathcal{C}$ has finite limits which are preserved by the inclusion $p\mathcal{C} \subseteq \mathcal{C}$. If $\mathcal{C}$ has enough regular projectives, the functor induced by the inclusion $\text{Cont}(p\mathcal{C}) \subseteq \text{Cont}_{\text{fp}}(\mathcal{C})$ has a weak right adjoint $f \mapsto \tilde{f}$ with $f = \tilde{f}$ for any object f of $\text{Cont}(p\mathcal{C})$.*

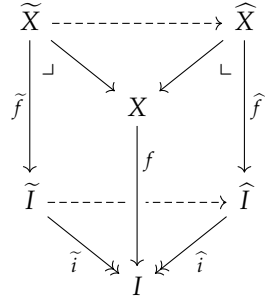
Proof. For any $f : X \rightarrow I$, we can obtain $\tilde{f}$ by pulling back along a regular projective cover $\tilde{I} \twoheadrightarrow I$; since we assume f was fibrewise projective, this ensures $\tilde{f}$ belongs to Cont_{fp} . The pullback itself gives us an horizontal morphism $\varepsilon_f : \tilde{f} \rightarrow f$ in Cont ; much like in the proof of [Lemma 2.4.16](#), this gives us a counit for a weak right adjoint, and its morphism part is obtained by postcomposing ε_f . This function is surjective because, given

³² It appears that these properties, alongside the weak right adjoint below, makes $\text{Cont}_{\text{fp}}(\mathcal{C})$ into a π -clan (Hua and Xu, *Polynomial functors in π -clans for the semantics of type theory*). π -clans are a setting proposed by Joyal (Joyal, *Notes on Clans and Tribes*) for generalising polynomial functors. We have not explored this further, but we thank Steve Awodey for bringing these to our attention.

a morphism (φ, ψ) from g to f , we can obtain a morphism $(\tilde{\varphi}, \psi \circ \theta)$ from g to $\tilde{f}$ as shown in the commuting diagram below. A morphism $\tilde{\varphi}$ such that $\epsilon_U \circ \tilde{\varphi} = \varphi$ exists because V is projective, and θ is defined using the universal property of the pullback of P along φ .



Remark 4.2.9. Suppose we have a fibrewise projective container $f : X \rightarrow I$ and I has two projective covers $\tilde{i} : \tilde{I} \twoheadrightarrow I$ and $\hat{i} : \hat{I} \twoheadrightarrow I$. Then there exists arrows $\tilde{I} \rightarrow \hat{I}$ and $\tilde{X} \rightarrow \hat{X}$ making the diagram below commute. As such, there is a container morphism $\tilde{f} \rightarrow \hat{f}$, and vice versa.



Now, assuming that $\mathcal{C}$ is locally cartesian closed, and $p\mathcal{C}$ satisfies the hypotheses of [Lemma 4.2.8](#), the sequential product $\star$ actually restricts to a functor $\text{Cont}(p\mathcal{C}) \times \text{Cont}(p\mathcal{C}) \rightarrow \text{Cont}_{\text{fp}}(\mathcal{C})$; this is because the morphism representing $P \star Q$ is the composition of two maps obtained by pullback from P and Q (see [Definition 4.2.3](#)), which when combined with the weak right adjoint from [Lemma 4.2.8](#) gives the following definition.

Definition 4.2.10 ($\star_p$). Let $\mathcal{C}$ be a locally cartesian closed category with enough projectives, such that $p\mathcal{C}$ has finite limits preserved by the inclusion $p\mathcal{C} \subseteq \mathcal{C}$, and fix a choice of projective covers. Then there is a quasi-functor $\star_p$ on $\text{Cont}(p\mathcal{C})$ defined by:

$$\begin{aligned} - \star_p - &: \text{Cont}(p\mathcal{C}) \times \text{Cont}(p\mathcal{C}) \rightsquigarrow \text{Cont}(p\mathcal{C}) \\ f \star_p g &:= \widetilde{f \star g}. \end{aligned}$$

This induces a monotone operator on $\text{Cont}/\leftrightarrow(p\mathcal{C})$:

$$[f] \star_p [g] := [f \star_p g].$$

Remark 4.2.11. While the precise definition of $- \star_p -$ (up to isomorphism³³) does depend on the choice of projective covers, [Remark 4.2.9](#) guarantees that the operator on $\text{Cont}(p\mathcal{C})/\leftrightarrow$ does not depend on that choice.

³³ Up to equality, it depends also on a choice of pullbacks.

Proposition 4.2.12. Let $f : X \rightarrow I, g : Y \rightarrow J \in \text{Wei}$. We have

$$w(f \star_p g) \equiv_W w(f) \star w(g).$$

Proof. First, note that $\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ is locally cartesian closed by [Theorem 2.4.2](#), and that $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ has finite limits which agree with those of $\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ by [Theorem 2.4.8](#).

If we treat f and g as maps in $\text{pAsm}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$, then we can use the composition from [Definition 4.2.3](#).

$$f \star g : \sum_{j \in I} \sum_{h \in I^{Y_j}} \sum_{y \in Y_j} X_{h(y)} \rightarrow \sum_{j \in I} I^{Y_j}$$

$$(f \star g)(j, h, y, x) := (j, h)$$

Its codomain has the realisability relation $(e, t) \Vdash_{\text{cod}(f \star g)} (j, h)$ if and only if $e \Vdash_J j$ and t tracks $h : Y_j \rightarrow I$. As $\text{cod}(f \star g)$ is not necessarily a projective, we want to work with the canonical cover $\widetilde{\text{cod}(f \star g)}$ ([Definition 2.4.5](#)). Then $f \star_p g$ is formed by taking the pullback of $f \star g$ along the quotient $\pi : \widetilde{\text{cod}(f \star g)} \rightarrow \text{cod}(f \star g)$. To summarise, we have³⁴:

$$\text{dom}(f \star_p g) := \sum_{j \in I} \sum_{y \in Y_j} \sum_{(h, t) \in \widetilde{I^j}} \sum_{x \in X_{h(y)}} \{e \in \mathbb{B} \mid e \Vdash_J j\}$$

$$(e, a, t, b) \Vdash_{\text{dom}(f \star_p g)} (j, y, h, t, x, e) \text{ iff } (e, a, t, b) \Vdash_{\text{cod}(f \star g)} (j, y, h, x)$$

$$\text{cod}(f \star_p g) := \widetilde{\text{cod}(f \star g)}$$

$$(f \star_p g)(j, y, h, t, x, e) := (j, h, e, t)$$

³⁴ There is a small subtlety in that I_j^Y is a regular projective which is not a subspace of $\mathbb{B}$, but it is isomorphic to one if we use the canonical cover $\widetilde{I_j^Y}$ and expose the trackers.

Before proving the required Weihrauch reductions, it is useful to unpack the definitions of the problems $w(f \star_p g)$ and $w(f) \star w(g)$.

First, we have that $(p, q) \in w(f \star_p g)$ iff $p = \langle e, t \rangle$, $q = \langle e, a, t, b \rangle$, and there exist $j \in J, y \in g^{-1}(j), h : g^{-1}(j) \rightarrow I, x \in f^{-1}(h(y))$ such that $e \Vdash_J j, a \Vdash_Y y, b \Vdash_X x$, and t tracks h .

Next, $(\langle p, e \rangle, \langle q, r \rangle) \in w(f) \star w(g)$ if and only if there exist $x \in |X|, y \in |Y|$, such that $p \Vdash_J g(y), q \Vdash_J y, \Phi(e, q) \Vdash_I f(x)$, and $r \Vdash_X x$.

$w(f \star_p g)$ is reducible to $w(f) \star w(g)$ The forward map φ of the reduction is simply the identity map. However, it requires a little checking to see that this does map $\text{dom}(w(f \star_p g))$ to $\text{dom}(w(f) \star w(g))$. Suppose that $\langle p, t \rangle \in \text{dom}(w(f \star_p g))$, then there exists $j \in J, h : g^{-1}(j) \rightarrow I$ such that $p \Vdash_J j$ and t tracks h . So $p \in \text{dom}(w(g))$ and as g is answerable (hence surjective), there is some $y \in Y_j$ such that $e \Vdash_J j = g(y)$. Since t tracks h , $\Phi(t, q) = t \cdot q \Vdash_I h(y)$, and so $\Phi(t, q) \in \text{dom}(w(f))$ for all $(p, q) \in w(g)$.

The backwards map ψ of the reduction is given by $(\langle p, t \rangle, \langle q, r \rangle) \mapsto \langle q, r, p, t \rangle$ with the obvious realiser. To see this is well-defined, suppose that $(\varphi(\langle p, t \rangle), \langle q, r \rangle) = (\langle p, t \rangle, \langle q, r \rangle) \in w(f) \star w(g)$. Then there exists $x \in |X|, y \in |Y|$ such that $p \Vdash_J g(y), q \Vdash_J y, \Phi(e, q) \Vdash_I f(x)$ and $r \Vdash_X x$. There is a map $h : g^{-1}(j) \rightarrow I$ defined by $y \mapsto f(x)$, which is tracked by t since if $\Phi(t, -)$ is defined for all elements of $w(g)(p)$. It follows that $(\langle p, t \rangle, \psi(\varphi(\langle p, t \rangle), \langle q, r \rangle)) \in w(f \star_p g)$.

$w(f) \star w(g)$ is reducible to $w(f \star_p g)$ Again, the forward map of the reduction will be the identity map. The backwards map ψ is defined by $(\langle p, t \rangle, \langle a, b, p, t \rangle) \mapsto \langle a, b \rangle$ with the obvious realiser. $\square$

While the previous proposition shows that [Definition 4.2.10](#) does give a satisfactory notion for composition of projective containers, it remains to be seen whether we can recover the algebraic laws we would expect of composition. In particular, we would like an analogue of [Corollary 4.2.5](#), and for $\star_p$ to left-distribute over $+$ and $\times$.

As an example, we will give a kind of linear distributivity result on the (extended) Weihrauch degrees. This proof will use the theory of polynomial functors and exploit we are working with a full subcategory of modest sets with enough projectives.

Lemma 4.2.13. ^{35,36} *If $\mathcal{C}$ is locally cartesian closed and extensive, then in $\text{Cont}(\mathcal{C})$ we have maps $(f \star g) \times h \rightarrow (f \times h) \star g$.*

Proof. Using the equivalence from [Proposition 4.2.4](#), it suffices to provide a strong³⁷ natural transformation

$$\llbracket (f \star g) \times h \rrbracket \implies \llbracket (f \times h) \star g \rrbracket.$$

Since we have an equivalence of categories, that the cartesian product of (polynomial) functors are computed pointwise in functor categories and that sequential product corresponds to (reverse) composition of polynomial functors, it thus suffices to provide a strong natural transformation

$$(\llbracket g \rrbracket \circ \llbracket f \rrbracket) \times \llbracket h \rrbracket \implies \llbracket g \rrbracket \circ (\llbracket f \rrbracket \times \llbracket h \rrbracket).$$

For this, it is sufficient to provide a transformation which is natural in A and B

$$\llbracket g \rrbracket(A) \times B \longrightarrow \llbracket g \rrbracket(A \times B).$$

But this amounts to showing that $\llbracket g \rrbracket$ is a strong functor, which is the case as it is a polynomial functor. $\square$

Corollary 4.2.14. *$(a \star b) \sqcap c \leq (a \sqcap c) \star b$ is valid in the (extended) Weihrauch degrees.*

Proof. First, note that if we have that $\mathcal{C}$ is lexextensive, has enough projectives and that its projectives are closed under coproduct and pullbacks, then $\times$ is preserved by the embedding $\text{Cont}(p\mathcal{C}) \subseteq \text{Cont}_{\text{fp}}(\mathcal{C})$ preserves the cartesian product and coproducts built in $\text{Cont}(p\mathcal{C})$. As a result, assuming now that $\mathcal{C}$ is also locally cartesian closed, we have that $(f \star_p g) \times h$ is obtained by applying the weak right adjoint to $\text{Cont}(p\mathcal{C}) \subseteq \text{Cont}_{\text{fp}}(\mathcal{C})$ to $(f \star g) \times h$; by definition, it is also the case for $(f \times h) \star_p g$. Hence we can map the morphism $(f \star g) \times h \rightarrow (f \times h) \star g$ through the weak right adjoint and obtain a morphism that witnesses the validity of the inequality in $\text{Cont}(p\mathcal{C})/\leftrightarrow$. The corollary can then be derived since $\text{Asm}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ and $\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ are extensive and locally cartesian closed. $\square$

4.3 Embedding the Medvedev Degrees

WE WILL TAKE a short detour from Weihrauch reducibility to discuss a related notion of reducibility which we can recover using properties of the shape fibration. A mass problem is a set of functions $\mathbb{N} \rightarrow \mathbb{N}$, which

³⁵ See, e.g., Spivak, *A reference for categorical structures on Poly*, (28).

³⁶ This lemma should be thought of as the analogue of strength in this setting.

³⁷ That $\star$ is strong follows from Propositions 1.12 and 1.15 of Gambino & Kock (Gambino and Kock, ‘Polynomial functors and polynomial monads’).

we should think of as the set of solutions to that problem. Medvedev³⁸ introduced the idea of reducing a problem A to a problem B whenever we can uniformly compute a solution to A from a solution to B . We will only scratch the surface of Medvedev reducibility; the interested reader is invited to read Rogers³⁹ or the survey of Hinman⁴⁰ for more detail.

Definition 4.3.1 (Medvedev Reducible). *Let $A, B \subseteq \mathbb{B}$. A Medvedev reduction from A to B is a partial computable function $F : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ with $B \subseteq \text{dom}(F)$ such that $F(B) \subseteq A$.*

A is Medvedev reducible to B , written⁴¹ $A \leq_S B$, if there exists a Medvedev reduction from A to B .

It is clear that Medvedev reductions form a category, in fact, Medvedev reductions are maps in the opposite of $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$. Since every projective is isomorphic to a subset of $\mathbb{B}$ (by [Convention 2.4.7](#)), we make the following definition.

Definition 4.3.2 (Medvedev Category). *The category of Medvedev reductions Med is defined to be $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})^{\text{op}}$.*

The poset of Medvedev Degrees is the posetal reflection of Med .

To relate the category of Medvedev Reductions to the category of Weihrauch reductions, we first note that the shape fibration has both left and right adjoints.

Proposition 4.3.3.⁴² *Let $\mathcal{C}$ be a lexextensive category. There is an adjoint triple*

$$\begin{array}{ccc} & \xrightarrow{K} & \\ & \top & \\ \mathcal{C} & \xleftarrow{\text{shape}} & \text{Cont}(\mathcal{C}) \\ & \top & \\ & \xrightarrow{ID} & \end{array}$$

where K maps object X to the constant container $!_X : 0 \rightarrow X$, and $ID(X) = id_X$. K and ID are fully faithful and shape is essentially surjective.

Proof. Let $f : X \rightarrow I$ be a container and J an object in $\mathcal{C}$.

shape is left adjoint to K Fix $u : I \rightarrow J$ in $\mathcal{C}$. Since $\mathcal{C}$ is lexextensive, initial objects are strict by [Proposition 4.1.5](#). Hence, the pullback of $K(J)$ along u is initial, and there is a unique container $(u, !) : f \rightarrow K(J)$. This construction is clearly a natural isomorphism, and K is fully faithful, since shape is the left inverse of K .

shape is right adjoint to ID Let $u : J \rightarrow I$. Then (u, ψ) is a container morphism if and only if $\psi = u^*f$, so we have an isomorphism. Naturality essentially follows by the pullback lemma, and ID is fully faithful since $\text{shape} \circ ID = \text{id}$.

shape is essentially surjective This is obvious as $\text{shape}(id_X) = X$ for any $X \in \mathcal{C}$. $\square$

Remark 4.3.4. *In locally cartesian closed categories, such as Set , it is possible to define a right adjoint to K by $(B_j)_{j \in I} \mapsto \sum_{j \in I} 0^{B_j}$. In terms of polynomial functors, this corresponds to taking the constant part of a polynomial/“evaluating at 0”⁴³.*

³⁸ Medvedev, ‘Degrees of difficulty of the mass problem’.

³⁹ Rogers, *Theory of Recursive Functions and Effective Computability*, §13.7.

⁴⁰ Hinman, ‘A Survey of Mučnik and Medvedev Degrees’.

⁴¹ The notation $\leq_S$ seems to stem from distinguishing Medvedev (strong) reducibility with Mučnik (weak) reducibility. Happily, this avoids a clash with $\leq_m$ from many-one reducibility, although Mučnik’s $\leq_W$ does conflict with Weihrauch’s $\leq_W$.

⁴² Spivak, *A reference for categorical structures on Poly*, §9.

⁴³ *ibid.*, Section 9.

While [Proposition 4.3.3](#) provides us with two fully faithful functors, only ID produces pullback-stable epimorphisms. Hence, ID is also a *contravariant* functor from Med to Wei . Thus, we recover the following result.

Proposition 4.3.5. ⁴⁴ *There is an order-reversing embedding of the Medvedev degrees into the Weihrauch degrees.*

Proof. Follows by [Theorem 3.5.3](#), using the contravariant functor ID . $\square$

Remark 4.3.6. *Since shape has both a left and right adjoint, it preserves both limits and colimits. In particular, the order-reversing embedding above will send meets (resp. joins) in the Weihrauch degrees to joins (resp. meets) in the Medvedev degrees.*

K fails to be an order-reversing embedding for uninteresting reasons: $K(A)$ is almost never⁴⁵ an answerable container. It is an order-reversing embedding for the partial Weihrauch degrees. A slightly more interesting case is the category-theoretic version of the order-preserving embedding from Brattka and Gherardi^{46,47}. The proof of the proposition below is straightforward, and is omitted.

Proposition 4.3.7. *Let $X \in \mathcal{C}$ be non-empty⁴⁸. There is a faithful functor $\mathcal{C} \rightarrow \text{Cont}(\mathcal{C})$ defined by*

$$\begin{aligned} C_X(A) &:= \pi_X : X \times A \rightarrow X \\ C_X(f : A \rightarrow B) &:= (id_X, id_X \times f) \end{aligned}$$

.

While $C_B(A)$ is the direct translation of Brattka and Gherardi's c_A , that functor fails to be full⁴⁹. Instead, we should view C_1 , which is full, as its counterpart in containers.

Corollary 4.3.8. ⁵⁰ *C_1 gives rise to order embedding of the Medvedev degrees into the Weihrauch degrees.*

⁴⁴ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’, Theorem 9.1 (2).

⁴⁵ The exception being if A is empty.

⁴⁶ Brattka and Gherardi, ‘Weihrauch Degrees, Omniscience Principles and Weak Computability’, Theorem 5.1.

⁴⁷ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’, Theorem 9.1 (1).

⁴⁸ An object in a lexensive category $\mathcal{C}$ is *non-empty* if there is no map $X \rightarrow 0$.

⁴⁹ $C_X(f)$ is always a *vertical* container morphism. Moreover, even if we restrict to the subcategory of vertical morphisms, it is possible the backwards map ψ does not behave uniformly.

Set $\mathcal{C} := \text{Set}$ and $X := \mathbb{2}$, where $\mathbb{2}$ is the set of Booleans. There is a vertical container morphism whose backwards map $\psi := \langle \pi_1, \text{and} \rangle$, which is not of the form $\text{id} \times f$.

Having said this, C_A is an order embedding for categories of projective modest sets.

⁵⁰ *ibid.*, Theorem 9.1 (1).

5 Operators Definable by Fixpoints

UNTIL NOW, we have been primarily interested in binary operators on the Weihrauch lattice and explaining them as posetal reflections of existing structure in $\text{Cont}(\mathcal{C})$. In this chapter, we turn to operations on problems definable by fixpoints of monotone operators and provide an appropriate counterpart for categories of containers.

We first introduce two extensions of the parallel product on degrees and see that these can be defined by fixpoints of monotone operators. We will then introduce fibred polynomial functors (Definition 5.2.1) as a suitable construction on containers to capture these types of monotone operators. Fibred polynomial functors will have not just least and greatest fixpoints (Theorem 5.2.8), but a third fixpoint ζ , which doesn't seem to exist in the literature on fixpoints. ζ seems to be a natural fixpoint in computable analysis, capturing both the infinite parallelisation (Definition 5.3.4) and infinite loop (Definition 5.3.13) operations. Finally, by introducing the answerable part of a container (Definition 5.4.7), we can capture many problems in the Weihrauch literature by closed ζ -expressions (Figure 5.5).

5.1 Extending the Parallel Product

SINCE THE PARALLEL PRODUCT $\times$ of degrees has a unit, we can construct a parallel product of any *fixed size* by iteration. This suggests two straightforward generalisations: the finite parallelisation of a problem f takes finite lists of f -questions and returns the list of f -answers to those questions¹; similarly, the infinite parallelisation of f takes an infinite sequence of f -questions and returns an infinite sequence of f -answers.

Definition 5.1.1 ((In)finite Parallelisation). *Let $f : \subseteq X \rightrightarrows W$ be a problem. The finite parallelisation² f^* of f is defined by:*

$$\begin{aligned} f^* &: \subseteq X^* \rightrightarrows Y^* \\ f^*(n, [x_1, \dots, x_n]) &:= \{n\} \times f(x_1) \times \dots \times f(x_n) \\ \text{dom}(f^*) &:= \text{dom}(f)^* \end{aligned}$$

The infinite parallelisation³ $\hat{f}$ of f is defined by:

$$\begin{aligned} \hat{f} &: \subseteq X^{\mathbb{N}} \rightrightarrows Y^{\mathbb{N}} \\ \hat{f}(x_n)_{n \in \mathbb{N}} &:= f(x_0) \times f(x_1) \times \dots \\ \text{dom}(\hat{f}) &:= \text{dom}(f)^{\mathbb{N}} \end{aligned}$$

¹ The list of answers will be of the same size, and in the same order, as the list of questions.

² Pauly, 'How Incomputable is Finding Nash Equilibria?', Definition 12.

³ Brattka, 'Computability over Topological Structures', Definition 8.1 (5).

It is clear that we have $f \times f \leq_W f^*$ and $f \times f \leq_W \widehat{f}$, but, in general, $f^* \leq_W f \times f$ and $\widehat{f} \leq_W f \times f$ do not hold.

Counterexample 5.1.2. *It is known that LPO is not idempotent ⁴, and hence $\text{LPO} <_W \text{LPO} \times \text{LPO}$. By monotonicity, it follows that $\text{LPO} \times \text{LPO} <_W \text{LPO} \times \text{LPO} \times \text{LPO} <_W \text{LPO}^* <_W \widehat{\text{LPO}}$. The last two inequalities are straightforward consequences of Definition 5.1.1.*

⁴ Brattka and Gherardi, ‘Weihrauch Degrees, Omniscience Principles and Weak Computability’, Corollary 6.7.

Both the finite and infinite parallelisations are definable as particular fixpoints of monotone operators on degrees.

Proposition 5.1.3. *f^* is the least fixpoint of the monotone operator on problems $p \mapsto \text{id} \sqcup f \times p$; $\widehat{f}$ is a fixpoint of the monotone operator $p \mapsto f \times p$.*

Proof. It is easy to see that $f^* \equiv_W \bigsqcup_{n \in \mathbb{N}} \underbrace{f \times \cdots \times f}_{n \text{ times}}$, and that this is a fixpoint. By induction, we have $\underbrace{f \times \cdots \times f}_{n \text{ times}} \leq_W g$ for any fixpoint g of $p \mapsto \text{id} \sqcup f \times p$, and so f^* is the least fixpoint.

The reduction $\widehat{f} \leq_W f \times \widehat{f}$ is witnessed by the maps (Φ, Ψ) below. Both maps are clearly bijections, so this actually gives $\widehat{f} \equiv_W f \times \widehat{f}$.

$$\begin{aligned} \Phi(p) &:= (p(0), \lambda n. p(n+1)) \\ \Psi(q_0, q) &:= \lambda n. \text{if } n = 0 \text{ then } q_0 \text{ else } q(n-1) \end{aligned} \quad \square$$

Remark 5.1.4. *It is important to note that we have not claimed that $\widehat{f}$ is either the least or greatest fixpoint. Indeed, the least fixpoint of $p \mapsto f \times p$ would be the empty relation, and the greatest fixpoint does not exist as a Weihrauch problem — although we will see in Proposition 5.3.5 that this does exist as a (non-answerable) container. The intermediate fixpoint ζ will play an important role going forward.*

It is instructive to note that both of these operations are not just fixpoints of monotone operators, but that the operators themselves are “nice”: they are combinations of basic problems, the algebraic structure from Chapter 4 and variables. In the next section we generalise this to $\text{Cont}(\mathcal{C})$.

5.2 Fixpoints of Polynomial Functors over Containers

DUE TO THE DIFFICULTY of working with the weak composition product of $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$, we will assume that we are working with a sufficiently nice base category, namely that $\mathcal{C}$ is a $\Pi\mathcal{WM}$ -category (Definition A.2.8). In the future, we hope to generalise this work to a suitable notion of weak polynomial functor over weakly locally cartesian closed categories, which would give a more faithful account of the theory. For now, we stick with $\Pi\mathcal{WM}$ -categories, and will use Remark 5.4.9 to show it is applicable to operators on the Weihrauch lattice.

5.2.1 Fibred Polynomial Functors

OUR INTENT IS TO generalise monotone operators with “nice expressions”. While monotone operators on degrees should clearly generalise to

While this thesis has used the terminology of containers throughout, we use the terminology of polynomial functors in this section to stress that we will be working with a base which is locally cartesian closed

endofunctors on $\text{Cont}(\mathcal{C})$, the latter quality is harder to define. We want our class of endofunctors to be closed under algebraic operations like $+$, $\times$ and $\otimes$, but also under taking fixpoints of multi-variable functors.

Given that polynomial functors are closed under many operations, and their fixpoints have been studied, they may initially seem like a promising candidate. However, while $\text{Cont}(\text{Set})$ is cartesian closed, it is not *locally* cartesian closed⁵. As such, we cannot draw on the work of Gambino & Hyland⁶ to establish the existence of fixpoints for polynomial functors on $\text{Cont}(\mathcal{C})$. However, even if we could, this approach would not help us explain the kind of fixpoint ζ used in defining the infinite parallelisation, as it is neither the greatest nor the least fixpoint. As such, we give a different kind of endofunctor on containers that will be able to explain ζ .

Definition 5.2.1 (Fibred Polynomial Functor). *We will say that a functor $F : \text{Cont}(\mathcal{C})^k \rightarrow \text{Cont}(\mathcal{C})$ is fibred polynomial when*

1. (F, F_0) is a fibred functor $\text{shape}^k \rightarrow \text{shape}$ for some uniquely determined $F_0 : \mathcal{C}^k \rightarrow \mathcal{C}$,
2. F_0 is a polynomial functor, and
3. for every object $I = (I_1, \dots, I_k)$ in $\mathcal{C}^k$, the functors on the fibres

$$F_I : \mathcal{C} / I_1 + \dots + I_k \longrightarrow \mathcal{C} / F_0(I)$$

induced by the isomorphism $\mathcal{C} / I_1 + \dots + I_k \cong \mathcal{C}^k / I$, and F are all polynomial functors.

We say that $F : \text{Cont}(\mathcal{C})^k \rightarrow \text{Cont}(\mathcal{C})^m$ is a fibred polynomial functor if every component $\pi_i \circ F$ is a fibred polynomial functor.

This definition may seem ad hoc⁷ and only serves only to help us prove [Theorem 5.2.8](#), however, it is not too difficult to justify. The idea is that when computing on a container, we want the shape of the output to only depend on the shape of the input, and that computing with the fibres behaves “sensibly” with respect to the shapes.

Proposition 5.2.2. *There is a category FibPoly where the objects are tuples of containers and the morphisms are fibred polynomial functors.*

Proof. We define composition of fibred polynomial functors in the obvious way: $(F, F_0) \circ (G, G_0) = (F \circ G, F_0 \circ G_0)$. Since both polynomial and fibred functors are closed under composition, so is FibPoly . Identities are given by (id, id) , since identity functors id are polynomial⁸ and id trivially preserves cartesian morphisms. $\square$

Now we turn to showing which monoidal products from [Chapter 4](#) are also fibred polynomial functors.

Lemma 5.2.3. *Let $\mathcal{C}$ be an extensive category. If P is the pullback of $f : X \rightarrow I$ along $g : Y \rightarrow I$ and Q is the pullback of $h : Z \rightarrow J$ along $j : W \rightarrow J$, then $P + Q$ is the pullback of $f + h$ along $g + j$.*

Proof. Since $\iota_1 \circ g = (g + j) \circ \iota_1$, pulling $f + h$ along either gives an object P' isomorphic to P . Similarly, pulling back along $(g + j) \circ \iota_2$ gives an object Q' isomorphic to Q . By extensivity⁹, $X + Y \cong P' + Q' \cong P + Q$. $\square$

⁵ Altenkirch, Levy and Staton, ‘Higher-Order Containers’.

⁶ Gambino and Hyland, ‘Wellfounded Trees and Dependent Polynomial Functors’.

⁷ Which it is.

⁸ Gambino and Kock, ‘Polynomial functors and polynomial monads’, Example 1.6 (i).

⁹ Carboni, Lack and Walters, ‘Introduction to extensive and distributive categories’, Proposition 2.2.

Lemma 5.2.4. *The functors $\times, +, \otimes : \text{Cont}(\mathcal{C})^2 \rightarrow \text{Cont}(\mathcal{C})$ are fibred polynomial, as are constant functors and projections.*

Proof. These properties follow from basic properties of polynomial functors and extensivity:

- Constant functors are polynomial functors¹⁰, and are fibred, since all container morphisms are mapped to the identity morphism, which is cartesian.
- Let $\pi_i : \text{Cont}(\mathcal{C})^n \rightarrow \text{Cont}(\mathcal{C})$ be the i -th projection functor, given by $\pi(x_1, \dots, x_n) = x_i$ for some $i \in \{1, \dots, n\}$. This is clearly a (multi-variable) polynomial functor, and it preserves fibred functors.
- The functors $+$ and $\otimes$ are clearly polynomial. $\otimes$ preserves cartesian morphisms, since cartesian container morphisms are essentially pullback squares, and both products and pullbacks are limits so they commute up to isomorphism. For $+$, we use Lemma 5.2.3.
- Recall that $P \times Q = [P \times \text{id}, \text{id} \times Q]$. As fibred polynomials are closed under composition, it remains to check that the cotupling of fibred polynomials is fibred polynomial; this will hold by extensivity¹¹. $\square$

Lemma 5.2.5. *The following functor is fibred polynomial:*

$$\begin{array}{ccc} \text{Cont}(\mathcal{C}) & \longrightarrow & \text{Cont}(\mathcal{C}) \\ X & \longmapsto & X \star P \end{array}$$

Proof. Given containers $P : A \rightarrow I$, $Q : B \rightarrow J$, then the composite $Q \star P$ is given by the projection

$$\sum_{i:I} A_i \times \sum_{f:A_i \rightarrow J} B_{f(u)} \longrightarrow \sum_{i:I} (A_i \rightarrow J)$$

assuming we set $A_i = P^{-1}(i)$ and $B_j = Q^{-1}(j)$. The base of the fibred polynomial functor $J \mapsto \sum_{i:I} J^{A_i}$ is clearly polynomial, as are the functors

$$(B_j)_{j:J} \mapsto \left(\sum_{a:A_i} B_{f(a)} \right)_{(i,f):\sum_{i:I} J^{A_i}}.$$

That the functor is fibred follows since if $\phi : P \rightarrow P'$, $\psi : Q \rightarrow Q'$ are cartesian, then so is $\psi \star \phi$ ¹², and id is an isomorphism, hence cartesian. $\square$

Counterexample 5.2.6. *On the other hand, $X \mapsto P \star X$ is not fibred.*

Taking the containers P and Q as above, then the base of such a polynomial functor would need to map I to $\sum_{i:I} J^{A_i}$. This depends not just on I but on the full container $P = \sum_{i:I} A_i$, and such a map may not be defined on all inputs: concretely, take Q to be any container over 0 , and P any container with some A_i non-empty.

Having established these lemmas, we have all the tools we need to give the examples of fibred polynomial functors in the rest of this thesis. For instance, $X \mapsto \mathcal{I} + X \star A$ is seen to be fibred polynomial by combining the above lemmas.

¹⁰ Gambino and Kock, ‘Polynomial functors and polynomial monads’, Example 1.6 (ii).

¹¹ Carboni, Lack and Walters, ‘Introduction to extensive and distributive categories’, Proposition 2.2.

¹² Niu and Spivak, *Polynomial Functors: A Mathematical Theory of Interaction*, Proposition 6.88.

Remark 5.2.7. Unlike the usual polynomial functors over a locally cartesian closed category, fibred polynomial functors do not necessarily have a canonical strength. A counter-example is the constant functor that maps any polynomial to $\mathcal{I}$.

To see this, for any container $P : A \rightarrow I$, we require a container morphism $P \otimes \mathcal{I} \rightarrow \mathcal{I}$. The forwards map can only be the canonical map $! : I \times 1 \rightarrow 1$. Next, we observe that the pullback of $!$ along $\mathcal{I} = \text{id}_1$ is (isomorphic to) I . Then the backwards map must be a map $I \rightarrow A$, which need not exist (say, if $P = 1$).

5.2.2 Existence of Fixpoints

THE MAIN RESULT of this section states that it is possible to lift fixpoints for polynomial functors $\mathcal{C} \rightarrow \mathcal{C}$ to fixpoints of fibred polynomials endofunctors over $\text{Cont}(\mathcal{C})$.

Theorem 5.2.8. If $F : \text{Cont}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$ is a fibred polynomial functor and $\mathcal{C}$ is an lexensive category with dependent $\mathcal{W}$ and $\mathcal{M}$ -types, then

- F has an initial algebra $(\mu F, a_{\mu F})$,
- F has a terminal coalgebra $(\nu F, c_{\nu F})$, and
- F has a (co)algebra $(\zeta F, b_{\zeta F})$ such that $(F_0(\zeta F), F_0(b_{\zeta F}))$ is an initial algebra for F_0 , and such that it induces a final coalgebra for the endofunctor induced by F over $\mathcal{C}/F_0(\zeta F)$.

The basic idea behind the proof of [Theorem 5.2.8](#) is that we may compute a fixpoint for a fibred polynomial endofunctor F by first computing a fixpoint γF_0 for the induced polynomial functor F_0 in the base, and then take a fixpoint of $i^* \circ F_{\gamma F_0}$, where $i : \gamma F_0 \xrightarrow{\sim} F_0(\gamma F_0)$. In both steps, we have complete freedom over which fixpoint we want to compute, so we may use either initial or terminal coalgebras with the outcomes summarised in [Figure 5.1](#). Note that there is a repeated entry in the matrix, due to the fact that taking the initial algebra in the base determines all fixpoints of $i^* \circ F_{\gamma F_0}$ ([Proposition 5.2.13](#)).

base \ total space	initial algebra μ	terminal coalgebra ν
	initial algebra μ	initial algebra μ
terminal coalgebra ν	terminal coalgebra ν	(co)algebra ζ

Figure 5.1: Matrix of the kinds of fixpoints we get from applying the recipe outlined after [Theorem 5.2.8](#) and [Proposition 5.2.13](#).

To aid reading, we break the proof of [Theorem 5.2.8](#) into [Lemmas 5.2.9](#), [5.2.11](#) and [5.2.12](#).

Lemma 5.2.9. There are four natural constructions of fixpoints for a fibred polynomial functor $F : \text{Cont}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$.

Proof. Since $\mathcal{C} \cong \mathcal{C}/1$, the endofunctor $F_0 : \mathcal{C} \rightarrow \mathcal{C}$ has both an initial algebra $(\mu, \text{in} : F_0(\mu) \rightarrow \mu)$ and a terminal coalgebra $(\nu, \text{in} : F_0(\mu) \rightarrow \mu)$. By Lambek's lemma, both in and out are isomorphism. Choosing one, we have a pair $(\alpha, \alpha : F_0(\alpha) \cong \alpha)$.

Similarly, the induced functor $F_\mu : \mathcal{C}/\alpha \rightarrow \mathcal{C}/F_0(\alpha) \cong \mathcal{C}/\alpha$ has both an initial algebra¹³ $(\mu', \text{in}' : F_\alpha(\mu') \rightarrow \mu')$ and a terminal coalgebra $(\nu', \text{out}' : \nu' \rightarrow F_\alpha(\nu'))$ in $\mathcal{C}/\alpha$. $(\alpha, \alpha : F_0(\alpha) \rightarrow \alpha)$. Again, each choice gives a pair $(\beta, \flat : \beta \cong F_\alpha(\alpha))$.

Together, since $F_\beta = \flat^*(F_\alpha(\beta))$, these give a representative $(\alpha, \flat)$ for a container morphism $F(\beta) \rightarrow \beta$, which is an isomorphism by construction.

$$\begin{array}{ccccc} F(\beta)_{\text{pos}} & \xleftarrow[\sim]{\flat} & \beta_{\text{pos}} & \xlongequal{\quad} & \beta_{\text{pos}} \\ \downarrow F(\nu) & & \downarrow & & \downarrow \nu \\ F_0(\alpha) & \xlongequal{\quad} & F_0(\alpha) & \xrightarrow[\sim]{\alpha} & \alpha \end{array} \quad \square$$

¹³ Note that the codomain of F_α is $\mathcal{C}/\alpha$ and not $\mathcal{C}/F_0(\alpha)$ as in [Definition 5.2.1](#); this is more convenient as it saves writing pullbacks of maps along α^{-1} .

Definition 5.2.10 ((α, β) -fixpoint). *Let $\alpha, \beta \in \{\mu, \nu\}$. The (α, β) -fixpoint for F is the fixpoint given by the construction in [Lemma 5.2.9](#), where $\alpha = \mu$ (resp. ν) if for the first choice we take the initial algebra (resp. terminal coalgebra) and $\beta = \mu$ (resp. ν) if for the second choice we take the initial algebra (resp. terminal coalgebra).*

Lemma 5.2.11. *The (μ, ν) -fixpoint is the initial algebra for F .*

Proof. To fix notation, we will call the container $\nu : \nu \rightarrow \mu$ and the container morphism $(\text{in} : F_0(\mu) \rightarrow \mu, \text{out} : \nu \rightarrow F_\mu(\nu)) : F(\nu) \rightarrow \nu$.

Weak initiality Let $(P : A \rightarrow I, (\alpha, \beta) : F(P) \rightarrow P)$ be an F -algebra.

$$\begin{array}{ccccc} F(P)_{\text{pos}} & \xleftarrow{\beta} & B & \xrightarrow{\quad} & A \\ \downarrow F(P) & & \downarrow b \lrcorner & & \downarrow P \\ F_0(I) & \xlongequal{\quad} & F_0(I) & \xrightarrow{\alpha} & I \end{array}$$

Let's now construct a container morphism $\nu \rightarrow P$. Since (μ, in) is an initial F_0 -algebra and α is an F_0 -algebra, there exists a unique F_0 -algebra homomorphism $\langle \alpha \rangle : \mu \rightarrow I$. Pulling back the commuting square for this homomorphism along P , we obtain the cube below. Note that all four vertical faces are pullback squares and $C \rightarrow D$ is an isomorphism.

$$\begin{array}{ccccc} & & D & \xrightarrow{\quad} & A \\ & \nearrow \sim & \downarrow R \lrcorner & & \downarrow P \\ C & \xrightarrow{\quad} & B & \nearrow & \\ \downarrow \lrcorner & & \downarrow & \lrcorner & \\ & \nearrow \text{in} & \mu & \xrightarrow{\langle \alpha \rangle} & I \\ & & \downarrow Q & \searrow \alpha & \\ F_0(\mu) & \xrightarrow{F_0(\langle \alpha \rangle)} & F_0(I) & & \end{array}$$

The rearmost square is a pullback square, which can be viewed as a (representative of a) cartesian morphism between the containers $R : D \rightarrow \mu$ and $P : A \rightarrow I$. Since F is a fibred functor, applying it to that square yields another pullback square, and hence a unique arrow $\gamma : C \rightarrow F(R)_{\text{pos}}$ making the diagram below commute.

$$\begin{array}{ccc}
 C & \xrightarrow{\quad} & B \\
 \downarrow \gamma & & \downarrow \beta \\
 F(R)_{\text{pos}} & \xrightarrow{\quad} & F(P)_{\text{pos}} \\
 \downarrow F(R) & \lrcorner & \downarrow F(P) \\
 F_0(\mu) & \xrightarrow{F_0(\langle \alpha \rangle)} & F_0(I)
 \end{array} \quad Q$$

The arrow $D \cong C \xrightarrow{\gamma} F(R)_{\text{pos}}$ in $\mathcal{C}$ yields a coalgebra $\mathfrak{d} : R \rightarrow F_\mu(R)$ in $\mathcal{C}/\mu$, and hence there is a unique coalgebra morphism $\llbracket \gamma \rrbracket : R \rightarrow \nu$. It follows that we have a container morphism $(\langle \alpha \rangle, \llbracket \gamma \rrbracket) : \nu \rightarrow P$.

Uniqueness We now check that the container morphism $(\langle \alpha \rangle, \llbracket \gamma \rrbracket)$ we built is the unique algebra morphism from $(\nu, (\text{in}, \text{out}))$ to $(P, (\alpha, \beta))$.

The first component of any such algebra morphism is an F_0 -algebra morphism from (μ, in) to (I, α) . Since (μ, in) is initial, that component therefore must be $\langle \alpha \rangle$.

Similarly, the second component must be an F_μ -coalgebra morphism from $(R, \mathfrak{d})$ to (ν, out) in $\mathcal{C}/\mu$. Since the morphisms $C \rightarrow F_\mu(\nu)$ are equal in both diagrams below, it follows that the second component must be equal to $\llbracket \gamma \rrbracket$.

$$\begin{array}{ccccccc}
 & & F(R)_{\text{pos}} & \xleftarrow{\gamma} & C & & \\
 & & \downarrow & & \downarrow \wr & & \\
 F_\mu(\nu) & \xleftarrow{\sim} & \nu & \xleftarrow{\llbracket \gamma \rrbracket} & D & \xrightarrow{\quad} & A \\
 \downarrow F(\nu) & & \downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow P \\
 F_0(\mu) & \xlongequal{\quad} & F_0(\mu) & \xrightarrow{\sim} & \mu & \xlongequal{\quad} & \mu & \xrightarrow{\langle \alpha \rangle} & I
 \end{array}$$

$$\begin{array}{ccccccc}
 & & C & & & & \\
 & & \downarrow \exists! & & \downarrow \exists! & & \\
 F_\mu(\nu) & \xleftarrow{\quad} & F(R)_{\text{pos}} & \xrightarrow{\quad} & F(P)_{\text{pos}} & \xleftarrow{\quad} & B & \xrightarrow{\quad} & A \\
 \downarrow F(\nu) & & \downarrow & \lrcorner & \downarrow F(P) & & \downarrow Q & \lrcorner & \downarrow P \\
 F_0(\mu) & \xlongequal{\quad} & F_0(\mu) & \xrightarrow{F_0(\langle \alpha \rangle)} & F_0(I) & \xlongequal{\quad} & F_0(I) & \xrightarrow{\alpha} & I
 \end{array} \quad \square$$

Lemma 5.2.12. *The (ν, μ) -fixpoint is the terminal coalgebra for F .*

Proof. To fix notation, we will call the container $\mu : \mu \rightarrow \nu$ and the container morphism $(\text{out} : \nu \rightarrow F_0(\nu), \text{in} : F_\nu(\mu) \rightarrow \mu) : \mu \rightarrow F_\nu(\mu)$.

Let $P : A \rightarrow I$ be a container and $(\alpha, \beta) : P \rightarrow F(P)$ be a F -coalgebra in $\text{Cont}(\mathcal{C})$. Then α is an F_0 -coalgebra with carrier I , and so there exists a unique F_0 -coalgebra morphism $\llbracket \alpha \rrbracket : I \rightarrow \nu$.

Next, observe that β is an $(\alpha^* \circ F_I)$ -algebra in $\mathcal{C}/I$. If $\llbracket \alpha \rrbracket^* \mu$ is the carrier for the initial- $(\alpha^* \circ F_I)$ -algebra, then there exists a unique algebra morphism $\langle \beta \rangle : \llbracket \alpha \rrbracket^* \mu \rightarrow \beta$, and $(\llbracket \alpha \rrbracket, \langle \beta \rangle) : P \rightarrow \mu$ is the required container morphism.

Using that $\llbracket \alpha \rrbracket$ is a coalgebra morphism and F is a fibred functor, we have the following natural isomorphisms.

$$\begin{aligned} \alpha^* \circ F_I \circ \llbracket \alpha \rrbracket^* &\cong \alpha^* \circ F_0(\llbracket \alpha \rrbracket)^* \circ F_V \\ &\cong (F_0(\llbracket \alpha \rrbracket) \circ \alpha)^* \circ F_V \\ &\cong (\text{out} \circ \llbracket \alpha \rrbracket)^* \circ F_V \\ &\cong \llbracket \alpha \rrbracket^* \circ \text{out}^* \circ F_V \end{aligned}$$

As such, $\llbracket \alpha \rrbracket$ is a companion for $\alpha^* \circ F_I$ and $\text{out}^* \circ F_V$ and so $\llbracket \alpha \rrbracket^* \mu$ is an initial $\alpha^* \circ F_I$ algebra by [Lemma A.2.13](#).

It is not difficult to see that being an F -coalgebra amounts to the first component being an F_0 -coalgebra morphism and the second component an $(\alpha^* \circ F_I)$ -algebra morphism. From this observation it follows that $(\llbracket \alpha \rrbracket, \llbracket \beta \rrbracket)$ is the unique F -coalgebra morphism $P \rightarrow \mu$. $\square$

While [Theorem 5.2.8](#) claims only three fixpoints, [Lemma 5.2.9](#) gives four constructions. We will rectify this discrepancy by showing that the (μ, ν) -fixpoint is also initial.

Proposition 5.2.13. *If (F_0, F) is a fibred polynomial functor over a $\Pi\mathcal{WM}$ -category $\mathcal{C}$, $\alpha : F_0(\mu F_0) \rightarrow \mu F_0$ is an initial algebra for F_0 , and there exist $S_0, S_1 \in \mathcal{C} / \mu F_0$ with $F_{\mu F_0}(S_i) \cong \alpha \circ S_i$ for $i \in \{0, 1\}$, then $S_0 \cong S_1$.*

Proof. Let us reason informally in the internal language of ELCCCS with $\mathcal{W}$ -types, i.e., EMLTT augmented with $\mathcal{W}$ -types. Within EMLTT, for any families A and B , we can define a type $A \cong B$ that is inhabited whenever A and B are isomorphic, a fortiori with $\prod_{x:\mu F_0} S_0(x) \cong S_1(x)$ holding if and only if $S_0 \cong S_1$. So as to exploit the internal language, let us assume that

$$F_0(X) = \prod_{i:I} X^{A_i} \quad \text{and} \quad F((Y_x)_{x:X}) = \left(\sum_{j:I_i, f} \prod_{x:X} B_{i,f,j,x} \rightarrow Y_x \right)_{(i,f):F_0(X)}.$$

for a suitable type I and families $(A_i)_{i:I}$, $(J_{i,f})_{(i,f):F_0(X)}$ and $(B_{i,f,j,x})_{j:I_i, f, x:X}$. Recall μF_0 is a $\mathcal{W}$ -type with constructor α , so we may proceed to show $\prod_{x:\mu F_0} S_0(x) \cong S_1(x)$ is inhabited by recursion over x . For this, it suffices to build a function in $S_0(\alpha(i, f)) \cong S_1(\alpha(i, f))$ assuming $i : I, f : A_i \rightarrow \mu F_0$ and that we inductively have an inhabitant of

$$\prod_{a:A_i} S_0(f(a)) \cong S_1(f(a)).$$

So let us do half of that and conclude by symmetry.

$$\begin{aligned} S_0(\alpha(i, f)) &\cong F_{\mu F_0}(S_0)(i, f) && \text{since } F_{\mu F_0}(S_0) \cong \alpha^* S_0 \\ &\cong \sum_{j:I_i} \prod_{x:\mu F_0} B_{i,f,j,x} \rightarrow S_0(x) && \text{by definition} \\ &\cong \sum_{j:I_i} \prod_{x:\mu F_0} B_{i,f,j,x} \rightarrow S_1(x) && \text{by } S_0(x) \cong S_1(x) \end{aligned} \quad \square$$

Let us now look at a couple of basic examples.

Example 5.2.14. *Since the identity functor is fibred polynomial, it has fixpoints. As usual, $\mu(X \mapsto X)$ is an initial object and $\nu(X \mapsto X)$ is a terminal object. The middle fixpoint $\zeta(X \mapsto X)$ is isomorphic to the monoidal unit $\mathcal{I}$ of $\otimes$.*

Example 5.2.15. The functor $X \mapsto 1 + X$ admits the following fixpoints:

- $\mu(X \mapsto 1 + X)$ is the container $0 \rightarrow \mathbb{N}$,
- $\nu(X \mapsto 1 + X)$ is the container $0 \rightarrow \mathbb{N}_\infty$, and
- $\zeta(X \mapsto 1 + X)$ is the container $\infty : 1 \rightarrow \mathbb{N}_\infty$.

Finally, [Theorem 5.2.8](#) also behaves well in presence of parameters.

Theorem 5.2.16. Let $F : \text{Cont}(\mathcal{C})^{k+1} \rightarrow \text{Cont}(\mathcal{C})$ be a fibred polynomial functor. For any $\gamma \in \{\mu, \nu, \zeta\}$, the map

$$\begin{aligned} \text{Obj}(\text{Cont}(\mathcal{C})^k) &\longrightarrow \text{Obj}(\text{Cont}(\mathcal{C})) \\ (f_0, \dots, f_{k-1}) &\longmapsto \gamma F(f_0, \dots, f_{k-1}, -) \end{aligned}$$

extends to a fibred polynomial functor $\gamma_k F : \text{Cont}(\mathcal{C})^k \rightarrow \text{Cont}(\mathcal{C})$.

Proof. The following proof is only for μ , but similar arguments hold for ν and ζ .

Functoriality Since F is fibred polynomial, so is $F(f_0, \dots, f_{k-1}, -)$ for any objects $f_i : X_i \rightarrow I_i$ in $\text{Cont}(\mathcal{C})$, so it is a well-defined map on objects.

Suppose we have objects $g_i : Y_i \rightarrow J_i$ and morphisms (φ_i, ψ_i) in $\text{Cont}(\mathcal{C})$ and let

$$\begin{aligned} \mu F(f_0, \dots, f_{k-1}, -) &: \nu F_{\mu F_0(I_0, \dots, I_{k-1}, -)}(f_0, \dots, f_{k-1}, -) \longrightarrow \mu F_0(I_0, \dots, I_{k-1}, -) \\ \mu F(g_0, \dots, g_{k-1}, -) &: \nu F_{\mu F_0(J_0, \dots, J_{k-1}, -)}(g_0, \dots, g_{k-1}, -) \longrightarrow \mu F_0(J_0, \dots, J_{k-1}, -) \end{aligned}$$

be the respective fixpoints. On the bases, we have two $F_0(I_0, \dots, I_{k-1}, -)$ -algebra morphisms: the initial one, and one obtained from the initial $F_0(J_0, \dots, J_{k-1}, -)$ -algebra by composing with $F_0(\varphi_0, \dots, \varphi_{k-1}, \psi_{k-1}, \text{id})$. Thus we obtain a unique algebra morphism, which we take to be the forward map of $\mu F((\varphi_0, \psi_0), \dots, (\varphi_{k-1}, \psi_{k-1}))$. Similarly, we obtain the backwards map by doing the same thing on the domains, but using that we have terminal coalgebras and the ψ_i . Functoriality of this construction is guaranteed by the uniqueness of the algebra and coalgebra maps.

Fibred polynomial It remains to see that this construction gives a fibred functor which is polynomial on the base and the fibres.

To check that we have a morphism of fibrations, we define the base functor by

$$(\mu F)_0(I_0, \dots, I_{k-1}) := \mu F_0(I_0, \dots, I_{k-1}, -).$$

Then $(\mu F)_0 \circ \text{shape} = \text{shape} \circ \mu F$ holds by construction. $(\mu F)_0$ is a “fix-point functor” of the polynomial functor F_0 and so is itself polynomial¹⁴. To see that μF preserves cartesian morphisms, recall that cartesian morphisms (φ, ψ) are those for which ψ is an isomorphism. In this case, the unique coalgebra morphism is an isomorphism, since we can do the same construction in the opposite direction.

Similarly, $\mu F(X_0, \dots, X_{k-1}, -) = \nu F_{(\mu F)_0}(X_0, \dots, X_{k-1}, -)$ is a fix-point functor, and so is polynomial. $\square$

¹⁴ Gambino and Hyland, ‘Wellfounded Trees and Dependent Polynomial Functors’, Theorem 14.

5.3 Examples of Operators Definable by Fixpoints

IN THIS SECTION, we discuss various endofunctors over $\text{Cont}(\mathcal{C})$ definable using [Theorem 5.2.8](#). In particular, we will address the (in)finite parallelisation, free monad on a polynomial, and a few others summarised in [Figure 5.2](#). Throughout this section we will assume $\mathcal{C}$ to be an $\Pi\mathcal{WM}$ -category.

Notation	Operation on problems	Fixpoint expression
P^*	finite parallelization	$\mu X. \mathcal{I} + P \otimes X$
$\hat{P}$	infinite parallelization	$\zeta X. P \otimes X$
$\tilde{P}$	ask ω questions, get one answer	$\nu X. P \times X$
$P^\diamond$	finite iterations (free monad over a polynomial)	$\mu X. \mathcal{I} + X \star P$
P^∞	ω -loop	$\zeta X. X \star P$

Figure 5.2: Endofunctors $\text{Cont}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$ defined via fixpoint equations. Since the definition of $\star$ requires $\mathcal{C}$ to be a LCCC, the last two definitions are not quite the same as the operators on Weihrauch degrees (see [Section 4.2](#)).

5.3.1 Parallelisations

FIX A CONTAINER $P : A \rightarrow I$ (write $(A_i)_{i:I}$ for the corresponding family) and let us consider fixpoints of the endofunctor $X \mapsto P \times X$ over $\text{Cont}(\mathcal{C})$. We first note that we have the following isomorphisms in $\mathcal{C}$.

$$\begin{aligned} \text{shape}(\mu(X \mapsto P \times X)) &\cong \mu(X \mapsto I \times X) \cong 0 \\ \text{and } \text{shape}(\gamma(X \mapsto P \times X)) &\cong \nu(X \mapsto I \times X) \cong I^{\mathbb{N}} \quad \text{for } \gamma \in \{\nu, \zeta\} \end{aligned}$$

As a result, we have $\mu(X \mapsto P \times X) \cong 0$. The situation with the other fixpoints is slightly more exciting. The greatest fixpoint yields the family $\left(\sum_{n:\mathbb{N}} A_{s(n)} \right)_{s:I^{\mathbb{N}}}$. Interpreted as a transformation of problems, the corresponding operator takes a problem P to the problem $\tilde{P}$ where a question is a sequence of P -questions, and an answer picks an index n and answers the n -th question.

$$\begin{array}{ccc} \widetilde{} : \text{Cont}(\mathcal{C}) & \longrightarrow & \text{Cont}(\mathcal{C}) \\ P & \longmapsto & \nu(X \mapsto P \times X) \end{array}$$

The ζ fixpoint corresponds to the family $\left(\sum_{x:\mathbb{N}_\infty} \prod_{n:\mathbb{N}} (x = \underline{n} \rightarrow A_{s(n)}) \right)_{s:I^{\mathbb{N}}}$.

In terms of problems, an answer now consists of a conatural number and, if this conatural number corresponds to a natural number, then an answer is eventually given for the relevant question.

Proposition 5.3.1. *The ν -fixpoint and ζ -fixpoint of $X \mapsto P \times X$ are respectively (isomorphic to) the following containers:*

$$\sum_{s:I^{\mathbb{N}}} \sum_{n:\mathbb{N}} A_{s(n)} \xrightarrow{\pi_1} I^{\mathbb{N}} \quad \text{and} \quad \sum_{s:I^{\mathbb{N}}} \sum_{x:\mathbb{N}_\infty} \prod_{n:\mathbb{N}} (x = \underline{n} \rightarrow A_{s(n)}) \xrightarrow{\pi_1} I^{\mathbb{N}}$$

Proof. We already argued that the shapes were $I^{\mathbb{N}}$. For this proof, we will work in type theory, representing $P : A \rightarrow I$ by the internal family $(A_i)_{i:\mathbb{N}}$, and write $S : \mathbb{N} \rightarrow \mathbb{N}$ for the successor map. It suffices to show that $(\sum_{n:\mathbb{N}} A_{s(n)})_{s:I^{\mathbb{N}}}$ and $(\sum_{x:\mathbb{N}_{\infty}} \prod_{n:\mathbb{N}} (x = \underline{n} \rightarrow A_{s(n)}))_{s:I^{\mathbb{N}}}$ are respectively carriers for an initial algebra and a terminal coalgebra for the functor $F : \mathcal{C}/I^{\mathbb{N}} \rightarrow \mathcal{C}/I^{\mathbb{N}}$ taking $(B_s)_{s:I^{\mathbb{N}}}$ to $(A_{s(0)} + B_{s \circ S})_{s:I^{\mathbb{N}}}$.

Now $(\sum_{n:\mathbb{N}} A_{s(n)})_{s:I^{\mathbb{N}}}$ is the carrier of an F -algebra whose underlying map is built using the following isomorphism that exists fibrewise in s by extensivity of $\mathcal{C}$:

$$A_{s(0)} + \sum_{n:\mathbb{N}} A_{s(n+1)} \cong \sum_{n:\mathbb{N}} A_{s(n)}$$

To see that it is initial, it will suffice to show that for any F -algebra $\alpha_s : A_{s(0)} + B_{s \circ S} \rightarrow B_s$, there *purely*¹⁵ exists a unique family of maps $f_s : \sum_{n:\mathbb{N}} A_{s(n)} \rightarrow B_s$ such that, for every $n : \mathbb{N}$,

$$f_s(0, a) = \alpha_s(\iota_1(a)) \quad \text{and} \quad f_s(n+1, a) = \alpha_s(\iota_2(f_s(n, a)))$$

This is easy to show by induction over $n : \mathbb{N}$.

Similarly, to check that $(\sum_{x:\mathbb{N}_{\infty}} \prod_{n:\mathbb{N}} (x = \underline{n} \rightarrow A_{s(n)}))_{s:I^{\mathbb{N}}}$ is the carrier of the obvious terminal F -coalgebra, we need to check that for any F -coalgebra $\gamma_s : B_s \rightarrow A_{s(0)} + B_{s \circ S}$, there purely exists a unique $f_s : B_s \rightarrow (\sum_{x:\mathbb{N}_{\infty}} \prod_{n:\mathbb{N}} (x = \underline{n} \rightarrow A_{s(n)}))$ such that, for any $b : B_s$,

- either $\gamma_s(b) = \iota_1(a)$ for some $a : A_{s(0)}$ and $f_s(b) = (\underline{0}, h)$ with $h : \prod_{n:\mathbb{N}} (\underline{0} = \underline{n} \rightarrow A_{s(n)})$ constant and equal to a ;
- or $\gamma_s(b) = \iota_2(b')$ for some $b' : B_{s \circ S}$, $f_{s \circ S}(b') = (x, h \circ S)$ and $f_s(b) = (\underline{S}(x), h)$ for some $x : \mathbb{N}_{\infty}$ and $h : \prod_{n:\mathbb{N}} (\underline{S}(x) = \underline{n} \rightarrow A_{s(n)})$.

This determines the first component $\ell_s := \pi_1 \circ f_s : B_s \rightarrow \mathbb{N}_{\infty}$ as we have

$$\gamma_s(b) = \iota_1(a) \Leftrightarrow \ell_s(b) = \underline{0} \quad \text{and} \quad \gamma_s(b) = \iota_2(b') \Leftrightarrow \ell_s(b) = \underline{S}(\ell_{s \circ S}(b'))$$

which uniquely determines ℓ_s by corecursion over the conatural numbers. Up to currying and swapping arguments, the second component r_s of f_s has type

$$r_s : \prod_{n:\mathbb{N}} \prod_{b:B_s} (\ell_s(b) = \underline{n} \rightarrow A_{s(n)})$$

and is inductively determined by

$$\begin{aligned} r_s(0)(b)(e) &= a & \text{when } \gamma_s(b) = \iota_1(a) \\ r_s(n+1)(b)(e) &= r_s(n)(b')(\text{Sinj}(e)) & \text{when } \gamma_s(b) = \iota_2(b') \end{aligned}$$

where Sinj is the obvious term of type $\underline{S}(\ell_{s \circ S}(b')) = \underline{S}(\underline{n}) \rightarrow \ell_{s \circ S}(b') = \underline{n}$. This can be improved to a recursive definition over n by pattern matching on $\gamma_s(b)$, whose the remaining cases admit absurdity proofs due to e . $\square$

Proposition 5.3.2. *Let $(A_i)_{i:I}$ be a container. There exist container morphisms $P \rightarrow \tilde{P}$ and $\tilde{P} \rightarrow P$.*

Proof. These constructions are most easily done in type theory. The following definitions give container morphisms $(\varphi, \psi) : P \rightarrow \tilde{P}$ and $(\varphi', \psi') : \tilde{P} \rightarrow P$.

¹⁵ In the type theoretic sense (Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*), meaning we can explicitly construct the witness in function of the parameters in type theory.

$$\begin{array}{ll}
\varphi : I \rightarrow I^{\mathbb{N}} & \varphi' : I^{\mathbb{N}} \rightarrow I \\
\varphi(i) = \lambda n. i & \varphi'(s) = s(0) \\
\psi : (i : I) \rightarrow \sum_{n:\mathbb{N}} A_{\varphi(i)(n)} \rightarrow A_i & \psi' : (s : I^{\mathbb{N}}) \rightarrow A_{\varphi(s)} \rightarrow \sum_{n:\mathbb{N}} A_{s(n)} \\
\psi(i, (n, a)) = a & \psi'(i, a) = (0, a) \quad \square
\end{array}$$

Remark 5.3.3. In general, [Proposition 5.3.2](#) cannot be extended to an isomorphism, since an isomorphism $P \cong \tilde{P}$ would imply an isomorphism $I \cong \text{shape}(P) \cong \text{shape}(\tilde{P}) \cong I^{\mathbb{N}}$ since functors preserve isomorphisms. Taking $I = \{0, 1\}$ provides a concrete counterexample in **Set**.

While useful as an example, [Proposition 5.3.2](#) is the container equivalent of the equivalence $P \equiv_{\mathbb{W}} \tilde{P}$ in the Weihrauch degrees¹⁶, meaning that such an operator is redundant in that setting.

We now turn to giving the container equivalent of [Definition 5.1.1](#). For this we naturally consider a fixpoint of the functor $X \mapsto P \otimes X$ induced by the binary parallelisation $\otimes$. Since $X \mapsto P \otimes X$ and $X \mapsto P \times X$ induce the same endofunctor $Y \mapsto I \times Y$ over $\mathcal{C}$, the shapes of the fixpoints are the same and $\mu(X \mapsto P \otimes X) \cong 0$. This time around, the greatest fixpoint is of little interest, but the intermediate fixpoint ζ gets us what we want; we write $\hat{P}$ for the infinite parallelisation of P defined thus.

Definition 5.3.4 (Infinite Parallelisation). Let $\mathcal{C}$ be a $\Pi\mathcal{WM}$ -category. Then the infinite parallelisation functor $\hat{} : \text{Cont}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$ is defined by

$$\hat{P} := \zeta(X \mapsto P \otimes X).$$

Proposition 5.3.5. The ν -fixpoint and ζ -fixpoint of $X \mapsto P \otimes X$ are respectively (isomorphic to) the following containers:

$$0 \longrightarrow I^{\mathbb{N}} \quad \text{and} \quad A^{\mathbb{N}} \xrightarrow{P^{\mathbb{N}}} I^{\mathbb{N}}$$

In particular¹⁷ $\nu(X \mapsto P \otimes X) \cong 1 \star \hat{P}$.

Similarly, we can mimic the operator characterising finite parallelisation from [Proposition 5.1.3](#) to define the finite parallelisation of containers

Definition 5.3.6 (Finite Parallelisation). Let $\mathcal{C}$ be a $\Pi\mathcal{WM}$ -category. Then the finite parallelisation functor $-^* : \text{Cont}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$ is defined by

$$P^* := \mu(X \mapsto \mathcal{I} + P \otimes X).$$

Proposition 5.3.7. P^* is isomorphic to $\sum_{n:\mathbb{N}} \underbrace{P \otimes \dots \otimes P}_{n \text{ times}}$. On the other hand, $\nu(X \mapsto \mathcal{I} + P \otimes X)$ and $\zeta(X \mapsto \mathcal{I} + P \otimes X)$ are respectively (isomorphic to)

$$\begin{array}{lll}
\sum_{n:\mathbb{N}} A^n & \longrightarrow & \sum_{m:\mathbb{N}_{\infty}} \prod_{\substack{k:\mathbb{N} \\ k < m}} I^k \\
(n, a) & \longmapsto & (\underline{n}, P \circ a)
\end{array}
\quad \text{and} \quad
\begin{array}{lll}
\sum_{m:\mathbb{N}_{\infty}} \prod_{\substack{k:\mathbb{N} \\ k < m}} A^k & \longrightarrow & \sum_{m:\mathbb{N}_{\infty}} \prod_{\substack{k:\mathbb{N} \\ k < m}} I^k \\
(x, a) & \longmapsto & (x, P \circ a)
\end{array}$$

¹⁶ At this time, we cannot immediately conclude this from [Proposition 5.3.2](#) as we are assuming we are working in a $\Pi\mathcal{WM}$ -category, and hence an LCCC, but that will be remedied later.

¹⁷ For any P , $1 \star P$ interpreted as a problem is the problem with the same questions as P , but with no answers whatsoever.

5.3.2 Iterations

NOW WE TURN to fixpoints built using the sequential composition of problems. As $\star$ is not commutative there are, a priori, two ways of defining functors whose fixpoints are iterations of some problem P . However, one of these is not fibred (Counterexample 5.2.6) and the other is (Lemma 5.2.5). We will thus mostly focus on the latter, although it is worth noting that $X \mapsto \mathcal{I} + P \star X$ does have a non-trivial initial algebra with a natural interpretation. It corresponds to the problem whose instances (n, f) are given by a number n of successive questions we want to ask to P and then a recipe f to ask those questions, and where results are the answers to n questions we could have successively asked¹⁸.

Proposition 5.3.8. *The carrier of the initial algebra for $X \mapsto \mathcal{I} + P \star X$ is isomorphic to $S = \sum_{n:\mathbb{N}} \underbrace{P \star \dots \star P}_{n \text{ times}}$.*

Proof. It is straightforward to define an algebra map $s : \mathcal{I} + P \star S \rightarrow S$ by using that $\underbrace{P \star \dots \star P}_{0 \text{ times}} \cong I$ and that $\star$ distributes over $+$.

$$\mathcal{I} + P \star \left(\sum_{n:\mathbb{N}} \underbrace{P \star \dots \star P}_{n \text{ times}} \right) \cong \mathcal{I} + \sum_{n:\mathbb{N}} \underbrace{P \star \dots \star P}_{n+1 \text{ times}} \cong \sum_{n:\mathbb{N}} \underbrace{P \star \dots \star P}_{n \text{ times}}$$

Given another algebra $(A, a : \mathcal{I} + P \star A \rightarrow A)$, we can define a family of maps $s_i : S_i \rightarrow A$ by induction, where $S_i = \sum_{n=0}^i \underbrace{P \star \dots \star P}_{n \text{ times}}$. First of

all, we set $s_0 : S_0 \cong \mathcal{I} \xrightarrow{\text{in}_1} \mathcal{I} + P \star A \xrightarrow{a} A$. Then we inductively define $s_{i+1} : S_{i+1} \cong \mathcal{I} + (P \star S_i) \xrightarrow{I + P \star s_i} \mathcal{I} + P \star A \xrightarrow{a} A$. The unique algebra morphism $(S, s) \rightarrow (A, a)$ is given by the coproduct of s_0 and $s_i \circ \text{in}_i : \underbrace{P \star \dots \star P}_{i \text{ times}} \rightarrow A$ for all $i > 0$, and hence (S, s) is an initial algebra. $\square$

On the other hand, the functor $X \mapsto \mathcal{I} + X \star P$ is fibred polynomial by Lemmas 5.2.4 and 5.2.5. So Theorem 5.2.16 says we have a corresponding polynomial endofunctor.

Definition 5.3.9 (Finite Iteration). *Let $\mathcal{C}$ be a $\Pi\mathcal{WM}$ -category. Then the finite iteration functor $-^\diamond : \text{Cont}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$ is defined by*

$$P^\diamond := \mu(X \mapsto \mathcal{I} + X \star P).$$

Regarding $P^\diamond$ as a problem, inputs are recipes for asking finitely many questions sequentially to P and outputs are possible sequences of answers to that. Note that, in general, this is strictly more powerful than the operator from Proposition 5.3.8 as the number of questions isn't fixed in advance.

While finite iterations of problems have been introduced in 2016 to the literature on Weihrauch complexity¹⁹, the endofunctor $(-)^\diamond$ has been known for much longer to category theorists: $\llbracket P^\diamond \rrbracket$ was characterised as the free monad over the polynomial functor $\llbracket P \rrbracket$ ²⁰. The construction given in²¹ corresponds to the (μ, μ) -fixpoint from Theorem 5.2.8. From that point of view, assuming for a moment $P \in \text{Cont}(\text{Set})$, the endofunctor

¹⁸ Brattka (Brattka, 'Loops, Inverse Limits and Non-Determinism') refers to these as "for loops" and notates the operator by $P^{[*]}$.

¹⁹ Neumann and Pauly, 'A topological view on algebraic computation models'.

²⁰ Kelly, 'A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on'.

²¹ Gambino and Kock, 'Polynomial functors and polynomial monads', Section 4.

$\llbracket P^\diamond \rrbracket$ may be described as follows. The shape of $P^\diamond$ consists of well-founded trees labelled by some $i : I$ or by a special symbol $\bullet$, and the arity of the node is $P^{-1}(i)$ or 0 if the special symbol is used. Then $\llbracket P^\diamond \rrbracket(X)$ consists of all those trees from $P^\diamond$ where every leaf ℓ is labelled by some $x_\ell \in X$ (see Figure 5.3).

Remark 5.3.10. While $\text{shape}(P^\diamond)$ can be regarded as a sort of well-founded tree structure in $\text{Cont}(\text{ReprSp})$ (see Figure 5.3), it is important to note that these trees are very different than those we encode in $\mathbb{B}$ (e.g., when defining WKL): any node has potentially continuum-many children, and computing the direction to a child corresponds to the ability to answer the P -question of its label, which is typically impossible to do continuously.

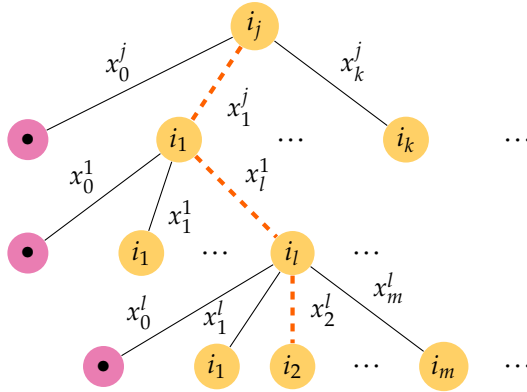


Figure 5.3: Picture of $P^\diamond$ assuming $P : A \rightarrow I$ represents $(A_i)_{i:I}$. Nodes correspond to asking a question $i \in I$ or indicating that no more questions will be asked with $\bullet$. Edges from question nodes i_j to their children are labelled with an answer $x_k^j : A_{i_j}$ to that question. A $P^\diamond$ -question is such a tree and a $P^\diamond$ -answer is a path in the tree (dashed) from the root (i_j).

A key property of $(-)^{\diamond}$ is that it associates well with composition of other problems, which is captured by the following theorem.

Theorem 5.3.11. $Q \star P^\diamond$ is the carrier of an initial algebra of the fibred polynomial endofunctor $X \mapsto Q + X \star P$.

Proof. Let $P : A \rightarrow I$ and $Q : B \rightarrow J$ be containers, and call the functor $X \mapsto Q + X \star P$ above F . In the rest of the proof, we will work in Martin-Löf type theory with $\mathcal{W}$ -types; in this context, let us call $(A_i)_{i:I}$ and $(B_j)_{j:J}$ the respective type families corresponding to P and Q . We take the following isomorphism to be the F -algebra structure on $Q \star P^\diamond$ that we claim to be initial:

$$(\text{in}^\diamond, \text{out}^\diamond) : F(Q \star P^\diamond) = Q + Q \star P^\diamond \star P \cong Q \star (\mathcal{I} + P^\diamond \star P) \cong Q \star P^\diamond$$

Further suppose we have an F -algebra $(R, (\varphi, \psi))$, for some container $R : C \rightarrow K$. In type-theoretic notations, we have

$$\varphi = [\varphi_1, \varphi_2] : J + \sum_{i:I} K^{A_i} \longrightarrow R$$

and the following families induced by ψ .

$$\begin{aligned} & (\psi_j : C_{\varphi_1(j)} \rightarrow B_j)_{j:J} \\ & (\psi_{i,f} : C_{\varphi_2(i,f)} \rightarrow \sum_{a:A_i} C_{f(a)}(i,f) : \sum_{i:I} K^{A_i}) \end{aligned}$$

We will explicitly define the unique algebra morphism $(\varphi^\diamond, \psi^\diamond) : Q \star P^\diamond \rightarrow R$ by constructing the forward and backward maps $\varphi^\diamond$ and $\psi^\diamond$ in type theory with $\mathcal{W}$ -types. For notational convenience, we will write $I^\diamond$ for $\text{shape}(P^\diamond)$ and $(A_i^\diamond)_{i:I^\diamond}$ for the type family corresponding to $P^\diamond$; recall from [Theorem 5.2.8](#) that we have maps $\text{leaf} : 1 \rightarrow I^\diamond$ and $\text{node} : \sum_{i:I^\diamond} (A_i \rightarrow I^\diamond) \rightarrow I^\diamond$ such that $(I^\diamond, [\text{leaf}, \text{node}])$ is the initial algebra of the endofunctor $X \mapsto \mathcal{I} + X \star P$. In what follows, let us bear in mind that we have $\text{shape}(Q \star P^\diamond) \cong \sum_{i:I^\diamond} (A_i^\diamond \rightarrow J)$ and that the direction corresponding to $(i, f) : \text{shape}(Q \star P^\diamond)$ are $\sum_{a:A_i^\diamond} B_{f(a)}$.

We define the maps by well-founded recursion on the first component of the argument $i : I^\diamond$ for both the forward and the backwards map; up to dependent currying, this is a standard dependent recursion. For the forward map, we have:

$$\begin{aligned} \varphi^\diamond : \sum_{i:I^\diamond} (A_i^\diamond \rightarrow J) &\longrightarrow K \\ (\text{leaf}(\bullet), f) &\longmapsto \varphi_1(f(\bullet)) \\ (\text{node}(i, g), f) &\longmapsto \varphi_2(i, \lambda a. \varphi^\diamond(g(a), \lambda b. f(a, b))). \end{aligned}$$

For the backwards part, we need to define a family of maps:

$$\psi_{(i,f)}^\diamond : C_{\varphi^\diamond(i,f)} \longrightarrow \sum_{a:A_i^\diamond} B_{f(a)}.$$

We again proceed by well-founded recursion on i . The two cases are given as follows:

$$\begin{aligned} \psi_{(\text{leaf}(\bullet), f)}^\diamond : C_{\varphi_1(f(\bullet))} &\longrightarrow \sum_{\bullet:1} B_{f(\bullet)} \\ r &\longmapsto (\bullet, \psi_{f(\bullet)}(r)) \end{aligned}$$

and, up to associativity of sum types $\sum$ in the codomain, and with the helper function $\beta(g, f) := \lambda a. \varphi^\diamond(g(a), \lambda b. f(a, b))$, we define:

$$\begin{aligned} \psi_{(\text{node}(i, g), f)}^\diamond : C_{\varphi_2(i, \beta(g, f))} &\longrightarrow \sum_{a:A_i} \sum_{\alpha:A_{g(a)}^\diamond} B_{f(a, \alpha)} \\ r &\longmapsto \begin{array}{l} \text{let } (a, r') = \psi_{(i, \beta(g, f))}(r) \text{ in} \\ \text{let } (b, w) = \psi_{(g(a), \lambda b. f(a, b))}^\diamond(r') \text{ in} \\ (a, b, w) \end{array} \end{aligned}$$

We leave it to the reader to check that $(\varphi^\diamond, \psi^\diamond)$ is the unique F -algebra morphism $(Q \star P^\diamond, (\text{in}^\diamond, \text{out}^\diamond)) \rightarrow (R, (\varphi, \psi))$. $\square$

As an application of [Theorem 5.3.11](#), we will prove the following theorem in [Appendix B](#), which shows that there is a natural monoid structure on $\mathcal{C}$ for which $P^\diamond$ should be considered the “free monoid on P ”.

Theorem B.1 (Free Monoid on P). *A monoid on P is a monoid $(P^\square, \mu, \eta)$ in the monoidal category $(\text{Cont}(\mathcal{C}), \star, \mathcal{I})$, equipped with a map $u : P \rightarrow P^\square$.*

There exists a monoid $(P^\diamond, \mu_P, \eta_P)$ with map $u_P : P \rightarrow P^\diamond$, such that, for any other monoid $(P^\square, \mu', \eta')$ and map $u' : P \rightarrow P^\square$, there exists a unique monoid morphism $\theta : P^\diamond \rightarrow P^\square$ satisfying $u' = \theta \circ u_P$.

That $P^\diamond$ is the free monoid on P is the categorified version of the main result from Westrick’s note on the diamond operator ²².

²² Westrick, ‘A note on the diamond operator’.

The interpretation of [Theorem 5.3.11](#) as an induction principle on degrees yields

$$Q \star P \leq Q \quad \implies \quad Q \star P^\diamond \leq Q.$$

This scheme corresponds to an induction axiom of the theory of right-handed Kleene algebras ²³. It is also a key part of a complete axiomatisation of the Weihrauch lattice augmented with iterated composition ²⁴, or rather, a slightly generalised version of it which may be derived using the cartesian closed structure of $\text{Cont}(\mathcal{C})$.

Lemma 5.3.12. *Let P, Q, R be containers. If there is a natural transformation $R \times (Q \star P) \rightarrow Q$ then there is a natural transformation $R \times (Q \star P^\diamond) \rightarrow Q$.*

Proof. In what follows, let us denote the exponential in $\text{Cont}(\mathcal{C})$ by $\Rightarrow$. First, construct a natural transformation $R \times ((R \Rightarrow Q) \star P) \rightarrow Q$ by:

$$\begin{aligned} & R \times ((R \Rightarrow Q) \star P) \\ \rightarrow & R \times R \times ((R \Rightarrow Q) \star P) && \text{via the diagonal map } R \rightarrow R \times R \\ \rightarrow & R \times (((R \Rightarrow Q) \times R) \star P) && \text{by [Lemma 4.2.13](#)} \\ \rightarrow & R \times (Q \star P) && \text{by using the evaluation map of } \Rightarrow \\ \rightarrow & Q && \text{by assumption.} \end{aligned}$$

After currying, we obtain a natural transformation $(R \Rightarrow Q) \star P \rightarrow R \Rightarrow Q$, so by [Theorem 5.3.11](#), we have $R \times ((R \Rightarrow Q) \star P^\diamond) \rightarrow Q$. We conclude by pre-composing the natural transformation $Q \rightarrow R \Rightarrow Q$ $\square$

As finite parallelisation has an infinitary countable counterpart, so does sequential iteration by considering a suitable fixpoint of the endofunctor $X \mapsto X \star P$. Much like in the case of parallelisation, the more interesting operator will be derived using the intermediate fixpoint construction ζ .

Definition 5.3.13 (Infinite Iteration). *Let $\mathcal{C}$ be a $\Pi\mathcal{WM}$ -category. Then the infinite iteration functor $-^\infty : \text{Cont}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$ is defined by*

$$P^\infty := \zeta(X \mapsto X \star P).$$

P^∞ corresponds to the problem whose instances are recipes to ask countably many questions to P in a sequential manner and whose solutions are ω -sequences of possible answers. An analogous operator has recently been introduced in Weihrauch reducibility ²⁵. In the case of Set , the corresponding endofunctor $\llbracket P^\infty \rrbracket$ maps X to the set of pairs (t, x) consisting of a tree t whose nodes are all labelled by P -questions, arities consist of the P -answers, and x maps infinite paths p through t to elements $x(p) \in X$.

When it comes to the less interesting fixpoints, similarly to the case of $X \mapsto P \times X$, we have $\mu(X \mapsto X \star P) \cong 0$ and $\nu(X \mapsto X \star P) \cong 1 \star P^\infty$. It is clear that, since $(1 \star P)^\infty \cong 1 \star P$, those two constructions yield idempotent operators on P . However, $(-)^\infty$ is not idempotent. From the perspective of Weihrauch complexity, we have that $P^{\infty\infty}$ is the problem where one can ask ω^2 questions to P sequentially and then get all of the answers, which is more powerful than asking ω questions sequentially (for instance, $\text{LPO}^\infty < \text{LPO}^{\infty\infty}$ ²⁶).

²³ Kozen and Silva, ‘Left-Handed Completeness’.

²⁴ Pradic, *The equational theory of the Weihrauch lattice with (iterated) composition*.

²⁵ Brattka, ‘Loops, Inverse Limits and Non-Determinism’.

²⁶ *ibid.*, Corollary 23.

5.4 Problems Definable as Closed ζ -Expressions

WE CAN DEFINE the following syntax for fibred polynomial functors over $\text{Cont}(\mathcal{C})$ when $\mathcal{C}$ is a $\Pi\mathcal{WM}$ -category.

Definition 5.4.1 (ζ -Expressions). *Let $\vec{X}$ be a collection of free variables. The set of ζ -expressions $\zeta\text{Expr}(\vec{X})$ with variables in $\vec{X}$ is generated by the following grammar:*

$$E, E' ::= X \mid \mu X. E \mid \zeta X. E \mid \nu X. E \mid E \times E' \mid E \otimes E' \mid E + E'$$

ζ -expressions E with k free variables have an obvious interpretation as fibred polynomial functors $\langle\!\langle E \rangle\!\rangle : \text{Cont}(\mathcal{C})^k \rightarrow \text{Cont}(\mathcal{C})$ given by [Lemma 5.2.4](#) and [Theorem 5.2.16](#).

When $k = 0$, we regard $\langle\!\langle E \rangle\!\rangle$ as an object of $\text{Cont}(\mathcal{C})$. We will also use the following obvious shorthands for ζ -expressions (where X is a fresh variable):

Notation 5.4.2.

$$\begin{array}{lll} 0 & := \mu X. X & \mathcal{I} := \zeta X. X \quad 1 := \nu X. X \\ E^* & := \mu X. \mathcal{I} + E \otimes X & \hat{E} := \zeta X. E \otimes X \quad \tilde{E} := \nu X. E \times X \end{array}$$

The operators discussed in [Section 5.3](#) are definable as denotations of ζ -expressions in a single variable, it is worthwhile to ask if any of the natural Weihrauch degrees we've seen, such as LPO ([Example 3.2.3](#)) or WKL ([Definition 3.2.5](#)), are definable as the denotations of *closed* ζ -expressions²⁷. The following result seems to rule this out, except in trivial cases.

²⁷ As usual, we identify functors from the terminal object with their images.

Proposition 5.4.3. *For any closed ζ -expression E , $\langle\!\langle E \rangle\!\rangle \equiv 0, \mathcal{I}$ or 1 .*

This result will follow immediately from the more general [Proposition 5.4.6](#), but first we need the following definition and lemma.

Definition 5.4.4 (Answer-Trivial). *We call a fibred polynomial functor $F : \text{Cont}(\mathcal{C})^k \rightarrow \text{Cont}(\mathcal{C})$ answer-trivial if $F(\vec{X})$ is degree-equivalent to $0, \mathcal{I}$ or 1 , for any tuple of objects $\vec{X}$ among $\{0, \mathcal{I}, 1\}$.*

Lemma 5.4.5. *If $F : \text{Cont}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$ is answer-trivial, so is $\gamma F : \text{Cont}(\mathcal{C})$ for any $\gamma \in \{\mu, \nu, \zeta\}$.*

Proof. Note that we have $\gamma F \equiv F^2(\gamma F) = F(F(\gamma F))$, $F^2(X) \equiv F^3(X)$ for any $X \in \{0, \mathcal{I}, 1\}$ and that $F^2(X)$ is degree-equivalent to $0, \mathcal{I}$ or 1 . We are going to establish that

$$\mu F \equiv F^2(0), \quad \zeta F \equiv F(\mathcal{I}) \quad \text{and} \quad \nu F \equiv F^2(1).$$

- If $\gamma = \mu$, then we have $F^2(0) \leq F^2(\mu F) \equiv \mu F$. Conversely, we can regard the map witnessing $F^3(0) \leq F^2(0)$ as an F -algebra and obtain $\mu F \leq F^2(0)$.
- If $\gamma = \nu$, we proceed dually with $F^2(1)$.
- If $\gamma = \zeta$, let us make a case distinction on the degree of $F(\mathcal{I})$, calling $F_0 : \mathcal{C} \rightarrow \mathcal{C}$ the functor such that $\text{shape} \circ F = F_0 \circ \text{shape}$:

- if $F(\mathcal{I}) \equiv 0$, then $F_0(0) \cong 0$. Since F_0 is a polynomial endofunctor and $\mathcal{C}$ is extensive, it means that F is constantly isomorphic to 0, and thus $\zeta F \cong 0$.
- Otherwise $F(\mathcal{I}) \equiv 1$ or $\mathcal{I}$ and $F_0(1) \equiv 1$. We then have a map $\varphi : 1 \rightarrow F_0(1)$. Since $(1, \varphi)$ is a F_0 -coalgebra, it induces a coalgebra map $\theta : 1 \rightarrow \nu F_0$. Then, by diagram chasing, we have $\theta^*(\zeta F) \cong \varphi^*(F(\theta^*(\zeta F)))$, which allows us to build a map $\theta^*(\zeta F) \rightarrow \varphi^*(F(\mathcal{I}))$ and conclude that $F(\mathcal{I}) \leq \zeta F$. If $F(\mathcal{I}) \equiv 1$, we are done. Otherwise, if $F(\mathcal{I}) \equiv \mathcal{I}$, $F(\mathcal{I})$ has a section (when regarded as a morphism of $\mathcal{C}$). Since F is fibred, this entails that $F(\text{id}_{\nu F_0})$ also has a section s . We then have a $c^* \circ F_{\nu F_0}$ -coalgebra with carrier the terminal object of $\mathcal{C}/\nu F_0$ for $F_{\nu F_0}$ taken as in Definition 5.2.1 and $c : \nu F_0 \rightarrow F_0(\nu F_0)$ the terminal F_0 -coalgebra in $\mathcal{C}$. Since ζF is the carrier of the terminal $c^* \circ F_{\nu F_0}$ coalgebra, we have a coalgebra map $\text{id}_{\nu F_0} \rightarrow \zeta F$ which is nothing more than a section of ζF , witnessing $\zeta F \leq \mathcal{I}$. $\square$

Proposition 5.4.6. *Any fibred polynomial functor F built inductively from projections, $\otimes$, $\times$, $+$, tupling, composition and γ -fixpoints for $\gamma \in \{\mu, \nu, \zeta\}$ is answer-trivial.*

Proof. The proof is via a straightforward induction after considering the following facts:

- Lemma 5.4.5 also implies that for any answer-trivial $F : \text{Cont}(\mathcal{C})^{k+1} \rightarrow \text{Cont}(\mathcal{C})$, the functor $\vec{X} \mapsto \gamma(Y \mapsto F(\vec{X}, Y))$ is also answer-trivial.
- All monoidal products we consider are answer-trivial (see Figure 5.4).
- Both the tupling and composition of answer-trivial functors is answer-trivial.
- The identity is answer-trivial. $\square$

+	0	$\mathcal{I}$	1
0	0	$\mathcal{I}$	1
$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	1
1	1	1	1

$\times$	0	$\mathcal{I}$	1
0	0	0	0
$\mathcal{I}$	0	$\mathcal{I}$	$\mathcal{I}$
1	0	$\mathcal{I}$	1

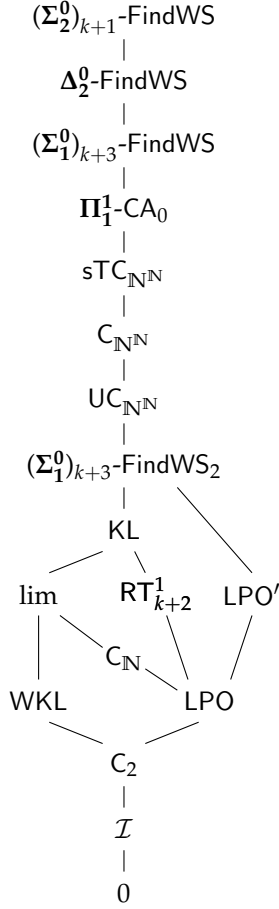
$\otimes$	0	$\mathcal{I}$	1
0	0	0	0
$\mathcal{I}$	0	$\mathcal{I}$	1
1	0	1	1

$\star$	0	$\mathcal{I}$	1
0	0	0	1
$\mathcal{I}$	0	$\mathcal{I}$	1
1	0	1	1

Figure 5.4: Multiplication table for degrees given by the monoidal products and their units.

The problem with many of the fixpoints we can construct is, essentially, that they contain “junk”; more specifically, viewed as internal families, they contain empty sets. To rectify this situation, we use the factorisation from Theorem 2.4.22.

Definition 5.4.7 (Answerable Part). *The answerable part $\text{Ans}(P)$ of a container $P : A \rightarrow I$ in $\text{Cont}(\text{ReprSp})$ is the corestriction of P to its image.*



Weihrauch degree D	ζ -expression E with $\text{Ans}(\langle\langle E \rangle\rangle) \equiv D$
$(\Sigma_2^0)_{2k}$ -FindWS	$\zeta X_{2k}.\nu X_{2k-1} \dots \zeta X_1.\nu X_0. \sum_{i=0}^{2k} X_i$
Δ_2^0 -FindWS	$\mu Z. \zeta X. \widehat{X} + (\nu Y. \widehat{Y} + Z)$
$sTC_{\mathbb{N}}$	$(\nu X. \widehat{X} + \mathcal{I}) \times (\zeta X. \widehat{X} + 1)$
$C_{\mathbb{N}}$	$\zeta X. \widehat{X} + 1$
Δ_1^0 -FindWS	$\mu X. \widehat{X} + \mathcal{I} + 1$
KL	$\zeta X. (\nu Y. X + Y) \times (\nu Y. X + Y)$
lim	$(\nu X. X + \mathcal{I}) \times (\zeta X. X + 1)$
WKL	$\zeta X. 1 + X \times X$
RT_k^1	$\underbrace{(\zeta X. \nu Y. X + Y) \times \dots}_{k \text{ times}}$
LPO'	$(\nu X. \zeta Y. X + Y) \times (\zeta X. \nu Y. X + Y)$
$C_{\mathbb{N}}$	$\zeta X. X + 1$
LPO	$(\nu X. X + \mathcal{I}) \times (\zeta X. X + 1)$
C_k	$\underbrace{(\zeta X. X + 1) \times \dots}_{k \text{ times}}$
$\mathcal{I}$	$\zeta X. X$
0	$\mu X. X$

It is obvious that the answerable part of a container is answerable. Note that, contrary to the other operations on containers we discussed, it is not functorial, nor monotone when we look at the induced degrees: for instance, the answerable part of 1 is 0, while the answerable part of $\mathcal{I} + 1$ is $\mathcal{I}$, although we have $1 \equiv \mathcal{I} + 1$. More generally, we have the following interaction between Ans and monoidal products (the expected $\mathbb{N}$ -ary versions also hold).

Proposition 5.4.8. *For any containers P and Q over ReprSp , we have*

$$\begin{array}{llll}
 \text{Ans}(0) & \cong & \text{Ans}(1) & \cong & 0 \\
 \text{Ans}(P + Q) & \cong & \text{Ans}(P) + \text{Ans}(Q) & & \\
 \text{Ans}(P \times Q) & \geq & \text{Ans}(P) \times \text{Ans}(Q) & & \\
 \text{Ans}(\mathcal{I}) & \cong & \mathcal{I} & & \\
 \text{Ans}(P \otimes Q) & \cong & \text{Ans}(P) \otimes \text{Ans}(Q) & & \\
 \text{Ans}(P \star Q) & \cong & \text{Ans}(P) \star \text{Ans}(Q) & &
 \end{array}$$

Remark 5.4.9. By Remark 2.4.11, it is possible to check the definition of $\langle\langle L \rangle\rangle$ always restricts to a functor $\text{Cont}(\text{rpReprSp})^{|x|} \rightarrow \text{Cont}(\text{rpReprSp})$, and hence answerable parts of denotations of ζ -expressions correspond to genuine Weihrauch degrees (in contrast to the composition operator $\star$ and its iterations in the previous section). The proof of this relies on the automata-theoretic characterisation of $\langle\langle - \rangle\rangle$ given by Pradic & Price²⁸ and is omitted.

At long last, we are equipped to define a number of Weihrauch degrees as answerable parts of denotations of fixpoints. We will only give a small selection of these, deferring to Pradic & Price²⁹ for the remainder of those in Figure 5.5.

Figure 5.5: Problems from the Weihrauch lattice as the answerable part of the interpretation of some ζ -expressions in represented spaces. On the left-hand side is the part of the Weihrauch lattice we capture, where all inequalities are known to be strict below $\Pi_1^1\text{-CA}_0$. Γ -FindWS is the answerable part of the problem of finding winning strategies in a Gale-Stewart game.

²⁸ Pradic and Price, 'Problems with Fixpoints of Polynomials of Polynomials', Def. 26, Thm. 27.

²⁹ *ibid.*

5.4.1 Examples

THE FIRST EXAMPLE is characterising LPO by a ζ -expression reminiscent of the definition of conatural numbers.

Proposition 5.4.10. $\text{LPO} \equiv_W \text{Ans}(\langle\langle \zeta X. X + 1 \rangle \times (\nu X. X + \mathcal{I}) \rangle\rangle)$.

Proof. If we computing the isomorphism type of $\langle\langle \zeta X. X + 1 \rangle\rangle$, we obtain a container corresponding to $\infty : 1 \rightarrow \mathbb{N}_\infty$. Similarly, $\langle\langle \nu X. X + \mathcal{I} \rangle\rangle$ will correspond to the embedding of $\mathbb{N}$ into $\mathbb{N}_\infty$. We thus have two problems that take as input some $x \in \mathbb{N}_\infty$ which can have a trivial answer. That answer exists only if $x = \infty$ in the first case, and conversely only if $x = \underline{n}$ for some $n \in \mathbb{N}$ in the second case. Hence, questions to $\langle\langle \zeta X. X + 1 \rangle \times (\nu X. X + \mathcal{I}) \rangle\rangle$ are pairs $(x, y) \in \mathbb{N}_\infty^2$ and legal answers are Booleans $b \in \mathbb{2}$ such that

$$b = 0 \implies x = \infty \quad \text{and} \quad b = 1 \implies y < \infty.$$

Taking the answerable part ensures that we only have inputs such that $y = \infty \implies x = \infty$. On the other hand, LPO is easily seen to be equivalent to the problem taking some $x \in \mathbb{N}_\infty$ and outputting the Boolean b with $b = 0 \iff x = \infty$. We can take the diagonal map $x \mapsto (x, x)$ to be the forward part of a reduction $\text{LPO} \leq \langle\langle \zeta X. X + 1 \rangle \times (\nu X. X + \mathcal{I}) \rangle\rangle$, and the backwards part to be trivial. For the converse reduction, the forward part is simply the second projection, and the backward part is trivial. $\square$

The next degree we consider captures *closed choice problems* over (subspaces of) $\mathbb{N}$ with the usual topology.

Definition 5.4.11 (Enumeration). An enumeration $A \subseteq \mathbb{N}$ is a map $e : \mathbb{N} \rightarrow \{\perp\} \sqcup \mathbb{N}$ such that $A = e(\mathbb{N}) \cap \mathbb{N}$. For $A \subseteq \mathbb{N}$, define C_A to be the following problem: questions are enumerations e of strict subsets of A , and answers are those $m \in A$ which are not enumerated by A .

Proposition 5.4.12. $C_{\mathbb{N}} \equiv \text{Ans}(\langle\langle \widetilde{\zeta X. X + 1} \rangle\rangle)$ and, for $k \in \mathbb{N}$,

$$C_k \equiv \text{Ans} \left(\left\langle \underbrace{\langle\langle \zeta X. X + 1 \rangle \times \dots \times (\zeta X. X + 1) \rangle\rangle}_{k \text{ times}} \right\rangle \right).$$

Proof. Recall from [Example 5.2.15](#) that $\langle\langle \zeta X. X + 1 \rangle\rangle$ is isomorphic to the container $\infty : 1 \rightarrow \mathbb{N}_\infty$. As a problem, a question is a conatural number, and a question can be (trivially) answered if and only if that conatural number is ∞ . This means that $\langle\langle \zeta X. X + 1 \rangle\rangle$ is the problem whose questions are sequences of conatural numbers $x \in \mathbb{N}_\infty^{\mathbb{N}}$ which are answered by any $n \in \mathbb{N}$ such that $x_n = \infty$; a fortiori, $\text{Ans} \langle\langle \zeta X. X + 1 \rangle\rangle$ restricts the questions so that an $n \in \mathbb{N}$ with $x_n = \infty$ does exist.

The forward part of a reduction $C_{\mathbb{N}} \leq \text{Ans} \langle\langle \widetilde{\zeta X. X + 1} \rangle\rangle$ simply sends an enumeration e to the sequence

$$(\sup\{x \in \mathbb{N}_\infty \mid \forall k \in \mathbb{N}. \underline{k} < x \Rightarrow e_k \neq n\})_{n \in \mathbb{N}},$$

and the corresponding backwards part is trivial. Conversely, from a sequence $x \in \mathbb{N}_\infty^\mathbb{N}$, we can compute

$$e(\langle n, m \rangle) = \begin{cases} n & \text{if } x_n \leq m \\ \infty & \text{otherwise} \end{cases}$$

as part of a forward reduction (assuming $\langle -, - \rangle : \mathbb{N}^2 \rightarrow \mathbb{N}$ is the standard Cantor pairing function) and have a trivial backwards reduction. The proof for C_k is essentially the same. $\square$

The final example we discuss is weak König's lemma (Definition 3.2.5), which is another instance of closed choice, namely $C_{2^\mathbb{N}}$.

Proposition 5.4.13. *WKL is Weihrauch equivalent to the answerable part of $\langle\langle \zeta X. 1 + X \times X \rangle\rangle$.*

Proof. The shape of $\langle\langle \zeta X. 1 + X \times X \rangle\rangle$ is the terminal coalgebra of $I \mapsto 1 + I \times I$. That coalgebra $(T^\infty, \text{out}_{T^\infty})$ has the set of possibly-infinite binary trees as its carrier, and the coalgebra map sends leaves leaf to $\star$ and nodes $\text{node}(t_l, t_r)$ to pairs (t_l, t_r) .

Then the induced functor on $\mathcal{C}/T^\infty$ described by Lemma 5.2.9 operates on families $(X_t)_{t \in T^\infty}$ as follows:

$$\begin{aligned} X_{\text{leaf}} &\mapsto \emptyset && (\text{leaf}) \\ X_{\text{node}(t_l, t_r)} &\mapsto X_{t_l} \sqcup X_{t_r} && (\text{node}) \end{aligned}$$

It is clear that the iterating this functor sends a family X_t to a disjoint union $\bigsqcup_{\text{paths } p \text{ in } t} X_p$ over the set of (potentially infinite) paths in the tree.

If a path p is finite, (leaf) guarantees that X_p is empty. For infinite paths, X_p will be inhabited in the *greatest* fixpoint. Taking the answerable part ensures that we only consider the subfamily where X_t is non-empty, i.e., t has infinite paths and so cannot be a finite tree. $\square$

It is very satisfying that the fixpoint expression resembles the inductive definition of binary trees.

6 Strong Reductions as Adaptors

AS WE HAVE SEEN, viewing problems as bundles is a useful viewpoint that enables to generalise Weihrauch reductions as container morphisms, however, computability theorists are also interested in a uniform (or strong) notion of reduction. In this section, we show how to accomodate these using a different kind of lens-like construction called “Adaptors”.

6.1 Internal Relations

IN THE PROBLEMS-AS-BUNDLES VIEWPOINT, we viewed (total) relations as projections from families of non-empty sets to the indexing set, which can then be generalised to pullback-preserving epimorphisms which we referred to as “answerable containers” (Definition 3.5.1). However, a different avenue of attack might have been to categorify the notion of an “internal relation” in a category.

One possibility is to take the definition of a relation literally as a subset of a cartesian product, from which you would generalise to a subobject of a product. This is a good start, however category-theorists prefer to think in terms of maps rather than objects. Let’s fix some notations: take objects R, X, Y , a monic map $i : R \hookrightarrow X \times Y$, and the projections $\pi_X : X \times Y \rightarrow X$, $\pi_Y : X \times Y \rightarrow Y$. Relations come with two natural maps: a map to its domain $\text{dom}_R = \pi_X \circ i$, and a map to its codomain $\text{cod}_R = \pi_Y \circ i$. Individually, we can’t say anything more about them, but taken together $\langle \text{dom}_R, \text{cod}_R \rangle = \langle \pi_X, \pi_Y \rangle \circ i = i$, and hence the pair of maps is monic. We will take this as our categorical definition of a relation, which is valid even in categories without products.

Definition 6.1.1 (Jointly Monic). *A pair of morphisms $f : R \rightarrow X, g : R \rightarrow Y$ in a category $\mathcal{C}$ are jointly monic when for any object Z and parallel morphisms $h, j : Z \rightarrow R$, if $f \circ h = f \circ j$ and $g \circ h = g \circ j$, then $h = j$.*

Definition 6.1.2 ((Internal) Relation). *An (internal) (binary¹) relation from an object X to an object Y in a category $\mathcal{C}$ is an object R and a pair of maps $\text{cod}_R : R \rightarrow Y, \text{dom}_R : R \rightarrow X$, such that cod_R and dom_R are jointly monic.*

¹ We shall only treat the binary case, but extending this to n -ary relations is straightforward.

Convention 6.1.3. *We will typically write give relations as spans*

$$X \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} Y,$$

and will frequently refer to a relation by the apex R of such a span. We will use $R, S, T \dots$ for the names of relations.

Definition 6.1.4 (Morphism of Relations). Let $X \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} Y$ and $Z \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} W$ be relations. A morphism $R \rightarrow S$ is a triple $(f : X \rightarrow Z, g : R \rightarrow S, h : Y \rightarrow W)$, such that the following diagram commutes.

$$\begin{array}{ccc}
 Y & \xrightarrow{h} & W \\
 \text{cod}_R \uparrow & & \uparrow \text{dom}_R \\
 R & \xrightarrow{g} & S \\
 \text{dom}_R \downarrow & & \downarrow \text{cod}_S \\
 X & \xrightarrow{f} & Z
 \end{array}$$

Relations obviously form a category, which we denote by $\text{RelSpan}(\mathcal{C})$.

Remark 6.1.5. In typical presentations, e.g., “Categories, Allegories”² or the Elephant book³, regular categories are used when working with relations, since this allows the definition of an associative composition of relations⁴. In this case, one can define a 2-category $\text{Rel}(\mathcal{C})$ whose objects are those of $\mathcal{C}$ and whose morphisms are (isomorphism classes of) relations in $\mathcal{C}$. However, all this structure is too much for our purposes, and we will stick to $\text{RelSpan}(\mathcal{C})$.

² Freyd and Andre, *Categories, Allegories*.

³ Johnstone, *Sketches of an Elephant: A Topos Theory Compendium*.

⁴ *ibid.*, Lemma A3.1.1.

A total relation is a relation for which every element of the domain is related to some element, i.e., the domain map is a surjection. Similarly, a cototal relation is one for which the codomain map is a surjection. More generally, we are interested in maps which belong to the left class in a factorisation system.

Definition 6.1.6 ((Co)total Relation). Let $\mathcal{C}$ be a category with a factorisation system $(\mathcal{L}, \mathcal{R})$. A total (resp. cototal) relation from an object X to an object Y is a relation $X \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} Y$ such that $\text{dom}_R \in \mathcal{L}$ (resp. $\text{cod}_R \in \mathcal{L}$).

Notation 6.1.7. In this chapter, because we will generally work up to a factorisation system $(\mathcal{L}, \mathcal{R})$, double-headed arrows $X \rightrightarrows Y$ will always indicate members of the class $\mathcal{L}$ and hooked arrows $X \hookrightarrow Y$ will always indicate members of $\mathcal{R}$.

If we go back to the indexed point of view, recall that we view a family $(X_i)_{i \in I}$ as the projection $\pi_I : \sum_{i \in I} X(i) \rightarrow I$. In this case, the right leg of a relation being an epimorphism that is jointly monic with π_I tells us that it is a valid quotient map $\sum_{i \in I} X_i \rightarrow \bigcup_{i \in I} X(i)$.

Total (resp. cototal) relations form a full subcategory $\text{RelTot}(\mathcal{C})$ (resp. $\text{RelCoTot}(\mathcal{C})$) of $\text{RelSpan}(\mathcal{C})$. We might ask ourselves if we still have these categories are fibred over $\mathcal{C}$. The following proposition is folklore, and suggests that there should be a fibration $\text{RelCoTot}(\mathcal{C}) \rightarrow \mathcal{C}$.

Proposition 6.1.8. Let $\mathcal{C}$ be a category with a factorisation system $(\mathcal{L}, \mathcal{R})$, where $\mathcal{L}$ and $\mathcal{R}$ are full subcategories of $\mathcal{C}^\rightarrow$. Then the domain functor $\mathcal{L} \rightarrow \mathcal{C}$ is a fibration.

Proof. Fix $\pi_J : Y \rightrightarrows J \in \mathcal{L}$ and $f : X \rightarrow Y \in \mathcal{C}$. We can obtain a morphism π in $\mathcal{L}$ by applying the factorisation to the map $\pi_J \circ f$.

$$\begin{array}{ccc}
 \text{Im}(\pi_J \circ f) & \xleftarrow{i} & J \\
 \uparrow \pi & & \uparrow \pi_J \\
 X & \xrightarrow{f} & Y
 \end{array}$$

To check that morphism $(i, f) : \pi \rightarrow \pi_J$ is cartesian, suppose we have an object $\pi_K : Z \rightarrow K$ and a morphism $(j : K \rightarrow J, g : Z \rightarrow Y) : \pi_K \rightarrow \pi_J$ in $\mathcal{L}$, and further suppose a map $h : Z \rightarrow X$ such that $f \circ h = g$. Then we need to show there is a unique map $k : K \rightarrow \text{Im}(\pi_J \circ f)$ such that the diagram below commutes.

$$\begin{array}{ccccc}
 & & K & & \\
 & \swarrow k & \uparrow & \searrow j & \\
 \text{Im}(\pi_J \circ f) & \xleftarrow{i} & & J & \\
 \uparrow \pi & & \pi_K & & \uparrow \pi_J \\
 & & Z & & \\
 \swarrow h & & & \searrow g & \\
 X & \xrightarrow{f} & & Y &
 \end{array} \quad (\text{exists-}k)$$

To define such a k , we first factorise the maps j and $\pi \circ h$. Since $\mathcal{L}$ and $\mathcal{R}$ are both closed under composition, we obtain two factorisations of $j \circ \pi_K$, and so there is a unique isomorphism m between the two making the following diagram commute.

$$\begin{array}{ccccc}
 & & \text{Im}(j) & & \\
 & \nearrow j^l & \downarrow & \searrow j^r & \\
 K & \xrightarrow{j} & & J & \\
 \uparrow \pi_K & & & & \uparrow i \\
 Z & \xrightarrow{h} X \xrightarrow{\pi} & \text{Im}(\pi_J \circ f) & & \\
 & \searrow (\pi \circ h)^l & \downarrow m & \nearrow (\pi \circ h)^r & \\
 & & \text{Im}(\pi \circ h) & &
 \end{array}$$

We can then read off the required map $k := (\pi \circ h)^r \circ m \circ j^l$. To see that k is uniquely determined, suppose there is another k' that makes diagram in Equation (exists- k) commute, and then factor that $k' = k^r \circ k^l$. Since $k' \circ \pi_K = \pi \circ h$, there is a unique isomorphism $\text{Im}(k') \cong \text{Im}(\pi \circ h)$. Similarly, since $i \circ k' = j$, there is a unique isomorphism $\text{Im}(k') \cong \text{Im}(j)$. Composing these isomorphisms gives you m by uniqueness, and $k' = k$ follows. This is summarised in the following commuting diagram.

$$\begin{array}{ccccc}
 & & \text{Im}(j) & \xleftarrow{j^r} & J \\
 & \nearrow j^l & \downarrow \exists! & & \uparrow i \\
 K & \xrightarrow{k^l} & \text{Im}(k') & \xleftarrow{k^r} & \text{Im}(\pi_J \circ f) \\
 \uparrow \pi_K & & \downarrow \exists! & & \\
 Z & \xrightarrow{(\pi \circ h)^l} & \text{Im}(\pi \circ h) & \xleftarrow{(\pi \circ h)^r} &
 \end{array} \quad \square$$

Indeed, there is a fibration $\text{RelCoTot}(\mathcal{C}) \rightarrow \mathcal{C}$ that sends cototal relations to their domains.

Proposition 6.1.9. ⁵ Let $\mathcal{C}$ be a category with pullbacks and a factorisation system $(\mathcal{L}, \mathcal{R})$, the domain functor $\text{dom} : \text{RelCoTot}(\mathcal{C}) \rightarrow \mathcal{C}$ is a fibration.

⁵ Braithwaite, Capucci, Gavranović, Hedges and Rischel, 'Fibre optics II', Proposition 10.

Proof. Let $J \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} Y$ be a relation and $f : I \rightarrow J$ an arrow in $\mathcal{C}$. The cartesian morphism $\text{cart}(f, S)$ is given by the following diagram:

$$\begin{array}{ccc}
 \text{Im}(\pi_Y \circ \pi_S) & \hookrightarrow & Y \\
 \uparrow \pi & & \uparrow \text{cod}_S \\
 I \times_J S & \xrightarrow{\pi_S} & S \\
 \downarrow \pi_I & \lrcorner & \downarrow \text{dom}_S \\
 I & \xrightarrow{f} & J
 \end{array}$$

Suppose we have two parallel morphisms $\alpha, \beta : W \rightarrow I \times_J S$ such that $\pi_I \circ \alpha = \pi_I \circ \beta$ and $\pi \circ \alpha = \pi \circ \beta$. It follows that $\pi_S \circ \alpha = \pi_S \circ \beta$, since dom_S and cod_S are jointly monic. But then, $\alpha = \beta$ by the universal property of the pullback. Hence π_I and π are jointly monic.

It is easy to see that the morphism is cartesian by inspecting the following diagram. The map $T \rightarrow I \times_J S$ exists (and is unique) by the pullback lemma, and the map $Z \rightarrow \text{Im}(\pi_Y \circ \pi_S)$ exists (and is unique) by functoriality of the factorisation system.

$$\begin{array}{ccccc}
 & & Z & & \\
 & & \uparrow & \searrow j & \\
 \text{Im}(\pi_Y \circ \pi_S) & \hookrightarrow & & & Y \\
 \uparrow \pi & & \text{cod}_T & & \uparrow \text{cod}_S \\
 & & T & & S \\
 & \swarrow \pi_S & \searrow h & & \\
 I \times_J S & & & & \\
 \downarrow \pi_I & \lrcorner & \text{dom}_T & & \downarrow \text{dom}_S \\
 & & K & & \\
 & \swarrow h & \searrow g & & \\
 I & \xrightarrow{f} & J & &
 \end{array}$$

□

The last property of $\text{RelCoTot}(\mathcal{C})$ we will need is the existence of a free/forgetful adjunction with the arrow category.

Proposition 6.1.10. ⁶ Let $\mathcal{C}$ be a category with pullbacks and a factorisation system $(\mathcal{L}, \mathcal{R})$. Then there is an adjunction

⁶ Braithwaite, Capucci, Gavranović, Hedges and Rischel, ‘Fibre optics II’, Proposition 11.

$$\begin{array}{ccc}
 \mathcal{C}^{\rightarrow} & \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{U} \end{array} & \text{RelCoTot}(\mathcal{C}) \\
 \downarrow \text{cod} & & \downarrow \text{dom} \\
 & \mathcal{C} &
 \end{array}$$

where the functors are defined as follows.

$$\begin{aligned}
 U : \text{RelCoTot}(\mathcal{C}) &\rightarrow \mathcal{C}^{\rightarrow} & F : \mathcal{C}^{\rightarrow} &\rightarrow \text{RelCoTot}(\mathcal{C}) \\
 U(I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X) &= \text{dom}_R & F(I \xleftarrow{\pi_I} X) &= I \xleftarrow{\pi_I} X \xrightarrow{\text{id}_X} X
 \end{aligned}$$

Proof. It is easily checked that both U and F are functors and that U is fibred. Note that $F(\pi_I)$ is automatically jointly monic since the right leg is the identity.

Since $U \circ F = \text{id}_{\mathcal{C}^\rightarrow}$ ⁷, we can take the unit η to be the identity natural transformation. For a cototal relation $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$, we define the component of the counit by $\epsilon_R = (\text{id}_I, \text{id}_R, \text{cod}_R)$.

It remains to check the triangular identities, however these are true as all four of $F(\eta)$, ϵF , ηU and $U(\epsilon)$ are identities. $\square$

Remark 6.1.11. *The above adjunction is very nearly a fibred adjunction, lacking only that F is a fibred functor. However, if F preserved cartesian morphisms then*

for any pullback square

$$\begin{array}{ccc} X & \xrightarrow{g} & Y \\ \pi_I \downarrow & \lrcorner & \downarrow \pi_J \\ I & \xrightarrow{f} & J \end{array}$$

example of this failing would be Set with the usual surjection/injection factorisation

system and the pullback square

$$\begin{array}{ccc} X \times Y & \xrightarrow{\pi_Y} & Y \\ \pi_X \downarrow & \lrcorner & \downarrow ! \\ X & \xrightarrow{!} & 1 \end{array}$$

Remark 6.1.12. *If $\mathcal{C}$ also has a terminal object, then U has a right adjoint that sends the arrow $I \xleftarrow{\pi_I} R$ to the relation $I \xleftarrow{\pi_I} R \xrightarrow{!} 1$, where $!^r \circ !^l$ is factorisation of the unique map.*

6.2 Adaptors

RECALL FROM SECTION 3.4.2 that the category of containers $\text{Cont}(\mathcal{C})$ over a category $\mathcal{C}$ arises as the total space of the shape fibration, which was in turn the opposite of the fibration $\text{cod} : \mathcal{C}^\rightarrow \rightarrow \mathcal{C}$. We will now take the same approach with the fibration $\text{dom} : \text{RelCoTot}(\mathcal{C}) \rightarrow \mathcal{C}$ from Proposition 6.1.9.

Definition 6.2.1 (Dependent Adaptors).⁸ *Let $\mathcal{C}$ be a category with pullbacks and a factorisation system. We define the category of dependent adaptors, written $\text{DAdt}(\mathcal{C})$, to be the total space of the fibration $\text{dom}^{(\text{op})}$. A morphism in $\text{DAdt}(\mathcal{C})$ is called a dependent adaptor in $\mathcal{C}$.*

Remark 6.2.2. *Let $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ and $J \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} Y$ be cototal relations. Then a morphism $R \rightarrow S$ can be explicitly given as (an equivalence class of) a triple (φ, ψ, χ) ⁹ which fits in a diagram in $\mathcal{C}$ of the form*

$$\begin{array}{ccccc} X & \xleftarrow{\chi} & \text{Im}(\pi_Y \circ \pi_S) & \xrightarrow{\quad} & Y \\ \text{cod}_R \uparrow & & \pi \uparrow & & \uparrow \text{cod}_S \\ R & \xleftarrow{\psi} & I \times_J S & \xrightarrow{\pi_S} & S \\ \text{dom}_R \downarrow & & \pi_I \downarrow & \lrcorner & \downarrow \text{dom}_S \\ I & \xlongequal{\quad} & I & \xrightarrow{\varphi} & J \end{array}$$

where the top right commuting square comes from the factorisation of $\pi_Y \circ \pi_S$.

⁷ Consequently, F is faithful and U is full.

⁸ *ibid.*, Definition 5.

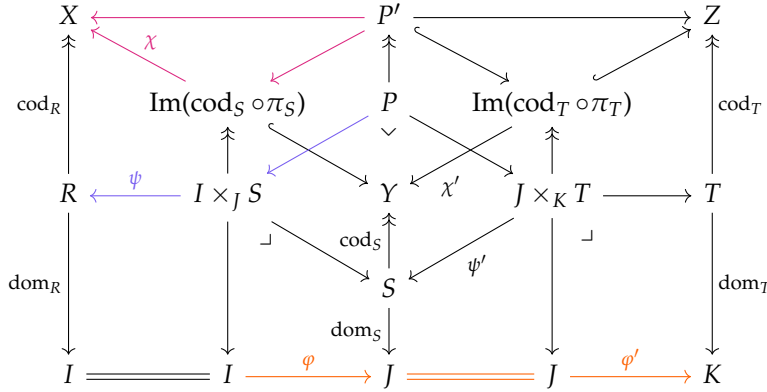
⁹ We reuse the convention of orange for forward maps and blue for backward maps from container morphisms, but we augment it with purple for the third map χ , which witnesses that ψ is uniform on shapes.

Convention 6.2.3. In order to agree with the letter conventions for containers (Convention 3.4.2), we will use $I, J, K, \dots$ for the domains¹⁰ of cototal relations in $\mathbf{DA}(\mathcal{C})$, the letters $R, S, T, \dots$ for relations and the apexes of their spans, and $X, Y, Z, \dots$ for their images¹¹.

¹⁰ That is, the codomain of the map dom .

¹¹ Similarly, the codomain of cod .

Remark 6.2.4. Suppose we have three cototal relations $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$, $J \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} Y$ and $K \xleftarrow{\text{dom}_T} T \xrightarrow{\text{cod}_T} Z$, and two adaptors $(\varphi, \psi, \chi) : R \rightarrow S$ and $(\varphi', \psi', \chi') : S \rightarrow T$. Then we can explicitly give their composition by $\gamma(I \xrightarrow{\varphi} J \xrightarrow{\varphi'} K, P \rightarrow I \times_J S \xrightarrow{\psi} Y, P' \rightarrow \text{Im}(\text{cod}_S \circ \pi_S) \xrightarrow{\chi} X) : R \rightarrow T$, as in the following commuting diagram.



P is the pullback of $I \times_J S$ and $J \times_K T$, $P \twoheadrightarrow P' \hookrightarrow Z$ is the factorisation of the map $P \rightarrow J \times_T T \rightarrow \text{Im}(\text{cod}_T \circ \pi_T) \twoheadrightarrow Z$, and the map $P' \rightarrow \text{Im}(\text{cod}_T \circ \pi_T)$ is uniquely guaranteed by functoriality of factorisations. The arrow $P' \rightarrow \text{Im}(\text{cod}_S \circ \pi_S)$ is the unique morphism that makes the following diagram commute.

$$\begin{array}{ccccc} P & \twoheadrightarrow & P' & \hookrightarrow & \text{Im}(\text{cod}_T \circ \pi_T) \\ \downarrow & & \downarrow & & \downarrow \\ I \times_J S & \twoheadrightarrow & \text{Im}(\text{cod}_S \circ \pi_S) & \hookrightarrow & Y \end{array}$$

The identity adaptor on $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ is just $(\text{id}_I, \text{id}_R, \text{id}_X)$. Vertical adaptors are those for which φ is an isomorphism and likewise cartesian adaptors are those for which ψ and χ are isomorphisms.

The $\text{dom}^{(\text{op})}$ fibration on adaptors is closely related to the category of bidirectional maps $\mathcal{C} \times \mathcal{C}^{\text{op}}$, which is also fibred over $\mathcal{C}$ by taking the first projection (Example A.4.4).

Proposition 6.2.5. Let $\mathcal{C}$ be a category with pullbacks and a factorisation system $(\mathcal{L}, \mathcal{R})$. Then there is a fibred functor $\mathcal{C} \times \mathcal{C}^{\text{op}} \rightarrow \mathbf{DA}(\mathcal{C})$ that takes the object (I, X) to $I \xleftarrow{\pi_I} I \times X \xrightarrow{\pi_X^l} \text{Im}(\pi_X)$ where $\pi_X^r \circ \pi_X^l = \pi_X$ is the $(\mathcal{L}, \mathcal{R})$ -factorisation of the projection $\pi_X : I \times X \rightarrow X$.

Proof. Firstly, $I \xleftarrow{\pi_I} I \times X \xrightarrow{\pi_X^l} X$ is jointly monic. If $\pi_X^l \circ f = \pi_X^l \circ g$, then $\pi_X \circ f = \pi_X^r \circ \pi_X^l \circ f = \pi_X^r \circ \pi_X^l \circ g = \pi_X \circ g$. Taken together with $\pi_I \circ f = \pi_I \circ g$, we have $f = g$ by the universal property of products.

Suppose we have a morphism $(\varphi, \psi) : (I, X) \rightarrow (J, Y) \in \mathcal{C}$. Then we define the functor's action on morphisms by sending it to $(\varphi, \text{id}_X \times \psi, \chi)$ according to the following commuting diagram:

$$\begin{array}{ccc} \mathcal{C} \times \mathcal{C}^{\text{op}} & \xrightarrow{\quad} & \mathbf{DA}(\mathcal{C}) \\ \pi_1 \searrow & & \swarrow \text{dom}^{(\text{op})} \\ & \mathcal{C} & \end{array}$$

$$\begin{array}{ccccc}
 X & \xleftarrow{\psi} & Y & \xlongequal{\quad} & Y \\
 \uparrow \pi_X^r & & \uparrow & & \uparrow \pi_Y^r \\
 \text{Im}(\pi_X) & \xleftarrow{\chi} & \text{Im}(\pi_Y)' & \xleftarrow{\omega} & \text{Im}(\pi_Y) \\
 \uparrow \pi_X^l & & \uparrow & & \uparrow \pi_Y^l \\
 I \times X & \xleftarrow{\text{id}_X \times \psi} & I \times Y & \xrightarrow{\varphi \times \text{id}_Y} & J \times Y \\
 \downarrow \pi_I & & \downarrow \pi_I & \lrcorner & \downarrow \pi_J \\
 I & \xlongequal{\quad} & I & \xrightarrow{\varphi} & J
 \end{array}$$

In this case $\text{Im}(\pi_Y)'$ comes from the factorisation of $\pi_Y : I \times Y \rightarrow Y$ and the morphisms χ and ω exist and are unique by the functoriality of factorisations, and $\omega \in \mathcal{R}$ by [Lemma A.3.4](#).

It is easy to see that this preserves identity morphisms. To check that it preserves composition, suppose that $(\varphi, \psi) : (I, X) \rightarrow (J, Y)$ and $(\varphi', \psi') : (J, Y) \rightarrow (K, Z)$ are morphism in $\mathcal{C} \times \mathcal{C}^{\text{op}}$. Composing according to [Remark 6.2.4](#), we obtain the following commuting diagram¹², which gives an adaptor for $(\varphi' \circ \varphi, \psi \circ \psi')$.

¹² In the interest of clarity, I have elided all inessential names. All vertically drawn arrows are either projections or arise from factorisations of projections. The diagonal maps out of $\text{Im}(\pi_Z)''$ arise from functoriality of factorisations in their respective commuting squares.

$$\begin{array}{ccccccc}
 & & & & Z & \xlongequal{\quad} & Z \\
 & & & & \uparrow & & \uparrow \pi_Z^r \\
 & & & & \text{Im}(\pi_Z)'' & & \text{Im}(\pi_Z)' \xrightarrow{\quad} \text{Im}(\pi_Z) \\
 & & & & \uparrow & & \uparrow \pi_Z^l \\
 X & \xleftarrow{\psi} & Y & \xleftarrow{\chi''} & Y & \xleftarrow{\text{id}_Y \times \psi'} & I \times Z \\
 \uparrow \pi_X^r & & \uparrow & & \uparrow & & \uparrow \pi_Z^r \\
 \text{Im}(\pi_X) & \xleftarrow{\chi} & \text{Im}(\pi_Y)' & \xleftarrow{\chi''} & \text{Im}(\pi_Y) & \xleftarrow{\chi'} & \text{Im}(\pi_Z)' \xrightarrow{\quad} \text{Im}(\pi_Z) \\
 \uparrow \pi_X^l & & \uparrow & & \uparrow & & \uparrow \pi_Z^l \\
 I \times X & \xleftarrow{\text{id}_X \times \psi} & I \times Y & \xleftarrow{\text{id}_X \times \psi'} & I \times Y & \xrightarrow{\varphi \times \text{id}_Z} & J \times Z \xrightarrow{\varphi' \times \text{id}_Z} K \times Z \\
 \downarrow \pi_I & & \downarrow \pi_I & & \downarrow \pi_I & & \downarrow \pi_K \\
 I & \xlongequal{\quad} & I & \xrightarrow{\varphi} & J & \xlongequal{\quad} & J \xrightarrow{\varphi'} K
 \end{array}$$

To see that the functor is fibred, recall that a morphism $(\varphi, \psi) \in \mathcal{C} \times \mathcal{C}^{\text{op}}$ is cartesian iff ψ is an isomorphism. We can see then that $\text{id} \times \psi$ is also an isomorphism, as indeed must the induced χ be, as taking factorisations is functorial. Therefore the adaptor $(\psi, \text{id} \times \psi, \chi)$ is cartesian over ψ . $\square$

It would be ideal if the functor $\mathcal{C} \times \mathcal{C}^{\text{op}} \rightarrow \text{DAdt}(\mathcal{C})$ was fully faithful, but, unfortunately, neither is true.

Counterexample 6.2.6. The functor F from [Proposition 6.2.5](#) is not full in general. Take our category to be Set with the usual surjection/injection factorisation system. Suppose we have objects $(0, 0)$ and (J, Y) in $\mathcal{C} \times \mathcal{C}^{\text{op}}$ such that $Y \neq 0$. There is an adaptor $(!, \text{id}_0, \text{id}_0) : F(0, 0) \rightarrow F(J, Y)$, but there is no map $Y \rightarrow 0$, so there is no map $(0, 0) \rightarrow (J, Y) \in \text{Set} \times \text{Set}^{\text{op}}$.

$$\begin{array}{ccccc}
0 & \xrightarrow{\text{pink}} & 0 & \xrightarrow{!} & Y' \\
\pi_0^! \uparrow & & \parallel & & \uparrow \pi_Y^! \\
0 \times 0 & \xrightarrow{\text{blue}} & 0 \times Y & \xrightarrow{! \times id_Y} & J \times Y \\
\pi_0 \downarrow & & \parallel & \lrcorner & \downarrow \pi_J \\
0 & \xrightarrow{\text{orange}} & 0 & \xrightarrow{!} & J
\end{array}$$

Counterexample 6.2.7. The functor F from [Proposition 6.2.5](#) is not faithful. For example, in $\mathbf{Set}$ there are two morphisms $(id_0, id_2), (id_0, \text{swap}) : (0, 2) \rightarrow (0, 2)$, where 0 is the empty set and 2 is a set with two elements, and the morphism swap is the non-trivial permutation on 2 . However, F takes $(0, 2)$ to the cototal relation $R = 0 \xleftarrow{id} 0 \xrightarrow{id} 0$, and there is a unique morphism $R \rightarrow R$.

6.3 Strong Reductions are Adaptors

THE REASON for introducing all this machinery with adaptors is that we will be able to recover the notion of “strongly Weihrauch reducible” from them by proving a result analogous to [Theorem 3.5.3](#). First, we will reiterate the notion of strongly reducible from [Section 3.2](#).

Definition 6.3.1 (Strongly Weihrauch Reducible). Let $f : \subseteq X \rightrightarrows Y$ and $g : \subseteq Z \rightrightarrows W$ be Weihrauch problems. Then f is strongly Weihrauch reducible to g , written $f \leq_{sW} g$, if there are type-2 computable $\Phi, \Psi : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ such that, for all realisers $G \vdash g$, $\Psi \circ G \circ \Phi \vdash f$.

Arguing as in [Section 3.3](#), we can work with the realiser part of problems, thus reducing problems to multi-valued functions on $\mathbb{B}$ and with the following equivalent definition of reducibility (using the axiom of choice).

Definition 6.3.2 (Strongly Weihrauch Reducible, Equivalent). Suppose we have Weihrauch problems $f : \subseteq \mathbb{B} \rightrightarrows \mathbb{B}$ and $g : \subseteq \mathbb{B} \rightrightarrows \mathbb{B}$. Then f is strongly Weihrauch reducible to g , written $f \leq_{sW} g$, if there are type-2 computable functions $\Phi, \Psi : \subseteq \mathbb{B} \rightarrow \mathbb{B}$ such that $\Psi \circ g \circ \Phi \sqsubseteq f$.

Two problems f and g are strongly Weihrauch equivalent, written $f \equiv_{sW} g$, if $f \leq_{sW} g$ and $g \leq_{sW} f$. Equivalence classes of this relation are called Strong Weihrauch degrees.

Contrasting this with regular Weihrauch reducibility ([Definition 3.2.8](#)), we see that the backwards map depends on both the initial argument and the result of the oracle, whereas strong reducibility only depends on the latter. It is clear that if f is strongly reducible to g , then f is reducible to g ; the converse is not true ([Counterexample 3.2.14](#)).

Example 6.3.3. As an example of a strong reduction, recall from earlier our proof that $\text{LPO} \leq_W \text{KL}$ ([Example 3.2.12](#)). The backwards map in this proof only relies on the output of the oracle, so we actually have $\text{LPO} \leq_{sW} \text{KL}$.

We now turn to stating and proving our characterisation. As before, we require an “answerability” condition on cototal relations.

Definition 6.3.4 (Problem). Let $\mathcal{C}$ be a category with pullbacks and a factorisation system. Then an (answerable) problem, $R : X \rightarrow I$, is a cototal relation $X \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} I$ such that dom_R is an answerable container.

For the following theorem, we take $\mathcal{C}$ to be the category of projectives ReprSp with a factorisation system given by pullback-stable epimorphisms and regular monomorphisms.

Theorem 6.3.5. The Strong Weihrauch degrees are isomorphic to the posetal reflection of the category StrWei of answerable problems and dependent adaptors over represented spaces ReprSp .

Proof. We will give explicit constructions from answerable problems to Weihrauch problems, and vice versa, and prove these constructions are inverse up to isomorphism.

Adaptors to Weihrauch problems Fix a cototal relation $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ over ReprSp . We can define its Weihrauch problem $w(R)$ by $(i, x) \in w(R)$ iff there exists a $r \in R$ such that $\text{dom}_R(r) = i$ and $\text{cod}_R(r) = x$. Since dom_R is surjective, there is always such an x , and vice versa; in other words, $w(R)$ is total and cototal.

Now suppose we have a second cototal relation $J \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} Y$ and a dependent adaptor (represented by) (φ, ψ, χ) . That is, we have the following commuting diagram.

$$\begin{array}{ccccc}
 X & \xleftarrow{\chi} & \text{Im}(\text{cod}_S \circ \pi_S) & \hookrightarrow & Y \\
 \uparrow \text{cod}_R & & \uparrow \pi & & \uparrow \text{cod}_S \\
 R & \xleftarrow{\psi} & I \times_J S & \xrightarrow{\pi_S} & S \\
 \downarrow \text{dom}_R & & \downarrow \pi_I & \lrcorner & \downarrow \text{dom}_S \\
 I & \xlongequal{\quad} & I & \xrightarrow{\varphi} & J
 \end{array}$$

Firstly, since both π_I and dom_S are surjective, φ gives a type-2 computable map between the domains of $w(R)$ and $w(S)$.

Next, up to isomorphism, the pullback in the center of the diagram consists of the set of pairs $(i : I, s : S)$ such that $\varphi(i) = \pi_J(s)$, with $\langle a, b \rangle \Vdash_{I \times_J S} (i, j)$ iff $a \Vdash_I i$ and $b \Vdash_S j$. Recall from the definition of our factorisation system (Theorem 2.3.22), that we can take $\text{Im}(\text{cod}_S \circ \pi_S)$ to be the subset $\text{cod}_S(\pi_S(I \times_J S)) \subseteq Y$ with its realisability relation inherited from Y .

Now suppose that $(\varphi(i), y) \in w(S)$ with $s : S$ witnessing this relation. Then (i, s) is in $I \times_J S$ and $y = \text{cod}_S(s) = \text{cod}_S(\pi_S(i, s)) = \pi_Y(i, s)$ is in $\text{Im}(\text{cod}_S \circ \pi_S)$. Inspecting the top left rectangle, $\chi(y) = \text{cod}_R(\psi(i, s))$ is in X , and hence $\psi(i, s)$ witnesses $(i, \chi(y)) \in w(R)$. Thus, (φ, χ) is a strong Weihrauch reduction.

Weihrauch problems to adaptors Given a Weihrauch problem $f : \subseteq \mathbb{B} \rightrightarrows \mathbb{B}$, there are projections $\pi_{\text{dom}(f)} : \sum_{u \in \text{dom}(f)} f(u) \rightarrow \text{dom}(f)$ and $\pi_{\text{cod}(f)} : \sum_{u \in \text{dom}(f)} f(u) \rightarrow \text{cod}(f)$. $\text{dom}(f)$ and $\text{cod}(f)$ will be projectives with the

trivial reflexive realisability relations, whereas $\sum_{u \in \text{dom}(f)} f(u)$ will be a projective with the realisability relation $\langle e, e' \rangle \Vdash_{\text{dom}(c(f))} (e, e')$. These projections are evidently surjective and jointly monic, hence we can define a cototal relation $a(f) := \text{dom}(f) \xleftarrow{\pi_{\text{dom}(f)}} \sum_{u \in \text{dom}(f)} f(u) \xrightarrow{\pi_{\text{cod}(f)}} \text{cod}(f)$.

Now, suppose we have two Weihrauch problems $f, g : \subseteq \mathbb{B} \rightrightarrows \mathbb{B}$ such that $f \leq_{\text{SW}} g$. In particular, we have type-2 computable functions $\Phi : \text{dom}(f) \rightarrow \text{dom}(g)$ and $\Psi : \text{cod}(f) \rightarrow \text{cod}(g)$ witnessing the strong reduction. Pulling $\pi_{\text{cod}(g)}$ back along Φ , we obtain the pullback $\sum_{u \in \text{dom}(f)} g(\Phi(u))$. Then factoring the map $\sum_{u \in \text{dom}(f)} g(\Phi(u)) \rightarrow \text{cod}(g)$ through its image Im , we can define a map $\chi : \text{Im} \rightarrow \text{cod}(f)$ by composing the inclusion $\text{Im} \hookrightarrow \text{cod}(g)$ with Ψ . We can lift χ to define a map $\psi(u, v) := (u, \chi(v))$ that makes the following diagram commute.

$$\begin{array}{ccccc}
 & & \Psi & & \\
 & \swarrow & & \searrow & \\
 \text{cod}(f) & \xleftarrow{\chi} & \text{Im} & \hookrightarrow & \text{cod}(g) \\
 \uparrow \pi_{\text{cod}(f)} & & \uparrow & & \uparrow \pi_{\text{cod}(g)} \\
 \sum_{u \in \text{dom}(f)} f(u) & \xleftarrow{\psi} & \sum_{u \in \text{dom}(f)} g(\Phi(u)) & \longrightarrow & \sum_{v \in \text{dom}(g)} g(v) \\
 \downarrow \pi_{\text{dom}(f)} & & \downarrow & \lrcorner & \downarrow \pi_{\text{dom}(g)} \\
 \text{dom}(f) & \xlongequal{\quad} & \text{dom}(f) & \xrightarrow{\Phi} & \text{dom}(g)
 \end{array}$$

Thus we have an adaptor $(\Phi, \psi, \chi) : a(f) \rightarrow a(g)$.

a and w(R) are inverses Let $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ be a cototal relation in ReprSp . Since dom_R is an answerable container, it is surjective, and hence $\text{dom}(w(R)) = I$. Similarly, surjectivity of X tells us that $\text{cod}(w(R)) = X$.

The map $\langle \text{dom}_R, \text{cod}_R \rangle : R \rightarrow \sum_{i \in I} w(R)(i) \in \text{ReprSp}$ is a bijection: π_I and π_X are jointly monic, and it is surjective by definition. As it is a bijection, it has an inverse ψ in Set ; moreover, ψ is easily seen to be tracked by the identity.

All this is summarised by the following dependent adaptor, which is clearly an isomorphism.

$$\begin{array}{ccccc}
 X & \xlongequal{\quad} & \text{cod}(w(R)) & \xlongequal{\quad} & \text{cod}(w(R)) \\
 \uparrow \text{cod}_R & & \uparrow & & \uparrow \pi_{\text{cod}(w(R))} \\
 R & \xleftarrow{\langle \text{dom}_R, \text{cod}_R \rangle} & \sum_{i \in I} w(R)(i) & \xlongequal{\quad} & \sum_{i \in \text{dom}(w(R))} w(R)(\text{id}(i)) \\
 \downarrow \text{dom}_R & \xleftarrow{\psi} & \downarrow \pi_I & \lrcorner & \downarrow \pi_{\text{dom}(w(R))} \\
 I & \xlongequal{\quad} & I & \xlongequal{\quad} & \text{dom}(w(R))
 \end{array}$$

In the other direction, if $f : \subseteq X \rightrightarrows Y$ is a Weihrauch problem, we see that $w(a(f))$ is the problem such that $u \sim x$ iff $(u, x) : \sum_{u \in \text{dom } f} f(u)$, i.e., if $(u, x) \in f$. Thus $w \circ a$ is the identity on Weihrauch problems. $\square$

Remark 6.3.6. *The purpose of χ is to force ψ not to use the initial argument.*

As we already know, if there is a strong reduction between problems, then there is a reduction between them.

Corollary 6.3.7. *Let f, g be problems. If $f \leq_{sW} g$ then $f \leq_W g$.*

Proof. This follows immediately using the forgetful functor U from [Proposition 6.1.10](#). $\square$

6.4 Tensor Products of Adaptors

JUST AS WE proved that containers had a tensor product in [Section 4.1.2](#), so too do adaptors with a straightforward modification, provided that the classes of your factorisation system are also closed under products.

Definition 6.4.1 (Tensor Product $\otimes$). *Let $\mathcal{C}$ be a category with finite limits and a factorisation system $(\mathcal{L}, \mathcal{R})$, and fix cototal relation $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ and $J \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} Y$. If $\text{cod}_R \times \text{cod}_S \in \mathcal{L}$, then we define the tensor product of R and S , written $R \otimes S$, by*

$$I \times J \xleftarrow{\text{dom}_R \times \text{dom}_S} R \times S \xrightarrow{\text{cod}_R \times \text{cod}_S} X \times Y.$$

It is clear that $\text{dom}_R \times \text{dom}_S$ and $\text{cod}_R \times \text{cod}_S$ are jointly monic by projecting out the components. This does give a categorical product on the category of cototal relations $\text{RelCoTot}(\mathcal{C})$, however, we refer to this as a tensor as it will not be the product in the category of adaptors. We will meet that product in [Section 6.6.2](#). For now, we content ourselves with proving $\otimes$ gives a monoidal structure on the category of adaptors.

Proposition 6.4.2. *Let $\mathcal{C}$ be a category with finite limits and a factorisation system $(\mathcal{L}, \mathcal{R})$, where both classes are closed under products. Then $(\text{DA}(\mathcal{C}), \otimes, \mathcal{I})$ is a monoidal category, where the unit is defined by $\mathcal{I} := 1 \xleftarrow{id_1} 1 \xrightarrow{id_1} 1$. $(\text{DA}(\mathcal{C}), \otimes, \mathcal{I})$ is strict whenever $(\mathcal{C}, \times, 1)$ is.*

Proof. Let $(\varphi, \psi, \chi) : R \rightarrow S$ and $(\varphi', \psi', \chi') : R' \rightarrow S'$ be adaptors. We obtain an adaptor $(\varphi \times \varphi', \psi \times \psi', \chi \times \chi') : R \otimes R' \rightarrow S \otimes S'$ by superimposing one adaptor diagram on the other, and noting that products of pullbacks are pullbacks of products⁴³. This is clearly a bifunctor.

⁴³ Since limits commute with limits.

It is straightforward to lift the associator α and unitors λ, ρ to $(\mathcal{C}, \times, 1)$ to an associator and unitors for $(\text{DA}(\mathcal{C}), \otimes, \mathcal{I})$ (see [Figure 6.1](#)). It is also straightforward to lift the coherence laws. $\square$

We already saw in [Proposition 4.1.15](#) that the tensor product of containers over $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ gives us the parallel product on Weihrauch degrees. The following proposition is the obvious extension to adaptors.

Proposition 6.4.3. *Let $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ and $J \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} Y$ be problems in StrWei . Then*

$$w(R) \times w(S) \equiv_{sW} w(R \otimes S).$$

Proof. It is clear that $((i, j), (x, y)) \in w(R \otimes S)$ if and only if $(i, x) \in w(R)$ and $(j, y) \in w(S)$. $\square$

$$\begin{array}{ccccc}
(X \times Y) \times Z & \xlongequal{\quad} & (X \times Y) \times Z & \xrightarrow{\alpha_{X,Y,Z}} & X \times (Y \times Z) \\
(\text{cod}_R \times \text{cod}_S) \times \text{cod}_T \uparrow & & \uparrow & & \uparrow \text{cod}_R \times (\text{cod}_S \times \text{cod}_T) \\
(R \times S) \times T & \xlongequal{\quad} & (R \times S) \times T & \xrightarrow{\alpha_{R,S,T}} & R \times (S \times T) \\
(\text{dom}_R \times \text{dom}_S) \times \text{dom}_T \downarrow & & \downarrow & \lrcorner & \downarrow \text{dom}_R \times (\text{dom}_S \times \text{dom}_T) \\
(I \times J) \times K & \xlongequal{\quad} & (I \times J) \times K & \xrightarrow{\alpha_{I,J,K}} & I \times (J \times K)
\end{array}$$

(a) Associator $\alpha_{R,S,T} : (R \otimes S) \otimes T \rightarrow R \otimes (S \otimes T)$

Figure 6.1: Associators and unitors for adaptors.

$$\begin{array}{ccccc}
1 \times X & \xlongequal{\quad} & 1 \times X & \xrightarrow{\lambda_X} & X \\
\text{id}_1 \times \text{cod}_R \uparrow & & \uparrow & & \uparrow \text{cod}_R \\
1 \times R & \xlongequal{\quad} & 1 \times R & \xrightarrow{\lambda_R} & R \\
\text{id}_1 \times \text{dom}_R \downarrow & & \downarrow & \lrcorner & \downarrow \text{dom}_R \\
1 \times S & \xlongequal{\quad} & 1 \times S & \xrightarrow{\lambda_S} & S
\end{array}$$

(b) Left Unitor $\lambda_R : \mathcal{I} \otimes R \rightarrow R$

6.5 Cylinders

IN THIS SECTION, we will talk about a special kind of problem known as a cylinder. Cylinders are those problems which, in effect, allow you to “push a value through” them. They have the property that a problem f is Weihrauch reducible to a cylinder g if and only if it is strongly reducible to them. In this section, we will develop the basic theory of cylinders in $\text{pMod}(\mathbb{A}, \mathbb{A}')$ with the epi-regular-mono factorisation system. Without loss, we will also treat all projective modest sets as subsets of $\mathbb{A}$.

Definition 6.5.1 (id , Δ). *There is a problem $\text{id} : \subseteq \mathbb{A} \rightrightarrows \mathbb{A}$ which maps each element of the PCA to itself. We will denote the corresponding adaptor in $\text{pMod}(\mathbb{A}, \mathbb{A}')$ by $\Delta := \mathbf{a}(\text{id})$.*

In Weihrauch reducibility, a problem f is a cylinder¹⁴ whenever there is a reduction $\text{id} \times f \leq_{\text{sW}} f$. Since $f \leq_{\text{sW}} \text{id} \times f$ is always true, this means that $f \equiv_{\text{sW}} \text{id} \times f$.

¹⁴ The geometric analogy is hopefully clear.

Definition 6.5.2 (Cylinder). *Given a cototal relation $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$, its cylinder is defined to be $\text{Cyl}(R) := \Delta \otimes R$.*

R is called a cylinder if there is an adaptor $\text{Cyl}(R) \rightarrow R$.

Remark 6.5.3. *The following are easy to see:*

- there is an adaptor $R \rightarrow \text{Cyl}(R)$,
- $U(R) \equiv_W U(\text{Cyl}(R))$, and
- $\text{Cyl}(R)$ is a cylinder.

Theorem 6.5.4. *The Cyl functor on adaptors extends to a weak right adjoint to the forgetful functor $U : \text{DAdt}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$.*

Proof. Given a container $f : X \rightarrow I$, we can turn this into a cototal relation $I \xleftarrow{f} X \xrightarrow{\text{id}_X} X$, then define $\text{Cyl}(f)$ to be the cylinder of that relation. The operation of turning containers into adaptors is not itself functorial ([Remark 3.2.16](#)).

Let $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ be a cototal relation, $g : Y \rightarrow J$ be a container, and suppose we have an adaptor $(\varphi, \psi, \chi) : R \rightarrow \text{Cyl}(g)$. $\text{dom}(\psi)$ is not just a pullback of φ along $\text{id}_A \times g$, but a pullback of $\pi_J \circ \varphi$ along g . Hence there is a container morphism $\eta_{R,g} := (\pi_J \circ \varphi, \psi) : U(R) \rightarrow g$. Furthermore, this map is surjective: given any container morphism $(\varphi, \psi) : U(R) \rightarrow g$, the adaptor $(\iota_I, \varphi) : R \rightarrow \text{Cyl}(g)$ is sent to (φ, ψ) by this construction, where $\iota_I : I \hookrightarrow A$ is the inclusion into the ambient space.

A straightforward diagram chase confirms that $\text{Cont}(U(\alpha), g) \circ \eta_{S,g} = \eta_{R,g} \circ \text{DA}(\alpha, \text{Cyl}(g))$ for any relations R, S , container g and adaptor $\alpha : R \rightarrow S$. $\square$

Corollary 6.5.5. ¹⁵ Let $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ a cototal relation and $g : X \rightarrow I$ be a container. Then there is an adaptor $\text{Cyl}(R) \rightarrow \text{Cyl}(g)$ if and only if there is a container morphism $U(R) \rightarrow g$

¹⁵ Brattka and Gherardi, ‘Weihrauch Degrees, Omniscience Principles and Weak Computability’, Proposition 3.5.

Proof. First, apply [Theorem 6.5.4](#) in the case of an adaptor $\text{Cyl}(R) \rightarrow \text{Cyl}(g)$. Then compose with container morphisms witnessing the equivalence $U(R) \equiv_W U(\text{Cyl}(R))$. $\square$

Corollary 6.5.6. Let f be a container, $\text{Cyl}(f)$ is a cylinder.

Proof. An adaptor $\text{Cyl}(\text{Cyl}(f)) \rightarrow \text{Cyl}(f)$ exists iff there is a container morphism $U(\text{Cyl}(f)) \rightarrow f$, but this exists by [Remark 6.5.3](#). $\square$

With these results established, it only remains to prove that we obtain the expected Galois connection when we move to the posets of (strong) Weihrauch degrees.

Theorem 6.5.7. ¹⁶ There is a Galois connection between the poset of Weihrauch degrees and the poset of strong Weihrauch degrees given by the inclusion and cylindrification operators. More specifically, let $f : \subseteq X \rightrightarrows Y$, $g : \subseteq Z \rightrightarrows W$ be Weihrauch problems, then

¹⁶ *ibid.*, Corollary 3.6.

$$f \leq_W g \quad \text{if and only if} \quad f \leq_{sW} \text{id} \times g.$$

Proof. Given a Weihrauch problem f , we will write f' for the projection $\sum_{x \in X} Y(x) \rightarrow X$ and the corresponding container, and F for the cototal relation $X \xleftarrow{f'} \sum_{x \in X} Y(x) \xrightarrow{\text{id}} \sum_{x \in X} Y(x)$. Similarly, for a problem g , we will take g' and G to be the respective containers and cototal relations. Note that $U(F) = f'$ and $U(G) = g'$.

By [Theorem 3.5.3](#), we have a reduction $f \leq_W g$ iff and only if we have a container morphism $U(F) \rightarrow g'$. This morphism exists if and only if we have an adaptor $F \rightarrow \Delta \otimes G$ by [Theorem 6.5.4](#). Then using [Theorem 6.3.5](#) such an adaptor exists iff there is a strong reduction $w(F) \leq_{sW} w(\Delta \otimes G)$.

By construction, $(x, y) \in f$ iff $(x, y) \in \sum_{x \in X} Y(x)$ iff $(x, (x, y)) \in w(F)$, hence $f \equiv_{sW} w(F)$, and similarly $w(G) \equiv_{sW} g$. It follows that we have a reduction $f \leq_W g$ if and only if $f \equiv_{sW} w(F) \leq_{sW} w(\Delta \otimes G) \equiv_{sW} w(\Delta) \times w(G) \equiv_{sW} \text{id} \times g$. $\square$

6.6 The Strong Weihrauch Lattice

IN THIS FINAL SECTION, we will prove that the Strong degrees have a lattice structure. In [Section 4.1.1](#), this was a corollary of $\text{Cont}(\mathcal{C})$ being a distributive category. While it will be easy to show that $\text{DAdt}(\mathcal{C})$ has products, we will only be able to give *weak* coproducts. Like the construction of Dzhabarov¹⁷, our construction will rely on a completion operation, but ours will readily generalise to any PCA.

¹⁷ Dzhabarov, ‘Joins in the strong Weihrauch degrees’.

6.6.1 Preliminaries

IN ORDER TO give the lattice structure, we are going to make extensive use of having an extensive structure, so it will be useful to record some lemmas about coproducts in this setting.

The following result is straightforward and its proof is omitted.

Lemma 6.6.1. *If $f : R \rightarrow X$ and $g : S \rightarrow Y$ are (regular) epimorphisms, then so is $f + g : R + S \rightarrow X + Y$.*

What is less clear, is that a similar thing holds for monomorphisms. It is true for extensive categories, but first we need a small lemma.

Lemma 6.6.2. *Let $\mathcal{C}$ be an extensive category. If we have a map $f : R \rightarrow X + Y$, then there are objects R_X and R_Y , and an isomorphism $i : R_X + R_Y \cong R$, such that $f \circ i = f_X + f_Y : R_X + R_Y \rightarrow X + Y$ for some maps $f_X : R_X \rightarrow X$ and $f_Y : R_Y \rightarrow Y$.*

Proof. We pull f back along the coprojection $\iota_X : X \rightarrow X + Y$ (resp. $\iota_Y : Y \rightarrow X + Y$) to obtain the diagram below. The morphism i exists, and is an isomorphism, as the pullback of coprojections is a coproduct diagram¹⁸.

¹⁸ Carboni, Lack and Walters, ‘Introduction to extensive and distributive categories’, Proposition 2.2.

$$\begin{array}{ccccc}
 & & R_X + R_Y & & \\
 & \swarrow & \downarrow \sim i & \nwarrow & \\
 R_X & \xrightarrow{\quad} & R & \xleftarrow{\quad} & R_Y \\
 f_X \downarrow & \lrcorner & \downarrow f & \lrcorner & \downarrow f_Y \\
 X & \xrightarrow{\quad \iota_X \quad} & X + Y & \xleftarrow{\quad \iota_Y \quad} & Y
 \end{array}
 \quad \square$$

If we have two parallel maps $h, k : Z \rightarrow R + S$ into a sum, we would like to provide a common decomposition provided that both h and k always send elements of z to the same side of the sum. This is provided by following lemma.

Lemma 6.6.3. *Let $\mathcal{C}$ be a lexextensive category and suppose $h, k : Z \rightarrow R + S$ are parallel morphisms in $\mathcal{C}$ such that $(f + g) \circ h = (f + g) \circ k$ for some $f : R \rightarrow X$, $g : S \rightarrow Y$. Then there are objects Z_R and Z_S , an isomorphism $i : Z_R + Z_S \rightarrow Z$, and maps $h_R, k_R : Z_R \rightarrow R$ and $h_S, k_S : Z_S \rightarrow S$ such that $h \circ i = h_R + h_S$ and $k \circ i = k_R + k_S$.*

Proof. Suppose $h, k : Z \rightarrow R + S$ satisfy the hypothesis for some f and g . Applying [Lemma 6.6.2](#) to h then we obtain $i : Z_R + Z_S \rightarrow Z$, $h_R : Z_R \rightarrow R$

and $h_S : Z_S \rightarrow S$ such that $h_R + h_S = h \circ i$. We would like a similar decomposition for k , and it turns out that we can reuse the same i . Consider the following diagram:

$$\begin{array}{ccccc}
 & & Z_R + Z_S & & \\
 & \nearrow & \downarrow \sim i & \nwarrow & \\
 Z_R & \xrightarrow{\quad} & Z & \xleftarrow{\quad} & Z_S \\
 \downarrow k_R & \lrcorner & \downarrow k & \lrcorner & \downarrow k_S \\
 R & \xrightarrow{\quad \iota_R \quad} & R + S & \xleftarrow{\quad \iota_S \quad} & S \\
 \downarrow f & & \downarrow f+g & & \downarrow g \\
 X & \xrightarrow{\quad \iota_X \quad} & X + Y & \xleftarrow{\quad \iota_Y \quad} & Y
 \end{array}$$

As $(f + g) \circ k = (f + g) \circ h$, the pullback lemma tells us that Z_R is a pullback for h along ι_R , and similarly for Z_S . Hence, we have maps $h_R : Z_R \rightarrow R$ and $h_S : Z_S \rightarrow S$ such that $h \circ i = h_R + h_S$. $\square$

Corollary 6.6.4. *Let $\mathcal{C}$ be an extensive category. If both $f : R \rightarrow X$ and $g : S \rightarrow Y$ are (regular) monomorphisms, then $f + g : R + S \rightarrow X + Y$ is also a (regular) monomorphism.*

Proof. For monos, fix $h, k : Z \rightarrow R + S$ such that $(f + g) \circ h = (f + g) \circ k$. Then by Lemma 6.6.3 there is an isomorphism $i : Z_R + Z_S$ and maps $h_R, k_R : Z_R \rightarrow R$, $h_S, k_S : Z_S \rightarrow S$ satisfying $h \circ i = h_R + h_S$ and $k \circ i = k_R + k_S$. Thus, $(f + g) \circ (h_R + h_S) = (f + g) \circ (k_R + k_S)$. Since f is a monomorphism, $f \circ h_R = f \circ k_R$ implies $h_R = k_R$; similarly, g being a monomorphism implies $h_S = k_S$. It follows that $h = k$.

We omit the proof for regular monos, except to remark that, once again, Lemma 6.6.3 is the key step. $\square$

6.6.2 Products

WE HAVE ALREADY SEEN that $\text{Cont}(\mathcal{C})$ has products, which are defined by $f \times g := [f \times 1_J, 1_I \times g]$, for $f : X \rightarrow I$ and $g : Y \rightarrow J$. We will extend this to products over adaptors.

Theorem 6.6.5. *Let $\mathcal{C}$ be an extensive category with a factorisation system $(\mathcal{L}, \mathcal{R})$, such that both $\mathcal{L}$ and $\mathcal{R}$ are closed under sums. If $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ and $J \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} Y$ are cototal relations in $\text{DA}(\mathcal{C})$, then their product is defined by*

$$I \times J \xleftarrow{[\text{cod}_R \times \text{id}_J, \text{id}_I \times \text{cod}_S]} R \times J + I \times S \xrightarrow{x^I + y^J} \text{Im}(x) + \text{Im}(y),$$

where $x := \text{cod}_R \circ \pi_R : R \times J \rightarrow X$ and $y := \text{cod}_S \circ \pi_S : I \times S \rightarrow Y$.

Proof.

$\otimes$ is well-defined First, we have to check that $[\text{cod}_R \times \text{id}_J, \text{id}_I \times \text{cod}_S]$ and $x^I + x^J$ are jointly monic. Let $f, g : A \rightarrow R \times J + I \times S$ be satisfying

$$(x^I + y^I) \circ f = (x^I + y^I) \circ g, \text{ and}$$

$$[\text{cod}_R \times \text{id}_J, \text{id}_I \times \text{dom}_S] \circ f = [\text{cod}_R \times \text{id}_J, \text{id}_I \times \text{dom}_S] \circ g.$$

By [Lemma 6.6.3](#), we can assume A is a coproduct $A_1 + A_2$, that f is the sum $f_1 + f_2$ of maps $f_1 : A_1 \rightarrow R \times J$ and $f_2 : A_2 \rightarrow I \times S$, and likewise for $g = g_1 + g_2$. It follows that $x^l \circ f_1 = x^l \circ g_1$, which implies $\text{cod}_R \circ \pi_R \circ f_1 = \text{cod}_R \circ \pi_R \circ g_1$. We also have $(\text{cod}_R \times \text{id}_J) \circ f_1 = (\text{cod}_R \times \text{id}_J) \circ g_1$, hence $\text{cod}_R \circ \pi_R \circ f_1 = \text{cod}_R \circ \pi_R \circ g_1$, which as cod_R and dom_S are jointly monic, gives $\pi_R \circ f_1 = \pi_R \circ g_1$. But we also have that $\text{dom}_S \circ f_1 = \text{dom}_S \circ g_1$, and hence $f_1 = g_1$ by the universal property of products. An analogous argument establishes that $f_2 = g_2$, and hence $f = g$, as required.

Finally, we have a cototal relation, since $\mathcal{L}$ is closed under sums.

Defining projections We can define the projection adaptors by:

$$\begin{array}{ccccc}
 \text{Im}(x) + \text{Im}(y) & \xleftarrow{\iota_{\text{Im}(x)}} & \text{Im}(x) & \xleftarrow{x^r} & X \\
 \uparrow x^l + y^l & & \uparrow x^l & & \uparrow \text{dom}_R \\
 R \times J + I \times S & \xleftarrow{\iota_{R \times J}} & R \times J & \xrightarrow{\pi_R} & R \\
 \downarrow [\text{dom}_R \times \text{id}_J, \text{id}_I \times \text{dom}_S] & & \downarrow \text{cod}_R \times \text{id}_J & \lrcorner & \downarrow \text{cod}_R \\
 I \times J & \xlongequal{\quad} & I \times J & \xrightarrow{\pi_I} & I
 \end{array}$$

By definition, the lower half of these projector adaptors agrees with projections in $\text{Cont}(\mathcal{C})$.

Defining tupling Now, let $K \xleftarrow{\text{dom}_T} T \xrightarrow{\text{cod}_T} Z$ be a cototal relation, and $(\varphi, \psi, \chi) : T \rightarrow R$, $(\varphi', \psi', \chi') : T \rightarrow S$ be adaptors. To fix notation, we will write $x^\diamond$ for the map $\text{dom}_R \circ f^* \varphi : K \times_I R \rightarrow R \rightarrow X$, so $\text{dom}(\chi) = \text{Im}(x^\diamond)$, and similarly for $\text{dom}(\chi') = \text{Im}(y^\diamond)$. Then the following diagram gives an adaptor $(\langle \varphi, \varphi' \rangle, [\psi, \psi'], [\chi, \chi']) : T \rightarrow R \times S$.

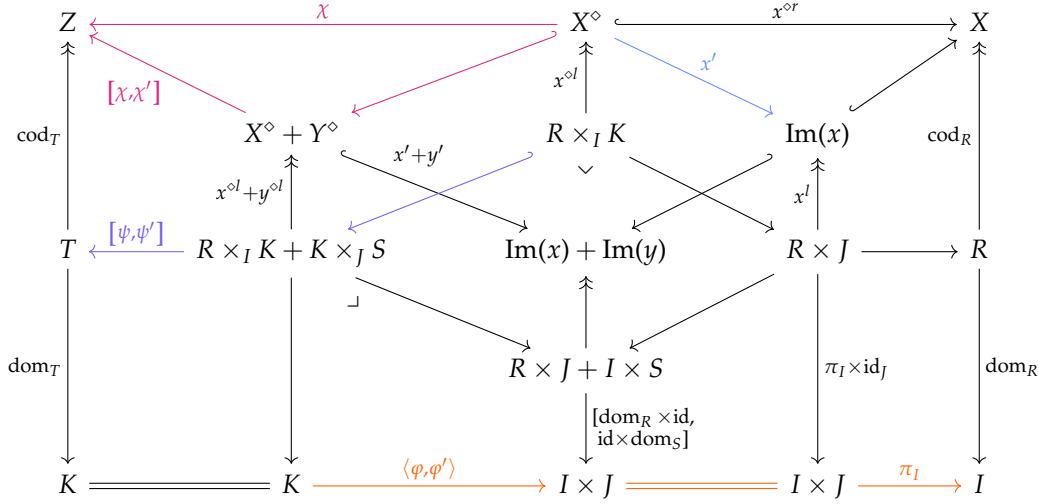
$$\begin{array}{ccccc}
 Z & \xleftarrow{[\chi, \chi']} & \text{Im}(x^\diamond) + \text{Im}(y^\diamond) & \xleftarrow{x^{\diamond r} + y^{\diamond r}} & \text{Im}(x) + \text{Im}(y) \\
 \uparrow \text{cod}_Z & & \uparrow x^{\diamond l} + y^{\diamond l} & & \uparrow x^l + y^l \\
 T & \xleftarrow{[\psi, \psi']} & R \times_I K + K \times_J S & \xrightarrow{\quad} & R \times J + I \times S \\
 \downarrow \text{dom}_T & & \downarrow & \lrcorner & \downarrow [\text{dom}_R \times \text{id}_J, \text{id}_I \times \text{dom}_S] \\
 K & \xlongequal{\quad} & K & \xrightarrow{\langle \varphi, \varphi' \rangle} & I \times J
 \end{array}$$

The diagram clearly commutes because the diagrams for (φ, ψ, χ) and (φ', ψ', χ') do. Both $x^{\diamond l} + y^{\diamond l} \in \mathcal{L}$ and $x^{\diamond r} + y^{\diamond r} \in \mathcal{R}$ by assumption.

Again, the lower half of this adaptor agrees with the tupling of the containers (φ, ψ) and (φ', ψ') . As such, we already know from [Proposition 4.1.6](#) that $(\langle \varphi, \varphi' \rangle, [\psi, \psi'])$ is the unique container c such that $(\pi_I, \iota_{R \times J}) \circ c = (\varphi, \psi)$ and $(\pi_J, \iota_{I \times S}) \circ c = (\varphi', \psi')$. All it remains to check is that everything works on the upper half of the adaptor.

Projections undo tupling Using the explicit definition of adaptor composition from [Remark 6.2.4](#), we obtain the following commuting diagram¹⁹.

¹⁹ To reduce clutter, I have suppressed the names of coprojections and projections from products and pullbacks.



The bottom half of the diagram is simply the diagram showing that projections undo tupling for containers. The only non-obvious morphism that needs addressing is $x' : \text{Im}(x^\diamond) \rightarrow \text{Im}(x)$ (and similarly for y'). This is the unique morphism (guaranteed by the factorisation system) that makes the following diagram commute.

$$\begin{array}{ccccc}
 R \times_J K & \xrightarrow{x^\diamond l} & \text{Im}(x^\diamond) & \xrightarrow{x^\diamond r} & X \\
 \downarrow & & \downarrow x' & & \parallel \\
 R \times J & \xrightarrow{x^l} & \text{Im}(x) & \xrightarrow{x^r} & X
 \end{array}$$

Uniqueness of $[\chi, \chi']$ It is clear that $[\chi, \chi']$ is the unique such arrow, as precomposing with a coprojection recovers χ or χ' . $\square$

By applying [Theorem 6.3.5](#) to the products we have just obtained in adaptors, we recover meets in the strong Weihrauch degrees.

Corollary 6.6.6. *Strong Weihrauch degrees have meets.*

Finally, mirroring the situation from [Proposition 4.1.8](#), there is a terminal adaptor, but it is not necessarily answerable.

Proposition 6.6.7. *Let $\mathcal{C}$ be an extensive category with a factorisation system $(\mathcal{L}, \mathcal{R})$. Then $\text{DAdt}(\mathcal{C})$ has a terminal object given by*

$$1 \xleftarrow{!} 0 \xlongequal{\quad} 0.$$

Proof. Fix a cototal relation $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$. Then $(\text{id}_1, !_R : 1 \rightarrow R, !_X : 1 \rightarrow X)$ is the unique adaptor to $1 \xleftarrow{!} 0 \xrightarrow{\text{id}_0} 0$. $\square$

6.6.3 Partial Map Classifiers

TO DEFINE THE COMPLETION of a space required for our construction of weak coproducts, we will make use of a specialisation of the concept of partial map classifier²⁰.

²⁰ The concepts of a partial map and their classifiers are standard, for example, see section A2.4 of the Elephant book (Johnstone, *Sketches of an Elephant: A Topos Theory Compendium*).

Definition 6.6.8 ((Regular) Partial Map). Let $\mathcal{C}$ be a category with pullbacks. A partial map $X \rightharpoonup Y$ is an isomorphism class²¹ of spans $X \xleftarrow{f} A \xrightarrow{g} Y$ such that f is a (regular) monomorphism.

Partial maps form a category $\text{Par}_{\mathcal{C}}$ with composition given by taking pullbacks of spans in the usual way. This construction will also work for regular partial maps in $\text{pAsm}(\mathbb{A}, \mathbb{A}')$ and $\text{pMod}(\mathbb{A}, \mathbb{A}')$, since regular monomorphisms are stable under pullback (Proposition A.3.5).

Definition 6.6.9 ((Regular) Partial Map Classifier). Let $\mathcal{C}$ be a category with pullbacks. An object $Y_{\perp}$ is a (regular) partial map classifier if and only if there is a (regular) monomorphism $\eta_Y : Y \rightarrow Y_{\perp}$ such that, for every (regular) partial map $X \xleftarrow{f} A \xrightarrow{g} Y$, there exists a unique map $g_{\perp}$ such that the following diagram is a pullback square:

$$\begin{array}{ccc} A & \xrightarrow{f} & X \\ g \downarrow & & \downarrow g_{\perp} \\ Y & \xrightarrow{\eta_Y} & Y_{\perp} \end{array}$$

Partial maps can be equivalently characterised by showing that partial maps are representable in the following sense.

Proposition 6.6.10. An object $Y_{\perp}$ is a (regular) partial map classifier if and only if the pullback functor η_Y^* is an isomorphism $\mathcal{C}(X, Y_{\perp}) \cong \text{Par}_{\mathcal{C}}(X, Y)$, which is natural in X .

The most obvious example of a category with partial map classifiers is Set with $X_{\perp} := X + 1$. More generally, this works for any Boolean extensive category. However, we also have *regular* classifiers in categories of assemblies.

Proposition 6.6.11. Let Y be an assembly over $\mathbb{A}$. The regular partial map classifier for $Y \in \text{Asm}(\mathbb{A})$ is given by $Y_{\perp} = Y \sqcup \{\perp\}$ with the realisability relation given by $e \Vdash_{Y_{\perp}} y \in Y$ iff $e \cdot () \downarrow$ and $e \cdot () \Vdash_Y y$, and $e \Vdash_{Y_{\perp}} \perp$ for every $e \in \mathbb{A}$.

Proof. We will take η_Y to be the inclusion map $Y \rightarrow Y_{\perp}$ (which is tracked by K). Now, suppose we have a partial map $X \xleftarrow{f} A \xrightarrow{g} Y$, then the map $g_{\perp} : X \rightarrow Y_{\perp}$ is given by:

$$g_{\perp}(x) = \begin{cases} g(x) & \text{if } x = f(a) \text{ for some } a \in A \\ \perp & \text{otherwise} \end{cases}$$

That this function is tracked is guaranteed by composing the trackers for η_Y and g with the effective witness i of the left-cancellativity of f given by Proposition 2.3.17.

It is easy to check that A is a pullback of $g_{\perp}$ along η_Y using the injectivity of η_Y and Proposition 2.3.17 to define trackers of the induced map. $\square$

This construction will not work for partitioned modest sets, because it relies on every name being a realiser for $\perp$. However, we will only require the weaker condition that the $g_{\perp}$ in Definition 6.6.9 exists (and isn't necessarily a pullback square). In this case, it is possible to obtain a suitable construction for projectives by adjoining more than one new element, in fact, by taking the whole PCA²².

²¹ Spans are isomorphic if there is an isomorphism between their apexes, making the obvious diagram commute.

²² As usual, we will make the convenient assumption that all assemblies are subsets of $\mathbb{A}$.

Proposition 6.6.12. *Let Y be an partitioned modest set over $\mathbb{A}$ and set $Y_\perp = \mathbb{A}$, which is equipped with the trivial reflexive realisability relation given by $e \Vdash_{Y_\perp} e$. There is a monomorphism $\eta_Y : Y \rightarrow Y_\perp$ such that, for every regular partial map $X \xleftarrow{f} A \xrightarrow{g} Y$, there exists a map $g_\perp$ such that the following diagram commutes:*

$$\begin{array}{ccc} A & \xrightarrow{f} & X \\ g \downarrow & & \downarrow g_\perp \\ Y & \xrightarrow{\eta_Y} & Y_\perp \end{array}$$

Proof. We will write e_x for the unique realiser of x , and set $\eta_Y(x) := K \cdot e_x$, which is tracked by K . Suppose we have a partial map $X \xleftarrow{f} A \xrightarrow{g} Y$. By [Remark 2.3.19](#), we may assume f is an inclusion of subsets tracked by I . Finally, we define $g_\perp(x) := K \cdot e_{g(x)} = K \cdot (t \cdot e_x)$, where t tracks g . $\square$

6.6.4 Weak Coproducts

ARMED WITH [PROPOSITION 6.6.12](#), we can turn to giving the construction of the weak coproduct. To match with Dzhamfarov, we denote weak coproducts with $\boxplus$, however, we should note that he calls his completion $\widetilde{X}$ rather than $X_\perp$ ²³.

Theorem 6.6.13. *$\text{pMod}(\mathbb{A}, \mathbb{A}')$ has weak coproducts, i.e., for every pair of objects $X, Y \in \text{pMod}(\mathbb{A}, \mathbb{A}')$, there is an object $X \boxplus Y$ and morphisms $\iota_X : X \rightarrow X \boxplus Y$, $\iota_Y : Y \rightarrow X \boxplus Y$, such that for any pair of morphisms $X \xleftarrow{f} Z \xrightarrow{g} Y$, there is a morphism $h : X \boxplus Y \rightarrow Z$ such that $h \circ \iota_X = f$ and $h \circ \iota_Y = g$.*

$$\begin{array}{ccccc} X & \xrightarrow{\iota_X} & X \boxplus Y & \xleftarrow{\iota_Y} & Y \\ & \searrow f & \downarrow h & \swarrow g & \\ & & Z & & \end{array}$$

Proof.

$\boxplus$ is well-defined. Let $I \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} X$ and $J \xleftarrow{\text{dom}_S} S \xrightarrow{\text{cod}_S} Y$ be cototal relations. We will take $R \times Y_\perp + X_\perp \times S$ to be the apex of $R \boxplus S$, and define the following maps:

$$\begin{aligned} d : R \times Y_\perp + X_\perp \times S &\rightarrow I + J \\ d &:= (\text{dom}_R \circ \pi_R) + (\text{dom}_S \circ \pi_S), \text{ and} \end{aligned}$$

$$\begin{aligned} c : R \times Y_\perp + X_\perp \times S &\rightarrow X_\perp \times Y_\perp \\ c &:= [(\eta_X \circ \text{cod}_R) \times \text{id}_{Y_\perp}, \text{id}_{X_\perp} \times (\eta_Y \circ \text{cod}_S)]. \end{aligned}$$

The map d will be the domain map of $R \boxplus S$. For the codomain leg, we factorise $c = c^r \circ c^l$ using the (pullback-preserving)-epi-regular-mono factorisation.

$$R \boxplus S := I + J \xleftarrow{d} R \times Y_\perp + X_\perp \times S \xrightarrow{c^l} \text{Im}(c)$$

²³In fact, since Johnstone uses $\widetilde{X}$ for the partial map classifier ([Johnstone, *Sketches of an Elephant: A Topos Theory Compendium*, Proposition 2.4.7](#)), there would have been a harmonious coincidence of notation. However, the author felt it is best not to overload the tilde a third time.

To see that $R \boxplus S$ is a valid relation, we need to show that it is jointly monic. Let $f, g : A \rightarrow R \times Y_\perp + X_\perp \times S$ such that $d \circ f = d \circ g$ and $c^l \circ f = c^l \circ g$. Without loss of generality, by [Lemma 6.6.3](#), we can assume that A , f and g have the following forms:

$$\begin{array}{lll} A = A_1 + A_2 & & f_1 = \langle f_1^1, f_1^2 \rangle : A_1 \rightarrow R \times Y_\perp \\ f = f_1 + f_2 & \text{where} & f_2 = \langle f_2^1, f_2^2 \rangle : A_2 \rightarrow X_\perp \times S \\ g = g_1 + g_2 & & g_1 = \langle g_1^1, g_1^2 \rangle : A_1 \rightarrow R \times Y_\perp \\ & & g_2 = \langle g_2^1, g_2^2 \rangle : A_2 \rightarrow X_\perp \times S \end{array}$$

Composing f and g with d gives two equations: $\text{dom}_R \circ f_1^1 = \text{dom}_R \circ g_1^1$ and $\text{dom}_S \circ f_2^2 = \text{dom}_S \circ g_2^2$. Similarly, composing f and g with $c = c^r \circ c^l$ yields four equations:

$$\begin{array}{ll} f_1^2 = g_1^2, & \eta_X \circ \text{cod}_R \circ f_1^1 = \eta_X \circ \text{cod}_R \circ g_1^1, \\ f_2^1 = g_2^1, & \eta_Y \circ \text{cod}_S \circ f_2^2 = \eta_Y \circ \text{cod}_S \circ g_2^2. \end{array}$$

Since, η_X is a monomorphism, $\text{cod}_R \circ f_1^1 = \text{cod}_R \circ g_1^1$. Because dom_R and cod_R are jointly monic, $f_1^1 = g_1^1$, and hence $f_1 = g_1$. A similar argument shows $f_2 = g_2$ and, $f = g$, as required.

Defining coprojections First, we factor the map $(\eta_X \circ \text{cod}_R) \times \text{id}_{Y_\perp}$ into $(\eta_X \times \text{id}_{Y_\perp}) \circ (\text{cod}_R \times \text{id}_{Y_\perp})$. Then, $\text{cod}_R \times \text{id}_{Y_\perp}$ is a pullback-stable epimorphism as the product of pullback-stable epimorphisms, and $\eta_X \times \text{id}_{Y_\perp}$ is a regular monomorphism as the product of regular monomorphisms²⁴. As such, we can define the coprojection $\iota_R : R \rightarrow R \boxplus S$ by the following adaptor, where the map $X \times Y_\perp \rightarrow \text{Im}(c)$ is induced by the factorisation, and is monic by [Lemma A.3.4](#). The other coprojection is defined similarly.

²⁴ As pullback-stable epimorphisms are just surjections, this follows by the same property in **Set**; for regular monomorphisms, this is follows from [Proposition 2.3.17](#).

$$\begin{array}{ccccc} & & \eta_X \times \text{id}_{Y_\perp} & \rightarrow & X_\perp \times Y_\perp \\ & & \nearrow & & \uparrow c^r \\ X & \xleftarrow{\pi_X} & X \times Y_\perp & \xrightarrow{\quad} & \text{Im}(c) \\ \uparrow \text{cod}_R & & \uparrow \text{cod}_R \times \text{id}_{Y_\perp} & & \uparrow c^l \\ R & \xleftarrow{\pi_R} & R \times Y_\perp & \xrightarrow{\quad} & R \times Y_\perp + X_\perp \times S \\ \downarrow \text{dom}_I & & \downarrow \text{dom}_R \circ \pi_R & \perp & \downarrow (\text{dom}_R \circ \pi_R) + (\text{dom}_S \circ \pi_S) \\ I & \xlongequal{\quad} & I & \xrightarrow{\iota_I} & I + J \end{array} \quad (\text{coproj})$$

Defining cotupling Let $K \xleftarrow{\text{dom}_T} T \xrightarrow{\text{cod}_T} Z$ be a third cototal relation, and suppose we have adaptors $(\varphi, \psi, \chi) : R \rightarrow T$ and $(\varphi', \psi', \chi') : S \rightarrow T$. Using the epi-regular-mono factorisation of $I \times T \xrightarrow{\pi_T} T \xrightarrow{\text{cod}_T} Z$, we obtain $i \circ z : I \times T \rightarrow \text{Im}(\text{cod}_T \circ \pi_T) \hookrightarrow Z$, where z is the corestriction of $\text{cod}_T \circ \pi_T$ to its image, and i is an inclusion. The following diagram shows explicitly the adaptor for $R \rightarrow T$; for the similar adaptor $S \rightarrow T$ we use primes to distinguish objects and maps from those shown here.

$$\begin{array}{ccccc}
 X & \xleftarrow{\chi} & \text{Im}(\text{cod}_T \circ \pi_T) & \xrightarrow{i} & Z \\
 \uparrow \text{cod}_R & & \uparrow z & & \uparrow \text{cod}_T \\
 R & \xleftarrow{\psi} & I \times_K T & \xrightarrow{\pi_T} & T \\
 \downarrow \text{dom}_R & & \downarrow \pi_I & \lrcorner & \downarrow \text{dom}_T \\
 I & \xlongequal{\quad} & I & \xrightarrow{\varphi} & K
 \end{array}$$

We need to create a third adaptor $R \boxplus S \rightarrow T$. Since we are in a lex-tensive category, $I \times_K T + I \times_K T$ is a pullback of dom_T along $[\varphi, \varphi']$. Next, we observe that $\text{cod}_T \circ [\pi_T, \pi'_T] = [i, i'] \circ (z + z')$, and that while $z + z'$ is epi (as a coproduct of epi's), $[i, i']$ is not necessarily a regular mono. As such, we factor $[i, i']$ into $j \circ q$, where $q = [q_1, q_2] : \text{Im}(\text{cod}_T \circ \pi_T) \twoheadrightarrow \text{Im}(\text{cod}_T \circ \pi_T) \cup \text{Im}(\text{cod}_T \circ \pi_T)$ quotients the disjoint union into the regular union²⁵, and the maps q_1, q_2, j are all inclusions. This will give us right half of the adaptor diagram.

²⁵ Note that both are subsets of Y .

For the top-left quarter of the adaptor diagram, we require a morphism $\text{Im}(\text{cod}_T \circ \pi_T) \cup \text{Im}(\text{cod}_T \circ \pi_T) \rightarrow X_\perp \times Y_\perp$, which is a tuple of maps into $X_\perp$ and $Y_\perp$ respectively. For the former, using [Proposition 6.6.12](#), it suffices to have a partial map from the union to X . Using the partial map $x := \text{Im}(\text{cod}_T \circ \pi_T) \cup \text{Im}(\text{cod}_T \circ \pi_T) \xleftarrow{q_1} \text{Im}(\text{cod}_T \circ \pi_T) \xrightarrow{\chi}$, we obtain $x_\perp : \text{Im}(\text{cod}_T \circ \pi_T) \cup \text{Im}(\text{cod}_T \circ \pi_T) \rightarrow X_\perp$; symmetrically for Y , we obtain a partial map y by taking the span of the inclusion q_2 and χ' . Thus $\langle x_\perp, y_\perp \rangle$ is the required map.

We also require a map $I \times_K T + J \times_K T \rightarrow R \times Y_\perp + J \times_K T$, which we will take to be the sum $\langle \psi, y_\perp \circ q_1 \circ z \rangle + \langle x_\perp \circ q_2 \circ z', \psi' \rangle$. We have the equality $x_\perp \circ q_1 \circ z = \eta_X \circ \chi \circ z = \eta_X \circ \text{cod}_R \circ \psi$ by [Proposition 6.6.12](#) and the adaptor in [Equation \(coproj\)](#). It follows that $\langle x_\perp, y_\perp \rangle \circ q_1 \circ z = ((\eta_X \circ \text{cod}_R) \times \text{id}_{Y_\perp}) \circ \langle \psi, y_\perp \circ q_1 \circ z \rangle$. Analogous reasoning shows that $\langle x_\perp, y_\perp \rangle \circ q_2 \circ z = (\text{id}_{X_\perp} \times (\eta_Y \circ \text{cod}_S)) \circ \langle x_\perp \circ q_2 \circ z, \psi' \rangle$.

This is all summarised by the following commuting diagram.

$$\begin{array}{ccccc}
 X_\perp \times Y_\perp & \xleftarrow{\langle x_\perp, y_\perp \rangle} & \text{Im}(\text{cod}_T \circ \pi_T) \cup \text{Im}(\text{cod}_T \circ \pi_T) & \xrightarrow{j} & Z \\
 \uparrow c^r & & \uparrow q & & \uparrow \text{cod}_T \\
 \text{Im}(c) & \xleftarrow{b} & \text{Im}(\text{cod}_T \circ \pi_T) + \text{Im}(\text{cod}_T \circ \pi'_T) & \xrightarrow{[i, i']} & Z \\
 \uparrow c^l & & \uparrow z+z' & & \uparrow \text{cod}_T \\
 R \times Y_\perp + X_\perp \times S & \xleftarrow{\langle \psi, y_\perp \circ q_1 \circ z \rangle + \langle x_\perp \circ q_2 \circ z', \psi' \rangle} & I \times_K T + J \times_K T & \xrightarrow{[\pi_T, \pi'_T]} & T \\
 \downarrow d & & \downarrow \pi_I + \pi_J & \lrcorner & \downarrow \text{dom}_T \\
 I + J & \xlongequal{\quad} & I + J & \xrightarrow{[\varphi, \varphi']} & K
 \end{array}$$

Finally, it remains to note that the dashed arrow b exists by functoriality of factorisations, and hence we have an adaptor $([\varphi, \varphi'], \langle \psi, y_\perp \circ q_1 \circ z \rangle + \langle x_\perp \circ q_2 \circ z', \psi' \rangle, b) : R \boxplus S \rightarrow T$.

Weak initiality This follows by straightforward diagram chase. $\square$

From the existence of weak coproducts in $\text{pMod}(\mathbb{A}, \mathbb{A}')$, we can use our characterisation of strong degrees [Theorem 6.3.5](#) to conclude that joins exist.

Corollary 6.6.14. ²⁶ *The strong Weihrauch degrees have joins.*

²⁶ Dzhaferov, ‘Joins in the strong Weihrauch degrees’, Theorem 3.8.

Categories of dependent adaptors also have an initial object.

Proposition 6.6.15. *Let $\mathcal{C}$ be a lexextensive category with a factorisation system $(\mathcal{L}, \mathcal{R})$. Then there is an initial object given by*

$$0 \equiv 0 \xrightarrow{!} \text{Im}(!),$$

where $!^r \circ !^l$ is the factorisation of the unique map $! : 0 \rightarrow 1$.

Proof. Fix a cototal relation $X \xleftarrow{\text{dom}_R} R \xrightarrow{\text{cod}_R} I$. The unique adaptor $0 \rightarrow R$ is given by the following commuting diagram. The map χ exists by functoriality of factorisations.

$$\begin{array}{ccccc}
 & & 1 & \xleftarrow{!} & \\
 & \uparrow \scriptstyle !^r & \downarrow & & \\
 \text{Im}(!) & \xleftarrow{\chi} & \text{Im}(\text{cod}_R \circ !)^{\circ} & \xrightarrow{\quad} & X \\
 \uparrow \scriptstyle !^l & & \uparrow \scriptstyle ! & & \uparrow \scriptstyle \text{cod}_R \\
 0 & \xrightarrow{\quad} & 0 & \xrightarrow{!} & R \\
 \parallel & & \parallel & \lrcorner & \downarrow \scriptstyle \text{dom}_R \\
 0 & \xrightarrow{\quad} & 0 & \xrightarrow{!} & I
 \end{array}$$

Corollary 6.6.16. ²⁷ *The strong Weihrauch degrees form a lattice with a bottom element.*

²⁷ *ibid.*, Corollary 3.11.

Remark 6.6.17. *There is no distributivity, even weakly. If we could obtain a weakly distributive category of adaptors, then [Theorem 6.3.5](#) would allow us to conclude that the Strong degrees are a distributive lattice. However, this conflicts with a theorem of Dzhaferov ²⁸.*

²⁸ *ibid.*, Theorem 4.4.

Finally, we note that the fibration $\text{DAdt}(\mathcal{C}) \rightarrow \mathcal{C}$ has *fibred* sums, meaning that its type theory has dependent sums, however we omit the proof²⁹.

²⁹ Since it doesn't really have anything to do with Weihrauch reducibility.

7 Conclusion

7.1 Weihrauch Reducibility à la Carte

IN THIS THESIS, we argued that Weihrauch reducibility can be usefully understood as an instance of the theory of Containers. Accordingly, we stated theorem requirements in terms of properties of the base category $\mathcal{C}$, and derived the corresponding results on Weihrauch reducibility as a corollary, using a small number of transfer theorems. As a consequence, we may vary the PCA, and we will discuss this in [Section 7.2](#).

The minimum structure we assumed was that $\mathcal{C}$ has pullbacks. This is already sufficient to define categories of containers ([Proposition 3.4.5](#)) and answerable containers ([Definition 3.5.1](#)), and to characterise three variants of Weihrauch reducibility: vanilla ([Theorem 3.5.3](#)), continuous ([Theorem 3.5.4](#)) and extended reducibility [Theorem 3.6.3](#).

If we further assume that $\mathcal{C}$ has finite limits, it becomes possible to define a monoidal product $\otimes$ on containers ([Proposition 4.1.12](#)). Since $\otimes$ is stable under answerability, the tensor product of containers specialises to the usual parallel product of problems ([Proposition 4.1.15](#)), with all the expected properties ([Proposition 4.1.16](#)).

Generalising $\mathcal{C}$ to be lexextensive further allows us to define a distributive category of containers ([Proposition 4.1.6](#)). Coproducts, initial objects and products are all stable under answerability, but terminal objects are not ([Proposition 4.1.8](#)). Moreover, these coproducts distribute and products distribute linearly with $\otimes$ ([Proposition 4.1.13](#)). From these we obtain the usual lattice of Weihrauch degrees ([Corollary 4.1.9](#)).

Next, if we assume that $\mathcal{C}$ is also locally cartesian closed, or is the full subcategory of regular projectives of a locally cartesian closed category, we can obtain a useful notion of composition of containers. In the former case, we can define container functors ([Definition 4.2.1](#)), and container composition $\star$ ([Definition 4.2.3](#)) corresponds to composition of container functors ([Proposition 4.2.4](#)). In the latter case, it is still possible to define a weak composition product $\star_p$ ([Definition 4.2.10](#)). We use the latter to recover the composition of Weihrauch degrees ([Proposition 4.2.12](#)).

Staying with LCCCs for the moment, we can define a category of fibred polynomial functors ([Proposition 5.2.2](#)) and give many examples ([Lemmas 5.2.4](#) and [5.2.5](#)). If we require a full $\Pi\mathcal{WM}$ -category, it is possible to prove the existence of three fixpoints for fibred polynomial functors ([Theorem 5.2.8](#)), and that fixpoints of multi-variable fibred polynomial functors yield another fibred polynomial functor ([Theorem 5.2.16](#)).

In this framework, we can define analogues of the (in)finite parallelisation (Definitions 5.3.4 and 5.3.6), diamond (Definition 5.3.9) and inverse limit (Definition 5.3.13) operations on Weihrauch degrees. The diamond operator is the well-known free monad construction (Theorem B.1) and induces an interesting induction axiom on degrees (Theorem 5.3.11).

Using the examples of fibred polynomial functors from Lemma 5.2.4, it is possible to give a grammar of ζ -expressions (Definition 5.4.1), where every expression denotes a fibred polynomial functor. Such expressions are answer-trivial (Lemma 5.4.5), but in combination with the answerable part of a problem (Definition 5.4.7), it becomes possible to express interesting Weihrauch problems as closed expressions. In this thesis, we only give proofs that LPO (Proposition 5.4.10), closed choice (Proposition 5.4.12) and WKL (Proposition 5.4.13) can be expressed this way; however, every example in Figure 5.5 can be found in Pradic & Price¹, alongside an automata model justifying that such functors restrict to answerable containers.

In addition to Weihrauch reducibility, we can accommodate other reducibility theories in a category-theoretic framework. While we generally only stated theorems for vanilla Weihrauch reducibility, it is possible to get similar theorems for continuous reducibility and extended reducibility² by substituting Theorem 3.5.4 or Theorem 3.6.3 for Theorem 3.5.3.

Recall that we defined the category of Medvedev reductions by $\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})^{\text{op}}$. Taking the opposite of a category requires no hypothesis on the base category $\mathcal{C}$, per se, but to obtain its lattice structure, we require $\mathcal{C}$ to be lextensive. The shape fibration requires only pullbacks to be defined, but to define its right adjoint K we require that $\mathcal{C}$ has an initial object (Proposition 4.3.3). In this case, the order-reversing embedding of the Medvedev degrees into the Weihrauch degrees (Proposition 4.3.5) will preserve the lattice structure Remark 4.3.6. Finally, if $\mathcal{C}$ has finite limits, then we also have an order-preserving embedding (Corollary 4.3.8).

For a category $\mathcal{C}$ with pullbacks and a factorisation system, we can define the category of dependent adaptors $\text{DA}(\mathcal{C})$ (Definition 6.2.1) and to characterise the strong Weihrauch degrees (Theorem 6.3.5). There is a forgetful functor $\text{DA}(\mathcal{C}) \rightarrow \text{Cont}(\mathcal{C})$ (Proposition 6.1.10), which corresponds to the observation that $f \leq_{\text{sw}} g$ implies $f \leq_{\text{w}} g$. If $\mathcal{C}$ has finite limits and the factorisation system is stable under products, we can define tensor products (Definition 6.4.1), which again correspond to parallel products of degrees (Proposition 6.4.3). If $\mathcal{C}$ is lextensive and the factorisation system is stable under sums, we can define products (Theorem 6.6.5), and hence the strong degrees have meets (Corollary 6.6.6). The cylindrification Galois connection (Corollary 6.5.6 Theorem 6.5.7) can be recovered in such a setting if we have an object $\top$ into which every object embeds in a sensible way.

Finally, if we also assume that $\mathcal{C}$ has a weak regular partial map classifier (Proposition 6.6.12), then it is possible to define a weak coproduct (Theorem 6.6.13), from which we can conclude that the strong degrees are a lattice (Corollary 6.6.16).

The table in Figure 7.1 summarises the correspondence between the category-theoretic objects and computable analysis objects.

¹ Pradic and Price, ‘Problems with Fix-points of Polynomials of Polynomials’, §5.

² For extended reducibility, we should generally expect a little extra, for example, the extended degrees are a bounded lattice (Corollary 4.1.7).

	Category Theory	Computable Analysis
Categories	$\mathbf{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})^{\text{op}}$ $\text{Cont}_{\text{Ans}}(\mathbf{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$ $\text{Cont}_{\text{Ans}}(\mathbf{pMod}(\mathcal{K}_2))$ $\text{Cont}_{\text{Ans}}(\mathbf{pAsm}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$ $\mathbf{DAdt}(\mathbf{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$	Medvedev Degrees Weihrauch Degrees Continuous Weihrauch Degrees Extended Weihrauch Degrees Strong Weihrauch Degrees
Operators on Weihrauch Degrees	Product $p \times q$ Coproduct $p + q$ Tensor $p \otimes q$ Composition $p \star_p q$ Finite Parallelisation p^* Infinite Parallelisation $\hat{q}$ Free Monad on p Infinite Iteration p^∞	Meet $p \sqcap q$ Join $p \sqcup q$ Parallel Product $p \times q$ Composition $p \star q$ Finite Parallelisation p^* Infinite Parallelisation $\hat{q}$ Diamond operator $p^\diamond$ Inverse Limit p^∞
Operators on Strong Degrees	Tensor $p \otimes q$ Cylinder $\text{Cyl}(p)$ Products $p \times q$ Weak Coproducts $p \boxplus q$	Parallel Product $p \times q$ $p \times \text{id}$ Meet $p \sqcap q$ Join $p \sqcup q$

7.2 Other Weihrauch Reducibilities

AS PREVIOUSLY MENTIONED, the recipe outlined in [Section 7.1](#) specialises to Weihrauch reducibility because rpReprSp satisfies all the properties we ask for. However, none of these category-theoretic properties depended on the specific choice of filtered PCA $\mathcal{K}_2^{\text{rec}} \subseteq \mathcal{K}_2$. We could equally well use any other filtered PCA $\mathcal{A}' \subseteq \mathcal{A}$ and derive an algebra of $\mathcal{A}$ -Weihrauch reductions. Similarly, we should expect to be able to express natural $\mathcal{A}$ -degrees using the ζ -expressions of [Section 5.4](#), although we should point out that certain things are simplified in the type-2 setting, since $X \mapsto X^{\mathbb{N}}$ preserves regular projectives (See [Remark 2.4.11](#)).

We now discuss several PCAs which would be interesting to explore using our framework to see how it compares to their existing reducibility theories. The obvious place to start is with type-1 computability $\mathcal{K}_1$ ([Example 2.1.14](#)). In our framework, a Turing reduction $A \leq_T B$ will correspond to a $\mathcal{K}_1$ -Weihrauch reduction ³ $A \leq B^\diamond$, a weak truth-table reduction ⁴ $A \leq_{\text{wtt}} B$ to a $\mathcal{K}_1$ -Weihrauch reduction $A \leq B^*$ and a truth-table reduction to a strong $\mathcal{K}_1$ -Weihrauch reduction $A \leq B^*$.

Another possibility is to explore transfinite computation⁵ using Hamkin and Lewis' infinite-time Turing machines⁶ or Koepke's ordinal Turing machines⁷. We know of no Weihrauch-style reductions for the former, but the latter has been studied by Carl⁸ and by Galeotti & Nobrega⁹. We conjecture that Carl's $A \leq_{\text{OTM}} B$ is related to reductions $A \leq B + 1$ for a filtered PCA of ordinal Turing machines, although the situation is a bit different because he uses single-valued functions.

A final PCA that would be useful to investigate is Scott's graph model¹⁰ $\mathcal{P}(\mathbb{N})$. After discussion with Andrej Bauer, we believe the enumeration degrees¹¹ are related to $\mathcal{P}(\mathbb{N})$ -Weihrauch reductions. Weihrauch-style

Figure 7.1: Dictionary linking category-theoretic objects on the left to computable analysis objects on the right.

³ Compare the treatment in Kihara, 'Lawvere-Tierney topologies for computability theorists'; Swan, 'Oracle modalities'.

⁴ Rogers, *Theory of Recursive Functions and Effective Computability*, Chapter 8.

⁵ The author was asked about transfinite computation at Computability in Europe, and was happy to report that "it just works".

⁶ Hamkins and Lewis, 'Infinite Time Turing Machines'.

⁷ Koepke, 'Turing Computations on Ordinals'.

⁸ Carl, 'Generalized Effective Reducibility', 'Complexities of Effective Reductions with Ordinal Turing Machines'.

⁹ Galeotti and Nobrega, 'Towards Computable Analysis on the Generalised Real Line'.

¹⁰ Scott, 'Data Types as Lattices'.

¹¹ Friedberg and Rogers jr., 'Reducibility and Completeness for Sets of Integers'.

reductions in the enumeration degrees have recently been considered by Soskova and Vidrine ¹².

Not every interesting notion of computation is a PCA. In particular, PCAs are not suited for discussing computation with finite states¹³. Another difficulty is polynomial-time computation. Polynomial-time reductions are a staple of computability theory and were introduced to the Weihrauch literature by Kawamura and Cook ¹⁴. Such reductions have been used, e.g., to develop the theory of integrable functions ¹⁵. Since we don't have a PCA of polynomial-time machines, we would need to work directly with properties of the category of polynomial-time functions. One such category, investigated by Kawamura and Pauly ¹⁶, is not cartesian closed. Given an appropriate category, the recipe of the previous section can be followed; it would be an amusing irony if polynomial functors¹⁷ couldn't capture polynomial-time Weihrauch reductions!

Other frameworks based on Weihrauch-style reducibilities and which don't obviously come from a PCA are due to Fujiwara ¹⁸, Sanders ¹⁹, and Kajiwaka et al. ²⁰ Fujiwara studies a primitive recursive variant in an intuitionistic setting. Sanders' N -reducibility is a theory for third-order statements. Kajiwaka et al. investigated total Weihrauch reducibility to compare subtoposes of the Grayson topos.

7.3 A Modest Proposal

IT IS A TRUISM in category theory that “It is better to have a good category with bad objects than a bad category²¹ with good objects”²². Given how well-behaved assemblies and modest sets are, the category theorist would be inclined to define Weihrauch reducibility using $\text{Cont}(\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$ rather than $\text{Cont}_{\text{Ans}}(\text{pMod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$. This raises the question of why the latter is the one used in computable analysis.

The focus on projectives is understandable, since computability theorists are generally interested in studying functions based on their intensional, rather than their extensional, behaviour. This results in more identifications than working extensionally, e.g., LPO and the problem of deciding equality on Sierpiński space are Weihrauch equivalent, but they are not equivalent in $\text{Cont}(\text{Mod}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}}))$ ²³. Also, as we've mentioned previously, it is possible that functional versions of a problem are not computable, but the multi-valued version is. Working with projectives allows us to work with multi-valued functions by working with the function on realisers instead.

As for requiring containers to be answerable, perhaps this is an area where they might reconsider. While it might seem unnatural to allow for “questions without answers”²⁴, it may improve the algebraic structure. The most obvious example is that there is now a natural top element, namely, the problem with a single question and a single answer. Another is that the implication of degrees ²⁵ would now be defined totally. This would follow from observing that implication is the same as the left adjoint to $\star$, which is known to exist for containers ^{26,27}. We also believe this would also make the partial degrees a Heyting algebra, contrary to

¹² Soskova and Vidrine, *Enumeration Weihrauch Reducibility*.

¹³ If the PCA is not trivial, it is possible to encode all the natural numbers by distinct elements (van Oosten, *Realizability: An Introduction to its Categorical Side*, §1.3).

¹⁴ Kawamura and Cook, ‘Complexity Theory for Operators in Analysis’.

¹⁵ Steinberg, ‘Complexity theory for spaces of integrable functions’.

¹⁶ Kawamura and Pauly, ‘Function Spaces for Second-Order Polynomial Time’.

¹⁷ Note, the two uses of polynomial are almost entirely unrelated.

¹⁸ Fujiwara, ‘Weihrauch and constructive reducibility between existence statements’.

¹⁹ Sanders, *A Notion of Reducibility for Third-Order Arithmetic*.

²⁰ Kajiwaka, Kaku, Kihara and Nakata, *Modified realizability subtoposes and total Weihrauch reducibility*.

²¹ In the author's opinion, the category of topological spaces is probably the canonical example.

²² This is usually attributed to Alexandre Grothendieck, but I have not found a citation for it. See MathOverflow for a discussion (Moskovich, *What is the source of this famous Grothendieck quote?*).

²³ In Mod , there is no injection from Sierpiński space to Cantor space.

²⁴ The author feels that some questions in life are too important to have answers.

²⁵ Brattka and Pauly, ‘On the algebraic structure of Weihrauch degrees’, §3.3.

²⁶ Spivak, *A reference for categorical structures on Poly*, §5, (68).

²⁷ Niu and Spivak, *Polynomial Functors: A Mathematical Theory of Interaction*, Proposition 6.57.

Higuchi & Pauly²⁸. However, because you can take the answerable part of a partial problem to obtain a problem, partial degrees will likely only serve an algebraic purpose.

A final thing worth addressing is categories of assemblies. The category $\text{Asm}(\mathcal{K}_2, \mathcal{K}_2^{\text{rec}})$ corresponds to what are known as multi-represented spaces, which were introduced in Schröder’s PhD Thesis²⁹. While he recognised that this category has excellent closure properties, e.g., admitting representations for *quotients* of countably-based spaces, they are not as frequently studied in the computable analysis literature. One reason, remarked by Rauzy³⁰, is that even if you fix a numbered subbasis on a topological space “there still are several, all seemingly natural, but sometimes non-equivalent, ways to define a multi-representation associated to this numbered subbasis”. Nevertheless, this structure has made assemblies, in the guise of Bauer’s extended degrees³¹, a popular target for category-theoretic treatments of Weihrauch reducibility (see Section 1.2). It remains to be seen if reductions between extended degrees will see as much attention as the non-extended degrees. So far, the only work we are aware of in this direction is that of Kihara and his collaborators³², and of Abou Samra & Madore³³.

7.4 Directions for Further Work

WHILE WE HAVE MANAGED to recast much of the survey paper of Brattka et al.³⁴ in a category-theoretic setting, there is, as always, much more that could have been done. In this section, we will discuss promising avenues to explore relating to other operations on degrees and containers, as well as listing some of the limitations of the current techniques.

7.4.1 Other Degree-Theoretic Operations

THE MAIN ALGEBRAIC OPERATIONS we did not cover were implication $\rightarrow$, box sum $\boxplus$ for *containers*, and the sum of degrees $+$. We already discussed implication in the previous section. The box sum of problems $p \boxplus q$ is known to be Weihrauch-equivalent to the join³⁵ $p \sqcup q$. The sum of problems is more interesting, and it corresponds to the operation³⁶ “given a p -question and a q -question, respond with a possible answer to both, one of which must be a valid answer”. As we saw when discussing weak co-products of adaptors, this can be accomplished by using a weak partial map classifier to give completions of a space. We believe that a similar technique will work for containers.

Other operators on the Weihrauch degrees include stashing³⁷ and the various “parts” of a degree, such as the finite³⁸, first-order³⁹ and deterministic⁴⁰ parts. Stashing may be expressible as a fixed point using the sum operation $+$ and the techniques of Section 5.3⁴¹. We conjecture that the finite and first-order parts can be defined for any locally cartesian closed category with a well-behaved natural numbers object, and that the deterministic part can be defined for any PCA.

²⁸ Higuchi and Pauly, ‘The degree-structure of Weihrauch-reducibility’, Theorem 4.1.

²⁹ Schröder, ‘Admissible Representations for Continuous Computations’.

³⁰ Rauzy, ‘Multi-representation associated to the numbering of a subbasis and formal inclusion relations’.

³¹ Which we characterised using assemblies in Theorem 3.6.3.

³² Kajikawa, Kaku, Kihara and Nakata, *Modified realizability subtoposes and total Weihrauch reducibility*; Kihara, ‘Lawvere-Tierney topologies for computability theorists’, *Rethinking the notion of oracle: A prequel to Lawvere-Tierney topologies for computability theorists*.

³³ Abou Samra and Madore, *An order-reversing embedding of Turing degrees into Arthur-Nimue-Merlin degrees*.

³⁴ Brattka, Gherardi and Pauly, ‘Weihrauch Complexity in Computable Analysis’.

³⁵ Dzharafarov, ‘Joins in the strong Weihrauch degrees’, Proposition 3.12.

³⁶ This is similar to the multiplicative disjunction $\wp$ of linear logic.

³⁷ Brattka, ‘Stashing And Parallelization Pentagons’.

³⁸ Cipriani and Pauly, ‘Embeddability of graphs and Weihrauch degrees’.

³⁹ Dzharafarov, Solomon and Yokoyama, ‘On the first-order parts of problems in the Weihrauch degrees’; Soldà and Valenti, ‘Algebraic properties of the first-order part of a problem’.

⁴⁰ Goh, Pauly and Valenti, ‘Finding Descending Sequences Through Ill-Founded Linear Orders’.

⁴¹ As Vasco Brattka pointed out to the author, all of the operators considered there are closure operators. We believe this is a happy accident and that interior operators like stashing can also be expressed

7.4.2 Other Category-Theoretic Operations

ONE OPERATOR ON CONTAINERS that we didn't consider was McBride's derivative operation ⁴². This operation is interesting because it is not a functor on $\text{Cont}(\mathcal{C})$, but arises from an adjunction on the subcategory of containers and *cartesian* morphisms. In terms of reductions, a cartesian Weihrauch reduction is one for which the backwards map is an isomorphism; in effect, all the work is done by the forward map. The author made a brief investigation of the derivative of a degree, but his attempt was not degree-theoretic⁴³ and was quickly abandoned. We know of no use of “cartesian” reductions in the computability literature.

Another interesting avenue of investigation would be to use Day convolution ⁴⁴ to generate new operators on degrees. Day convolution allows the category of functors on a monoidal category to inherit that structure. Many of the monoidal structures on Poly have already been shown to be instances of Day convolution ⁴⁵, and we should expect a similar situation on categories of assemblies.

7.4.3 Limitations of Current Techniques

WHILE WE WERE generally able to recover the expected laws of operations on degrees, e.g., that the Weihrauch degrees form a lattice (Corollary 4.1.9) and that parallel products behave correctly Proposition 4.1.16, we did not derive the expected laws for composition $\star_p$. For example, it is easy to establish associativity of $\star$ by using associativity of functor composition and Lemma 3.4.8, but it is less obvious how to do it for $\star_p$. The other laws below can be derived in a similar manner to Lemma 4.2.13.

$$\begin{array}{ll} (f \star g) \star h \cong f \star (g \star h) & f \star \text{id}_1 \cong f \cong \text{id}_1 \star f \\ f \star (g + h) \cong (f \star g) + (f \star h) & f \star 0 \cong 0 \\ f \star (g \times h) \cong (f \star g) \times (f \star h) & f \star 1 \cong 1 \end{array}$$

The current proof that denotations of ζ -expressions give genuine degrees (Remark 5.4.9) relies on an automata model which is particular to $\mathbb{B}$. It is desirable to have a proof that is purely category-theoretic, or at least to investigate what conditions are required of categories of projective modest sets $\text{pMod}(\mathbb{A}, \mathbb{A}')$ for that theorem to be true. One possible strategy would be to develop the theory of fibred polynomial quasi-functors and redo our theorems on fixpoints in that setting.

While we have been able to characterise many interesting degrees in Figure 5.5, it is not clear if we can fully characterise problems expressible using ζ -expressions. For example, can we characterise Ramsey degrees such as RT_2^2 ? What is clear is that we can only characterise problems below $(\Sigma_2^0)_k\text{-FindWS}$, for some $k \in \mathbb{N}$. To be sure, this is a lot, but it excludes problems such as⁴⁶ $\Sigma_3^0\text{-FindWS}$ and Borel determinacy $\Delta_1^1\text{-FindWS}$.

⁴² Abbott, Altenkirch, Ghani and McBride, ‘Derivatives of Containers’.

⁴³ Thanks to Arno Pauly for an example of this.

⁴⁴ Day, ‘On closed categories of functors’.

⁴⁵ Spivak, *A reference for categorical structures on Poly*.

⁴⁶ To our knowledge, these kinds of problems have not been investigated for the Weihrauch lattice, although Borel determinacy was studied by Friedman (Friedman, ‘Higher set theory and mathematical practice’) in the pre-history of reverse mathematics.

A Category Theory

THIS APPENDIX lists definitions and results that go beyond what can be understood as “basic category theory”, which we interpret to mean “goes beyond what is found in Mac Lane ¹”. In particular, we give useful results about coalgebras, factorisation systems and fibrations.

¹ Mac Lane, *Categories for the working mathematician*.

A.1 Basics

PREORDERS AND POSETS can be regarded as categories whose objects are elements of the preorder and with a unique morphism between two objects x and y if and only if $x \leq y$. Posets form a reflective subcategory of the category of small categories $\mathbf{Cat}$, which means that the inclusion of posets into categories has a left adjoint functor $\mathcal{C} \mapsto \mathcal{C}/\leftrightarrow$. $\mathcal{C}/\leftrightarrow$ is the quotient of $\mathcal{C}$ where objects X and Y are equivalent (written $X \equiv Y$) when there exists morphisms $X \rightarrow Y$ and $Y \rightarrow X$.

A *subobject*² of $C \in \mathcal{C}$ is an equivalence class $[i : A \hookrightarrow C]_{\sim}$ of monomorphisms, where two monomorphisms are equivalent $i \sim j$ if there is an isomorphism h such that $i \circ h = j$. The subobjects of C form a poset $\text{Sub}(C)$. By putting constraints on the category $\mathcal{C}$ it is possible to give that poset more structure, such as meets. In particular, it is possible to do *regular logic*³ if $\mathcal{C}$ is a regular category.

² Johnstone, *Sketches of an Elephant: A Topos Theory Compendium*, A1.3.1.

Definition A.1.1 (Regular Category). ⁴ A category $\mathcal{C}$ is regular if

- it has finite limits,
- the kernel pair⁵ of a morphism f has a coequaliser, and
- the pullback f^*g of a regular epimorphism $f : X \rightarrow Z$ along any morphism $g : Y \rightarrow Z$ is again a regular epimorphism.

³ Regular logic has truth $\top$, conjunction $\wedge$ and existential quantification $\exists$.

⁴ *ibid.*, §A1.3.

⁵ The kernel pair of f are the projections from the pullback of f along itself.

Definition A.1.2 (Regular Functor). A functor between regular categories is regular if it preserves finite limits and regular epimorphisms.

One particularly important concept in this thesis is that of a locally cartesian closed category.

Definition A.1.3 (Locally Cartesian Closed Category). ⁶ A category with pullbacks $\mathcal{C}$ is locally cartesian closed⁷ if $f^* : \mathcal{C}/Y \rightarrow \mathcal{C}/X$ has a right adjoint $\prod_f$ for every morphism $f : X \rightarrow Y \in \mathcal{C}$.

⁶ *ibid.*, Corollary A1.5.3.

⁷ Which we will frequently shorten to “ $\mathcal{C}$ is an LCCC”.

Remark A.1.4. It is conventional to assume that a locally cartesian closed category has a terminal object 1 , in which case, it is cartesian closed by the isomorphism $\mathcal{C} \cong \mathcal{C}/1$. Every LCCC we consider will have a terminal object.

A.2 (Co)algebras

IN THIS SECTION we give the definitions of algebras and coalgebras for an endofunctor, as well as prove a characterisation for dependent W -types that we use in [Section 5.2](#).

Definition A.2.1 (F -Algebra). *An algebra for an endofunctor $F : \mathcal{C} \rightarrow \mathcal{C}$ (an F -algebra), is a pair $(A, \mathfrak{a})$, where A is an object of $\mathcal{C}$ and $\mathfrak{a} : F(A) \rightarrow A$ is a map. We refer to A as the carrier of the algebra, and $\mathfrak{a}$ as its structure map.*

Definition A.2.2 (Algebra Morphisms). *Given two F -algebras, $(A, \mathfrak{a})$, $(B, \mathfrak{b})$, an F -algebra morphism $f : (A, \mathfrak{a}) \rightarrow (B, \mathfrak{b})$ is a map $f : A \rightarrow B$ in the category, such that the following diagram commutes.*

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

Convention A.2.3. *We will use $\mathsf{mathfrak{a}}$ for the names of (co)algebras and their structure maps.*

F -algebras and their morphisms form a category $\mathsf{Alg}(F)$ with identity and composition being inherited from the base category. If an F -algebra is initial in $\mathsf{Alg}(F)$, we will denote it by (μ_F, in_F) , and will write $\langle \mathfrak{a} \rangle$ for the unique morphism $(\mu_F, \mathsf{in}_F) \rightarrow (A, \mathfrak{a})$. The following lemma is frequently useful when dealing with F -algebras.

Lemma A.2.4 (Lambek's Lemma). ⁸ *If (μ_F, in_F) is an initial F -algebra, then in_F is an isomorphism.*

⁸ Lambek, 'A fixpoint theorem for complete categories'.

Proof. Notice that $(F(\mu_F), F(\mathsf{in}_F))$ is also an F -algebra, and hence there is a unique algebra map $! : \mu_F \rightarrow F(\mu_F)$. Trivially, $\mathsf{in}_F : F(\mu_F) \rightarrow \mu_F$ is itself an algebra morphism. Composing these, we have that $\mathsf{in}_F \circ ! = \mathsf{id}_{\mu_F}$ by initiality. Finally since $!$ is an algebra map,

$$! \circ \mathsf{in}_F = F(\mathsf{in}_F) \circ F(!) = F(\mathsf{id}_{\mu_F}) = \mathsf{id}_{F(\mu_F)}. \quad \square$$

By working in the opposite category $\mathcal{C}^{\mathsf{op}}$, we obtain dual notions of coalgebra and coalgebra morphism.

Definition A.2.5 (Coalgebra (morphism)). *Given an endofunctor $F : \mathcal{C} \rightarrow \mathcal{C}$, an F -coalgebra is an F^{op} -algebra in $\mathcal{C}^{\mathsf{op}}$. Similarly, an F -coalgebra morphism is an F^{op} -algebra morphism.*

Likewise, F -algebras and their morphisms form a category $\mathsf{CoAlg}(F)$ with identity and composition being inherited from the base category. If an F -coalgebra is terminal in $\mathsf{CoAlg}(F)$, we will denote it by (ν_F, out_F) , and will write $\llbracket \mathfrak{c} \rrbracket$ for the unique morphism $(\mu_F, \mathsf{in}_F) \rightarrow (C, \mathfrak{c})$. Analogously to [Lemma A.2.4](#), out_F is an isomorphism.

We will be interested in categories that have inductive and coinductive types, as given by initial algebras and terminal coalgebras for polynomial functors.

Definition A.2.6 ($\mathcal{W}/\mathcal{M}$ -types). A category $\mathcal{C}$ is said to have $\mathcal{W}$ -types⁹ (resp. $\mathcal{M}$ -types) if, for every container P of $\text{Cont}(\mathcal{C})$, $\llbracket P \rrbracket : \mathcal{C} \rightarrow \mathcal{C}$ has an initial algebra (resp. a terminal coalgebra).

⁹ W for “well-founded”.

We will also use the dependent generalisation of this definition.

Definition A.2.7 (Dependent $\mathcal{W}/\mathcal{M}$ -types). A category $\mathcal{C}$ has dependent $\mathcal{W}$ -types (resp. $\mathcal{M}$ -types) if, for every object $I \in \mathcal{C}$ and indexed container $(P, t, u) \in \text{Cont}(\mathcal{C})(I, I)$, the functor $\llbracket P, t, u \rrbracket : \mathcal{C}/I \rightarrow \mathcal{C}/I$ has an initial algebra (resp. a terminal coalgebra).

If the category you are working in has a terminal object then having dependent $\mathcal{W}/\mathcal{M}$ -types strictly includes having $\mathcal{W}/\mathcal{M}$ -types.

Definition A.2.8 ($\Pi\mathcal{WM}$ -category). A $\Pi\mathcal{WM}$ -category is an extensive locally cartesian closed category which also has dependent $\mathcal{W}$ and $\mathcal{M}$ -types.

Example A.2.9. Set and ReprSp are $\Pi\mathcal{WM}$ -categories¹⁰.

¹⁰ Bauer, ‘The realizability approach to computable analysis and topology’, Theorems 1.3.8 & 1.3.10.

For technical reasons, we will want to define a strengthened notion of dependent $\mathcal{W}$ -types. Before we give it, we require the following definition and proposition.

Definition A.2.10 (Compatible Functors, Companion).¹¹ Let $G : \mathcal{C} \rightarrow \mathcal{C}$ and $G' : \mathcal{C}' \rightarrow \mathcal{C}'$ be two endofunctors. A functor $F : \mathcal{C} \rightarrow \mathcal{C}'$ is (G, G') -compatible if there exists a natural transformation $\gamma : F \circ G \rightarrow G' \circ F$. We will call γ the companion.

¹¹ Bonchi, Petrişan, Pous and Rot, ‘Coinduction up-to in a fibrational setting’, Definition 2.

Compatible functors have desirable closure properties¹², but we will not need these in this thesis.

¹² *ibid.*, Prop. 1.

Proposition A.2.11. Let $f : I \rightarrow J$ be a map in $\mathcal{C}$ and $F : \mathcal{C}/J \rightarrow \mathcal{C}/J$, $G : \mathcal{C}/I \rightarrow \mathcal{C}/I$ be endofunctors with initial algebras (μ_F, in_F) and (μ_G, in_G) . If f^* is (F, G) -compatible with companion γ , then there is a canonical G -algebra morphism $\mu_G \rightarrow f^* \mu_F$.

Proof. Take the pullback along f of in_F and composing with γ gives a G -algebra, then use that μ_G is the initial G -algebra. $\square$

The strengthening of dependent $\mathcal{W}$ -types we will need in this thesis is the following.

Definition A.2.12. A category $\mathcal{C}$ has strong dependent $\mathcal{W}$ -types if it has dependent $\mathcal{W}$ -types and if whenever we have $f : I \rightarrow J$, polynomial endofunctors F and G , and a natural isomorphism $\alpha : f^* \circ F \cong G \circ f^*$, then the canonical morphism from [Proposition A.2.11](#) is an isomorphism.

Fortunately, this strengthening holds in $\Pi\mathcal{WM}$ -categories.

Lemma A.2.13. Let $\mathcal{C}$ be a locally cartesian closed category. Then $\mathcal{C}$ has dependent $\mathcal{W}$ -types if and only if it has strong dependent $\mathcal{W}$ -types.

Proof. It is immediate that if $\mathcal{C}$ has strong dependent $\mathcal{W}$ -types then it has dependent $\mathcal{W}$ -types.

Now, fix a $f : I \rightarrow J$ be a map, and let $F : \mathcal{C}/J \rightarrow \mathcal{C}/J$ and $G : \mathcal{C}/I \rightarrow \mathcal{C}/I$ be polynomial endofunctors. We call the assumed natural

transformation $\alpha : f^* \circ F \cong G \circ f^*$ and denote the adjunction $f^* \dashv \Pi_f$ by $\phi_{h,j} : \mathcal{C}/I(f^*h, j) \cong \mathcal{C}/J(h, \Pi_f j)$.

To prove the claim, we will show that $(f^*\mu_F, f^*\text{in}_F \circ \alpha_{\mu_F}^{-1})$ is also an initial G -algebra. It will then immediately follow that the unique G -algebra morphism $f^*\mu_F \rightarrow \mu_G$ is the inverse to the map from [Proposition A.2.11](#).

Suppose we have a G -algebra $(A, a : GA \rightarrow A)$, then $a \circ G\epsilon \circ \alpha$ is a map $f^*F\Pi_f A \rightarrow A$. Applying ϕ gives us an F -algebra with carrier $\Pi_f A$, and hence we have a F -algebra morphism $\langle \phi(a \circ G\epsilon \circ \alpha) \rangle$ from μ_F to $\Pi_f A$. Applying ϕ^{-1} gives a map¹³ $\beta := \epsilon \circ f^*(\langle \Pi_f(a \circ G(\epsilon_A) \circ \alpha) \circ \eta \rangle) : f^*\mu_F \rightarrow A$, which the following commuting diagram confirms is a G -algebra morphism.

¹³ Recall that the adjunction isomorphism ϕ is related to the unit η and counit ϵ by $\phi(g) = \Pi_f(g) \circ \eta$ and $\phi^{-1}(h) = \epsilon \circ f^*h$.

$$\begin{array}{ccccccc}
 Gf^*\mu_F & \xrightarrow{\alpha_{\mu_F}^{-1}} & f^*F\mu_F & \xrightarrow{f^*\text{in}_F} & f^*\mu_F & & \\
 \downarrow Gf^*\langle \phi(a \circ G(\epsilon_A) \circ \alpha) \rangle & & \downarrow f^*F\langle \phi(a \circ G(\epsilon_A) \circ \alpha) \rangle & & \downarrow f^*\langle \phi(a \circ G(\epsilon_A) \circ \alpha) \rangle & & \\
 Gf^*\Pi_f A & \xrightarrow{\alpha^{-1}} & f^*F\Pi_f A & \xrightarrow{f^*\eta} & f^*\Pi_f f^*F\Pi_f A & \longrightarrow & f^*\Pi_f A \\
 \downarrow G(\epsilon_A) & & \searrow \text{id} & & \downarrow \epsilon_{(f^*F\Pi_f A)} & & \downarrow \epsilon_A \\
 GA & & & & f^*F\Pi_f A & \xrightarrow{a \circ G(\epsilon_A) \circ \alpha} & A \\
 & & \swarrow a & & & & \uparrow
 \end{array}$$

It only remains to check that β is unique. However, if there was a $\beta' : f^*\mu_F \rightarrow A$ such that $a \circ G(\beta') = \beta' \circ f^*\text{in}_F \circ \alpha^{-1}$, then we could apply Π_f to that identity and derive the following commuting diagram showing that $\delta := \phi(\beta')$ is a F -algebra morphism from in_F to $\phi(a \circ G\epsilon \circ \alpha)$. Since there is a unique such morphism, we must have that $\delta = \langle \phi(a \circ G\epsilon \circ \alpha) \rangle_F$ and $\beta = \phi^{-1}(\delta) = \beta'$.

$$\begin{array}{ccccccc}
 & & F(\delta) & & & & \\
 & \swarrow & & \searrow & & & \\
 F\Pi_f A & \xleftarrow{F\Pi_f \beta'} & F\Pi_f f^*\mu_F & \xleftarrow{F\eta} & F\mu_F & \xrightarrow{\text{in}_F} & \mu_F \\
 \downarrow \eta & & & & \downarrow \eta & & \downarrow \eta \\
 & \rightarrow & \Pi_f f^*FA & \xleftarrow{\Pi_f f^*F\delta} & \Pi_f f^*F\mu_F & \xrightarrow{\Pi_f f^*\text{in}_F} & \Pi_f f^*\mu_F \\
 & & \downarrow \Pi_f \alpha & & \downarrow \Pi_f \alpha & & \downarrow \Pi_f \alpha \\
 & & \Pi_f Gf^*A & \xleftarrow{\Pi_f Gf^*\delta} & \Pi_f Gf^*\mu_F & & \Pi_f \beta' \\
 & & \downarrow \Pi_f G\epsilon & & \downarrow \Pi_f G\beta' & & \downarrow \Pi_f \beta' \\
 & & GA & \xrightarrow{a} & A & & A
 \end{array}$$

$\phi(\beta') := \delta$

□

A.3 Factorisation Systems

IN THE CATEGORY OF SETS $\mathbf{Set}$, any function $f : X \rightarrow Y$ factors through its image into a surjection followed by an inclusion $X \xrightarrow{f} \text{Im}(f) \hookrightarrow Y$. The category-theoretic generalisation of this situation is a factorisation system. A useful introductory note on factorisations was written by Riehl ¹⁴.

¹⁴ Riehl, *Factorization Systems*.

Definition A.3.1 (Factorisation System). A factorisation system $(\mathcal{L}, \mathcal{R})$ for a category $\mathcal{C}$ consists of two classes of morphisms $(\mathcal{L}, \mathcal{R})$ such that:

Closure $\mathcal{L}$ and $\mathcal{R}$ are closed under composition;

Isos Both $\mathcal{L}$ and $\mathcal{R}$ contain all isomorphisms of $\mathcal{C}$;

Factoring Every morphism f can be factored as $f = r \circ l$ for some morphisms $l \in \mathcal{L}$ and $r \in \mathcal{R}$; and

Functoriality Given a commuting rectangle $g \circ r \circ l = r' \circ l' \circ f$, where $l, l' \in \mathcal{L}$ and $r, r' \in \mathcal{R}$, there exists a unique map m , such that $m \circ l = l' \circ f$ and $r' \circ m = g \circ r$.

$$\begin{array}{ccccc} X & \xrightarrow{l} & Y & \xrightarrow{r} & Z \\ \downarrow f & & \downarrow m & & \downarrow g \\ X' & \xrightarrow{l'} & Y' & \xrightarrow{r'} & Z' \end{array}$$

Convention A.3.2. For a factorisation system $(\mathcal{L}, \mathcal{R})$, we will generalise the convention that epimorphisms are denoted by double-headed arrows $\twoheadrightarrow$ to arrows of $\mathcal{L}$. Likewise, we will use the hooked arrows of monomorphisms $\hookrightarrow$ for arrows of $\mathcal{R}$.

Convention A.3.3. Given a fixed choice of factorisations, we will decompose an arrow $f : X \rightarrow Y$ into $f^r \circ f^l$ with $f^l \in \mathcal{L}$ and $f^r \in \mathcal{R}$. We will also write $\text{Im}(f)$ for the domain of f^r in this factorisation, or just Im when there is no chance of confusion.

In $\mathbf{Set}$, a composition $g \circ f$ is surjective if and only if g is surjective. Similarly, $g \circ f$ is injective if and only if f is injective. This situation generalises to any factorisation system.

Lemma A.3.4. Let $\mathcal{C}$ be a category with a factorisation system $(\mathcal{L}, \mathcal{R})$. If $f : X \twoheadrightarrow Y \in \mathcal{L}$, $g : Y \rightarrow Z$ and $h : Z \rightarrow W \in \mathcal{R}$, then

- $g \circ f \in \mathcal{L}$ if and only if $g \in \mathcal{L}$, and
- $h \circ g \in \mathcal{M}$ if and only if $h \in \mathcal{M}$.

Proof. If $g \in \mathcal{L}$ then $g \circ f \in \mathcal{L}$ since $\mathcal{L}$ is closed under composition. Conversely, if $g \circ f \in \mathcal{L}$ then by functoriality of factorisations, there is a unique isomorphism i such that the following diagram commutes.

$$\begin{array}{ccccc} X & \xrightarrow{f} & Y & \xrightarrow{g} & Z \\ \parallel & & \downarrow i & & \parallel \\ X & \xrightarrow{g \circ f} & Z & \xlongequal{\quad} & Z \end{array}$$

Then $g = i$ is an isomorphism, hence $g \in \mathcal{L}$, and, since $\mathcal{L}$ is closed under composition, $g \circ f \in \mathcal{L}$.

The proof of second claim is similar. $\square$

Similarly, we know that monomorphisms are stable under pullback, and this is true for the right class $\mathcal{R}$.

Proposition A.3.5. *Let $\mathcal{C}$ be a category with a factorisation system $(\mathcal{L}, \mathcal{R})$. The class $\mathcal{R}$ is stable under pullbacks.*

Proof. Let $f : X \rightarrow Z$, $g : Y \rightarrow Z$ be morphisms in $\mathcal{C}$ and take the pullback of these. We then apply the factorisation system to the map $g^*f = r \circ l$ with $l \in \mathcal{L}$, $r \in \mathcal{R}$. The factorisation system induces the arrow i in the diagram below.

$$\begin{array}{ccc}
 P & \xrightarrow{f^*g} & X \\
 \downarrow l & & \parallel \\
 \text{Im} & \xrightarrow{i} & X \\
 \downarrow r & & \downarrow f \\
 Y & \xrightarrow{g} & Z
 \end{array}$$

(A large bracket on the left of the diagram is labeled g^*f)

It is easily seen that the lower square of the diagram is also a pullback of f along g and the claim holds. $\square$

Definition A.3.6 (Stable Factorisation System). *A factorisation system $(\mathcal{L}, \mathcal{R})$ is stable if $\mathcal{L}$ is stable under arbitrary pullbacks.*

A.4 Fibrations

A FIBRATION is a generalisation of the notion of fibre bundle from topology, and captures the notion of collections of objects varying over a base category. Fibrations are useful in category-theoretic logic for modelling logical theories over a particular context. The reader is referred to the first chapter of Jacobs¹⁵ and the lecture notes of Streicher¹⁶ for more information on Fibrations.

Definition A.4.1 (Cartesian, Fibration).¹⁷ *Let $p : \mathbb{E} \rightarrow \mathbb{B}$ be a functor. A morphism $f : X \rightarrow Y$ in $\mathbb{E}$ is above a morphism $u : I \rightarrow J$ in $\mathbb{B}$ if $p(f) = u$. f is cartesian over u if*

- f is above u , and
- for every $g : Z \rightarrow Y$ in $\mathbb{E}$ such that $p(g) = u \circ w$, for some $w : p(Z) \rightarrow I$, there is a unique $h : Z \rightarrow X$ satisfying $f \circ h = g$.

We say p is a fibration if for every $Y \in \mathbb{E}$ and $u : I \rightarrow p(Y) \in \mathbb{B}$, there is a cartesian morphism over u .

The domain of a fibration is known as the total space and the codomain as the base space. The subcategory of $\mathbb{E}$ consisting of the objects sent to I and the maps sent to id_I is known as the fibre of I , written $\mathbb{E}_I$.

¹⁵ Jacobs, *Categorical Logic and Type Theory*.

¹⁶ Streicher, *Fibered Categories à la Jean Bénabou*.

¹⁷ Jacobs, *Categorical Logic and Type Theory*, Definition 1.1.3.

We will only really be interested in fibrations that come with a choice of fibration. We refer to such fibrations as *cloven*, and will denote the chosen cartesian morphism for X (above J) and $u : I \rightarrow J$ by $\text{cart}(u, Y) : \text{cart}(u, I) \rightarrow Y$.

The most important examples of fibrations are families of objects and the codomain functor.

Example A.4.2 (Family Fibration). ¹⁸ Let $\mathcal{C}$ be a category. The category of set-indexed families $\text{Fam}(\mathcal{C})$ is defined as follows:

¹⁸ van Oosten, *Realizability: An Introduction to its Categorical Side*, Definition 1.2.1.

- The objects are set-indexed families $(X_i \in \mathcal{C})_{i \in I}$;
- Morphisms $(X_i \in \mathcal{C})_{i \in I} \rightarrow (Y_j \in \mathcal{C})_{j \in J}$ consist of a function $u : I \rightarrow J$ and a family of arrows $(f_i : X_i \rightarrow Y_{u(i)})_{i \in I}$;
- The composition of compatible morphisms

$$(X_i)_{i \in I} \xrightarrow{(u, (f_i)_{i \in I})} (Y_j)_{j \in J} \xrightarrow{(v, (g_j)_{j \in J})} (Z_k)_{k \in K}$$

is given by $(v \circ u, (g_{u(i)} \circ f_i)_{i \in I})$.

The family fibration is the functor sending families to their indexing sets. If we have a family $(Y_j \in \mathcal{C})_{j \in J}$ and a function $u : I \rightarrow J$, then there is a cartesian morphism $\text{cart}(u, (Y_j \in \mathcal{C})_{j \in J}) := (u, (id_{Y_{u(i)}})_{i \in I}) : (Y_{u(i)})_{i \in I} \rightarrow (Y_j)_{j \in J}$.

Example A.4.3 (Codomain Fibration). ¹⁹ Let $\mathcal{C}$ be a category with pullbacks. The functor $\text{cod} : \mathcal{C}^{\rightarrow} \rightarrow \mathcal{C}$ sending arrows $f : X \rightarrow I$ to their codomains I is a fibration. Given an object $g : Y \rightarrow J \in \mathcal{C}^{\rightarrow}$ and an arrow $u : I \rightarrow J \in \mathcal{C}$, then $\text{cart}(u, g)$ is the pullback square below.

¹⁹ *ibid.*, Proposition 1.1.6.

$$\begin{array}{ccc} u^*Y & \xrightarrow{g^*u} & Y \\ u^*g \downarrow & & \downarrow g \\ I & \xrightarrow{u} & J \end{array}$$

Example A.4.4. Let $\mathcal{C}$ be a category. Then there is a fibration $\pi : \mathcal{C} \times \mathcal{C}^{\text{op}} \rightarrow \mathcal{C}$ that projects onto the first component. Given $u : I \rightarrow J \in \mathcal{C}$ and $(J, Y) \in \mathcal{C}^{\text{op}}$ over J , there is a cartesian morphism $(u, id_Y) : (I, Y) \rightarrow (J, Y)$ over u . More generally, a morphism (u, f) is cartesian iff f is an iso.

It is possible to factor arrows in the total space into arrows that “don’t move” in the base space, followed by a cartesian morphism which does.

Proposition A.4.5 (Vertical–Cartesian Factorisation System). ²⁰ Suppose we have a fibration $p : \mathbb{E} \rightarrow \mathbb{B}$. If $f : X \rightarrow Y$ is above an identity morphism id_I , we call f a vertical morphism²¹.

²⁰ *ibid.*, Exercises 1.1.3, 1.1.4.

If V is the class of all vertical morphisms and C is the class of all cartesian morphisms, then $\mathbb{E}$ has a (V, C) -factorisation system.

²¹ The upshot of this proposition is that we might call cartesian morphisms *horizontal morphisms*.

In fibrations, there is an analogue of taking pullbacks that allows us to transfer morphisms between fibres.

Definition A.4.6 (Reindexing functor). ²² Let $p : \mathbb{E} \rightarrow \mathbb{B}$ be a cloven fibration and $u : I \rightarrow J$ a morphism in $\mathbb{B}$. The reindexing functor²³ $u^* : \mathbb{E}_J \rightarrow \mathbb{E}_I$ sends $f : X \rightarrow Y \in \mathbb{E}_J$ to the unique²⁴ morphism u^*f making the diagram below commute.

$$\begin{array}{ccc} \text{cart}(u, I) & \xrightarrow{\text{cart}(u, X)} & X \\ u^*f \downarrow & & \downarrow f \\ \text{cart}(u, I) & \xrightarrow{\text{cart}(u, Y)} & Y \end{array}$$

As always, it's not just fibrations we are interested in, but also the maps between them.

Definition A.4.7 (Morphism of Fibrations, Fibred Functor). ²⁵ Suppose that $p : \mathbb{E} \rightarrow \mathbb{B}$ and $q : \mathbb{D} \rightarrow \mathbb{A}$ are fibrations. A morphism of fibrations $p \rightarrow q$ consists of a pair of functors $K : \mathbb{B} \rightarrow \mathbb{A}$ and $H : \mathbb{E} \rightarrow \mathbb{D}$ such that $q \circ H = K \circ p$ and H sends cartesian morphisms to cartesian morphisms. Such a functor H is called a fibred functor.

With this definition of morphism, we can form a category of fibrations called Fib , moreover, Fib is a 2-category. If our fibrations are over a common base category $\mathbb{B}$, then we will have a 2-category $\text{Fib}(\mathbb{B})$, where the functor on base categories is always $\text{id}_{\mathbb{B}}$. We will give a proof of these facts after defining the 2-cells.

Definition A.4.8 (2-Cell, Fibred Natural Transformation). ²⁶ Fix fibrations $p : \mathbb{E} \rightarrow \mathbb{B}$ and $q : \mathbb{D} \rightarrow \mathbb{A}$, and let (K, H) and (L, G) be morphisms of fibrations $p \rightarrow q$. A 2-cell $(K, H) \Rightarrow (L, G)$ consists of a pair of natural transformations $\sigma : K \Rightarrow L$ and $\tau : H \Rightarrow G$ with τ above²⁷ σ , such that the following diagram commutes:

$$\begin{array}{ccc} \mathbb{E} & \begin{array}{c} \xrightarrow{H} \\ \tau \Downarrow \\ \xrightarrow{G} \end{array} & \mathbb{D} \\ p \downarrow & & \downarrow q \\ \mathbb{B} & \begin{array}{c} \xrightarrow{K} \\ \sigma \Downarrow \\ \xrightarrow{L} \end{array} & \mathbb{A} \end{array}$$

If H, G are the identity functor and σ is the identity natural transformation, we called such a 2-cell a fibred natural transformation.

Theorem A.4.9. ²⁸ Both Fib and $\text{Fib}(\mathbb{B})$ are 2-categories.

Proof. We will only give the proof for Fib . Firstly, morphisms of fibrations $(H, K) : p \rightarrow q$ and $(G, L) : p \rightarrow q$ obviously compose as $(G \circ H, K \circ L)$ with $\text{id}_p = (\text{id}_{\mathbb{E}}, \text{id}_{\mathbb{B}})$ the identity morphism for a fibration $p : \mathbb{E} \rightarrow \mathbb{B}$. Since both H and G preserve cartesian morphisms, their composition does.

Similarly, it is clear that we can define both vertical and horizontal composition by performing vertical and horizontal composition on both components of the 2-cell. $\square$

²² Jacobs, *Categorical Logic and Type Theory*, §1.4.

²³ We will reuse the notation for pullbacks.

²⁴ It exists, and is unique, because $\text{cart}(u, X)$ and $\text{cart}(u, Y)$ are cartesian.

²⁵ *ibid.*, Definition 1.7.1.

²⁶ *ibid.*, Definition 1.7.7.

²⁷ Every τ_X is above σ_X .

²⁸ *ibid.*, Definition 1.7.1, 1.7.3.

Definition A.4.10 (Fibred Adjunction). ²⁹ A fibred adjunction between fibrations over $\mathbb{B}$ is an internal adjunction in $\text{Fib}(\mathbb{B})$. More explicitly, a fibred adjunction between fibrations $p : \mathbb{E} \rightarrow \mathbb{B}$ and $p : \mathbb{D} \rightarrow \mathbb{B}$ consists of two morphisms of fibrations $F : p \rightarrow q$, $G : q \rightarrow p$ and two natural transformations $\eta : id_{\mathbb{E}} \Rightarrow G \circ F$, $\eta : F \circ G \Rightarrow id_{\mathbb{D}}$, such that the triangular laws $G\epsilon \circ \eta G = id$, $\epsilon F \circ F\eta = id$ hold.

²⁹ *ibid.*, Definition 1.8.6.

An important construction on fibrations is the fibrewise opposite.

Definition A.4.11 (Opposite of a Fibration). ^{30,31} Let $p : \mathbb{E} \rightarrow \mathbb{B}$ be a fibration. Define a new fibration $p^{\text{op}} : \mathbb{E}^{(\text{op})} \rightarrow \mathbb{B}$ as follows:

³⁰ *ibid.*, Definition 1.10.11.

³¹ Streicher, *Fibred Categories à la Jean Bénabou*, §5.

- Objects of $\mathbb{E}^{(\text{op})}$ are exactly the objects of $\mathbb{E}$.
- Morphisms $X \rightarrow Y$ of $\mathbb{E}^{(\text{op})}$ are spans $X \xleftarrow{f} Z \xrightarrow{g} Y$ such that f is vertical and g is cartesian, modulo the equivalence relation

$$(X \xleftarrow{f} Z \xrightarrow{g} Y) \sim (X \xleftarrow{f'} Z' \xrightarrow{g'} Y) \quad \text{iff} \quad \begin{array}{ccccc} & & Z & & \\ & f & \swarrow & g & \\ X & & & & Y \\ & f' & \swarrow & g' & \\ & & Z' & & \end{array}$$

for some (necessarily unique) vertical isomorphism $h : Z \cong Z'$.

- Identities are given by $id_X = [X \xleftarrow{id} X \xrightarrow{id} X]_{\sim}$.
- Composition is defined by taking (equivalence classes) of reindexings of representatives. Suppose we have morphisms $[X \xleftarrow{\alpha} A \xrightarrow{\beta} Y]_{\sim} : X \rightarrow Y$ and $[Y \xleftarrow{\gamma} B \xrightarrow{\delta} Z]_{\sim} : Y \rightarrow Z$, then the composite is given by equivalence classes of the following span between X and Z .

$$\begin{array}{ccccc} & & P & & \\ & \beta^* \gamma & \swarrow & \gamma^* \beta & \\ & A & & B & \\ \alpha \swarrow & & & & \searrow \delta \\ X & & Y & & Z \end{array} \quad (\text{spans compose})$$

- The functor $p^{\text{op}} : \mathbb{E}^{(\text{op})} \rightarrow \mathbb{B}$ is defined by:

$$\begin{aligned} X &\mapsto p(X) \\ [X \xleftarrow{f} Z \xrightarrow{g} Y] &\mapsto p(g) \end{aligned}$$

We call this fibration p^{op} the opposite (fibration) of p .

That this construction is well-defined relies on a few observations. First, the requirement that the morphism h which makes two spans equivalent is a vertical isomorphism ensures that the respective cartesian morphisms are over the same map in the base. Secondly, the maps $\beta^* \gamma$ and $\gamma^* \beta$ come from the definition of the change-of-base functor, which is why they are vertical and cartesian respectively.

The following lemma justifies the name “opposite” for the construction.

Lemma A.4.12. ³² Let $p : \mathbb{E} \rightarrow \mathbb{B}$ be a fibration. The following are true:

³² van Oosten, *Realizability: An Introduction to its Categorical Side*, Lemma 1.10.12.

1. The functor $p^{\text{op}} : \mathbb{E}^{(\text{op})} \rightarrow \mathbb{B}$ is a fibration and morphisms $[f_1, f_2] : \mathbb{E}^{\text{op}}$ are cartesian iff the vertical map f_2 is an iso.
2. p^{op} is the fibrewise opposite of p , i.e., for each $I : \mathbb{B}$ there is a natural isomorphism of fibre categories $(E^{(\text{op})})_I \cong (E_I)^{(\text{op})}$.
3. There is an isomorphism of fibrations $(p^{\text{op}})^{\text{op}} \cong p$.

Proof.

¹ We've already seen that the map p^{op} is well-defined, and it evidently preserves identities. We can see from the definition of composition in Equation (spans compose) that, as both γ and $\beta^* \gamma$ are vertical, both β and $\gamma^* \beta$ are above the same morphism in the base. It follows that p^{op} preserves composition, since p does.

In the case that $[f_1, f_2]$ is vertical, then f_2 is both cartesian and vertical, and hence an isomorphism. Thus, any such morphism $[f_1, f_2]$ can be equivalently represented by $[f_1 \circ f_2^{-1}, \text{id}]$.

It is easy to see that if $f_2 : X \rightarrow Y$ is a cartesian morphism of the fibration p , then $[\text{id}_X, f_2]$ is a cartesian morphism of the p^{op} fibration. Then $[f_1, f_2]$ is cartesian iff $[f_1, f_2] \circ -$ is an isomorphism iff f_1 is an isomorphism, using that $[\text{id}, f_2]$ is cartesian and that $[f_1, \text{id}]$ is the unique vertical morphism such that $[\text{id}, f_2] \circ [f_1, \text{id}] = [f_1, f_2]$.

² Taking the identity on objects and mapping $[f, \text{id}] \mapsto f$ gives a well-defined isomorphism $(E^{(\text{op})})_I \cong (E_I)^{(\text{op})}$.

³ All morphisms $f : X \rightarrow Y$ in the total space of a fibration p have a factorisation $f = f^c \circ f^v$ where $f^c : Z \rightarrow Y$ is cartesian and $f^v : X \rightarrow Z$ is vertical. We can define a functor $F : p \rightarrow (p^{\text{op}})^{\text{op}}$ that is the identity on objects and acts on morphisms by

$$(f : X \rightarrow Y) \mapsto [[f^v, \text{id}_X] : Z \rightarrow X, [\text{id}_Z, f^c] : Z \rightarrow Y].$$

This functor is well-defined, because for any other vertical-cartesian factorisation $f = f^{c'} \circ f^{v'}$ there exists a unique isomorphism h such that $h \circ f^v = f^{v'}$ and $f^{c'} \circ h = f^c$; it follows that $[f^v, \text{id}_X] = [f^{v'}, \text{id}_X]$ and similarly for f^c . We also have that $(p^{\text{op}})^{\text{op}}(F(f)) = p^{\text{op}}(f^c) = p(f^c) = p(f)$, and F evidently preserves cartesian morphisms.

Similarly, we can define a functor $G : (p^{\text{op}})^{\text{op}} \rightarrow p$ that is the identity on objects and acts on morphisms by

$$[[f^v, i^v], [i^c, f^c]] \mapsto f^c \circ (i^c)^{-1} \circ f^v \circ (i^v)^{-1}.$$

It is easy to check that G is well-defined, preserves cartesian morphisms, and is inverse to F . $\square$

The final observation we make is that taking opposites of fibrations is a 2-functor, but we omit the proof.

B Free Monoid on $\text{Cont}(\mathcal{C})$

IN THIS SECTION, we show that $P^\diamond$ is a monoid in the monoidal category $(\text{Cont}(\mathcal{C}), \star, \mathcal{I})$, and prove that it is the free one in the sense of [Theorem B.1](#). When passing to the category of polynomial functors on $\mathcal{C}$, this is precisely the free monad on $\langle\langle P \rangle\rangle$ from Gambino & Kock ¹, except that here it is constructed directly on *polynomials (containers)* rather than polynomial functors.

¹ Gambino and Kock, ‘Polynomial functors and polynomial monads’, §4.

Theorem B.1 (Free Monoid on P). *A monoid on P is a monoid $(P^\square, \mu, \eta)$ in the monoidal category $(\text{Cont}(\mathcal{C}), \star, \mathcal{I})$, equipped with a map $u : P \rightarrow P^\square$.*

There exists a monoid $(P^\diamond, \mu_P, \eta_P)$ with map $u_P : P \rightarrow P^\diamond$, such that, for any other monoid $(P^\square, \mu', \eta')$ and map $u' : P \rightarrow P^\square$, there exists a unique monoid morphism $\theta : P^\diamond \rightarrow P^\square$ satisfying $u' = \theta \circ u_P$.

Before giving this proof, we will need to fix some notation and give a few lemmas, all of which have straightforward proofs. Throughout, we assume that $\mathcal{C}$ is IWM -category and will make heavy use of [Definition 5.3.9](#) and [Theorem 5.3.11](#). We will also fix a particular container P .

Notation B.2. Use $(\text{lfp}_A, \text{in}_A)$ for the initial algebra of $F_A(X) = A + X \star P$.

Lemma B.3. *If $f : A \rightarrow B$ then $\text{in}_B \circ (f + \text{id})$ is an F_A -algebra.*

Lemma B.4. $(P^\diamond, [\text{id}, \text{in}_\mathcal{I} \circ \iota_2])$ is a $F_{P^\diamond}$ -algebra.

Definition B.5 (η_P, μ_P) . *Let $P \in \text{Cont}(\mathcal{C})$ be a container. Then we define:*

- $\eta_P = \text{in}_\mathcal{I} \circ \iota_1$, and
- μ_P to be the unique $F_{P^\diamond}$ -algebra morphism from $\text{in}_{P^\diamond}$ to $(P^\diamond, [\text{id}, \text{in}_\mathcal{I} \circ \iota_2])$.

Lemma B.6. *The map $A \rightarrow F_A$ is a functor $\mathcal{C} \rightarrow \mathcal{C}^\mathcal{C}$, sending f to $f + \text{id}$.*

Corollary B.7. *The map $A \mapsto \text{lfp}_A$ is a functor $\mathcal{C} \rightarrow \mathcal{C}$ such that, for any $f : A \rightarrow B$, the following diagram commutes.*

$$\begin{array}{ccc} F_A(\text{lfp}_A) & \xrightarrow{\quad \text{in} \quad} & \text{lfp}_A \\ F_A(\text{lfp}_f) \downarrow & & \downarrow \text{lfp}_f \\ F_A(\text{lfp}_B) & \xrightarrow{f + \text{id}} F_B(\text{lfp}_B) \xrightarrow{\quad \text{in} \quad} & \text{lfp}_B \end{array}$$

Moreover, $\text{lfp}_f = f \star \text{id}$ and (in_A) is a natural isomorphism $\text{id} \rightarrow \text{lfp}$.

We now have enough to show we have a monoid in $(\text{Cont}(\mathcal{C}), \star, \mathcal{I})$.

Proposition B.8. $(P^\diamond, \mu_P, \eta_P)$ is a monoid in $(\text{Cont}(\mathcal{C}), \star, \mathcal{I})$.

Proof. To prove the monoidal axioms hold, we will in each case use the strategy of showing that both arrows in an identity are unique algebra morphisms from an appropriate initial algebra to a chosen target algebra.

Pentagon Identity To show the identity $\mu_P \circ (\mu_P \circ \text{id}) = \mu_P \circ (\text{id} \circ \mu_P) \circ \alpha$ holds, we will prove that both maps are $F_{P^\diamond \star P^\diamond}$ -algebra morphisms from $\text{in}_{P^\diamond \star P^\diamond}$ to $[\text{id}, \text{in}_I \circ \iota_2] \circ (\mu_P + \text{id})$.

For the left hand side, we combine the diagram from [Corollary B.7](#) with the definition of μ_P to obtain the following diagram.

$$\begin{array}{ccccc}
 F_{P^\diamond \star P^\diamond}(\text{lf}_{P^\diamond \star P^\diamond}) & \xrightarrow{\text{in}} & \text{lf}_{P^\diamond \star P^\diamond} & & \\
 \downarrow F_{P^\diamond \star P^\diamond}(\mu_P \star \text{id}) & & \downarrow \mu_P \star \text{id} & & \\
 F_{P^\diamond \star P^\diamond}(\text{lf}_{P^\diamond}) & \xrightarrow{\mu_P + \text{id}} & F_{P^\diamond}(\text{lf}_{P^\diamond}) & \xrightarrow{\text{in}} & \text{lf}_{P^\diamond} \\
 \downarrow F_{P^\diamond \star P^\diamond}(\mu_P) & & \downarrow F_{P^\diamond}(\mu_P) & & \downarrow \mu_P \\
 F_{P^\diamond \star P^\diamond}(\text{lf}_I) & \xrightarrow{\mu_P + \text{id}} & F_{P^\diamond}(\text{lf}_I) & \xrightarrow{[\text{id}, \text{in}_I \circ \iota_2]} & \text{lf}_I
 \end{array}$$

For the right side, we must recall from [Theorem 5.3.11](#) that the initial algebra in_Q for F_Q was given as the composite of a natural isomorphism $Q + Q \star P^\diamond \star P \cong Q \star (I + P^\diamond \star P)$ (which we will call **distr**) and $\text{id} \star \text{in}_I$. The following diagram proving that $\mu_P \circ (\text{id} \circ \mu_P) \circ \alpha$ is an algebra morphism can be constructed; all squares and triangles hold by functoriality, naturality, or the definitions of in_Q or μ_P .

$$\begin{array}{ccccc}
 F_{P^\diamond \star P^\diamond}(\text{lf}_{P^\diamond \star P^\diamond}) & \xrightarrow{\text{in}} & \text{lf}_{P^\diamond \star P^\diamond} & & \\
 \downarrow F_{P^\diamond \star P^\diamond}(\alpha) & \searrow \text{distr} & \downarrow \text{id} \star \text{in} & & \downarrow \alpha \\
 & P^\diamond \star F_{P^\diamond}(\text{lf}_{P^\diamond}) & & & \\
 & \downarrow \text{id} \star F_{P^\diamond}(\mu_P) & & & \\
 F_{P^\diamond \star P^\diamond}(\text{id} \star \mu_P) & \searrow \text{distr} & P^\diamond \star F_{P^\diamond}(\text{lf}_I) & \xrightarrow{\text{id} \star [\text{id}, \text{in}_I \circ \iota_2]} & \downarrow \text{id} \star \mu_P \\
 & & \downarrow [\text{id}, \text{id} \star (\text{in} \circ \iota_2)] & & \\
 F_{P^\diamond \star P^\diamond}(\text{lf}_{P^\diamond}) & \xrightarrow{\mu_P + \text{id}} & F_{P^\diamond}(\text{lf}_{P^\diamond}) & \xrightarrow{\text{in}} & \text{lf}_{P^\diamond} \\
 \downarrow F_{P^\diamond \star P^\diamond}(\mu_P) & & \downarrow F_{P^\diamond}(\mu_P) & & \downarrow \mu_P \\
 F_{P^\diamond \star P^\diamond}(\text{lf}_I) & \xrightarrow{\mu_P + \text{id}} & F_{P^\diamond}(\text{lf}_I) & \xrightarrow{[\text{id}, \text{in}_I \circ \iota_2]} & \text{lf}_I
 \end{array}$$

Triangle Identities To prove the triangle identities, it will be enough to show that both $\mu_P \circ (\eta_P \star I) \circ \lambda^{-1}$ and $\mu_P \circ (I \star \eta_P) \circ \rho^{-1}$ are F_I -algebra morphisms from in_I to in_I , since id is the unique such algebra morphism. Again, [Corollary B.7](#) is useful for establishing the first identity, and the second relies crucially on in being a natural transformation.

$$\begin{array}{ccc}
F_{\mathcal{I}}(P^{\diamond}) & \xrightarrow{\text{in}} & P^{\diamond} \\
F_{\mathcal{I}}(\lambda^{-1}) \downarrow & & \downarrow \lambda^{-1} \\
F_{\mathcal{I}}(\mathcal{I} \star P^{\diamond}) & \xrightarrow{\text{in}} & \mathcal{I} \star P^{\diamond} \\
F_{\mathcal{I}}(\eta_P \star \text{id}) \downarrow & & \downarrow \eta_P \star \text{id} \\
F_{\mathcal{I}}(\text{lfp}_{P^{\diamond}}) & \xrightarrow{\eta_P + \text{id}} F_{P^{\diamond}}(\text{lfp}_{P^{\diamond}}) \xrightarrow{\text{in}} \text{lfp}_{P^{\diamond}} & \\
F_{\mathcal{I}}(\mu_P) \downarrow & \downarrow F_{P^{\diamond}}(\mu_P) & \downarrow \mu_P \\
F_{\mathcal{I}}(P^{\diamond}) & \xrightarrow{\eta_P + \text{id}} F_{P^{\diamond}}(P^{\diamond}) \xrightarrow{[\text{id}, \text{in}_{\mathcal{I}} \circ \iota_2]} P^{\diamond} & \\
& \searrow \text{in} & \\
& & P^{\diamond}
\end{array}$$

□

Corollary B.9. $\llbracket P^{\diamond} \rrbracket$ is a monad on $\text{Poly}(\mathcal{C})$.

Proof. This follows because the $\text{Poly}(\mathcal{C})$ is equivalent to $\text{Cont}(\mathcal{C})$ and monads are monoids in the category of endofunctors. □

We require one last definition and lemma before proving the main theorem of this section.

Definition B.10 (u_P). We define a morphism $u_P : P \rightarrow P^{\diamond}$ by

$$u_P := \text{in}_I \circ \iota_2 \circ (\eta_P \star \text{id}) \circ \lambda^{-1}.$$

Lemma B.11. $\mu_P \circ (\text{id} \star u_P) = \text{in}_{\mathcal{I}} \circ \iota_2$

Proof. The following diagram commutes.

$$\begin{array}{ccc}
P^{\diamond} \star P & \xleftarrow{\iota_2} & P^{\diamond} + (P^{\diamond} \star P) \\
\text{id} \star \lambda^{-1} \downarrow & & \downarrow \text{id} + \rho^{-1} \star \text{id} \\
P^{\diamond} \star (\mathcal{I} \star P) & \xrightarrow{\iota_2 \circ \alpha} & P^{\diamond} + (P^{\diamond} \star \mathcal{I}) \star P \\
\text{id} \star (\eta_P \star \text{id}) \downarrow & & \downarrow \text{id} + (\text{id} \star \eta) \star \text{id} \\
P^{\diamond} \star (P^{\diamond} \star P) & \searrow \iota_2 \circ \alpha & \\
\text{id} \star \iota_2 \downarrow & & \downarrow \text{id} + (\text{id} \star \eta) \star \text{id} \\
P^{\diamond} \star (\mathcal{I} + P^{\diamond} \star P) & \xrightarrow{\text{distr}} & P^{\diamond} + (P^{\diamond} \star P^{\diamond}) \star P \\
\text{id} \star \text{in}_{\mathcal{I}} \downarrow & & \downarrow \text{id} + \mu_P \star \text{id} \\
P^{\diamond} \star P^{\diamond} & \xleftarrow{\text{in}_{P^{\diamond}}} & P^{\diamond} + (P^{\diamond} \star P^{\diamond}) \star P \\
\mu_P \downarrow & & \downarrow \text{id} + \mu_P \star \text{id} \\
P^{\diamond} & \xleftarrow{[\text{id}, \text{in}_I \circ \iota_2]} & P^{\diamond} + P^{\diamond} \star P
\end{array}$$

□

Proof of Theorem B.1. We have seen in Proposition B.8 that $(P^{\diamond}, \mu_P, \eta_P)$ is a monoid, and we take u_P as defined above. Let $(P^{\square}, \mu', \eta')$ be another monoid with map $u' : P \rightarrow P^{\square}$. The map θ is the unique $(\mathcal{I} + - \star P)$ -algebra

morphism from $\text{in}_{(\mathcal{I} + - \star P)}$ to $[\eta', \mu' \circ (\text{id} \star u')]: \mathcal{I} + P^\diamond \star P \rightarrow P^\square$, i.e., it uniquely satisfies the diagram:

$$\begin{array}{ccc} \mathcal{I} + P^\diamond \star P & \xrightarrow{\text{in}} & P^\diamond \\ \text{id} + \theta \star \text{id} \downarrow & & \downarrow \theta \\ \mathcal{I} + P^\square \star P & \xrightarrow{[\eta', \mu' \circ (\text{id} \star u')]} & P^\square \end{array}$$

θ preserves monoidal units The law $\eta' = \theta \circ \eta_P$ follows quickly from the definition of θ .

$$\begin{aligned} \theta \circ \eta_P &= \theta \circ \text{in}_I \circ \iota_I \\ &= [\eta', \mu' \star (\text{id} \circ u')] \circ (\text{id} + \theta \star \text{id}) \circ \iota_1 \\ &= [\eta', \mu' \star (\text{id} \circ u')] \circ \iota_1 \\ &= \eta' \end{aligned}$$

$u' = \theta \circ u_P$ holds Again, this can be established by a simple calculation.

$$\begin{aligned} \theta \circ u_P &= \theta \circ \text{in} \circ \iota_2 \circ (\eta_P \star \text{id}) \circ \lambda^{-1} \\ &= [\eta', \mu' \circ (\text{id} \star u')] \circ (\text{id} + \theta \star \text{id}) \circ \iota_2 \circ (\eta_P \star \text{id}) \circ \lambda^{-1} \\ &= \mu' \circ (\text{id} \star u') \circ (\theta \star \text{id}) \circ (\eta_P \star \text{id}) \circ \lambda^{-1} \\ &= \mu' \circ ((\theta \star \eta_P) \star \text{id}) \circ (\text{id} \star u') \circ \lambda^{-1} \\ &= \mu' \circ (\eta' \star \text{id}) \circ (\text{id} \star u') \circ \lambda^{-1} \\ &= \lambda \circ (\text{id} \star u') \circ \lambda^{-1} \\ &= u' \end{aligned}$$

θ is a monoid morphism We have already proven $\eta' = \theta \circ \eta_P$, so it remains to check that $\theta \circ \mu_P = \mu' \circ (\theta \star \theta)$. We will establish the equality by checking that both maps are $(P^\diamond + - \star P)$ -algebra morphisms from the initial algebra to $[\theta, \mu' \circ (\text{id} \star u')]$. In the case of $\theta \circ \mu_P$, this is straightforward from the definitions of θ and μ_P .

$$\begin{array}{ccc} P^\diamond + (P^\diamond \star P^\diamond) \star P & \xrightarrow{\text{in}} & P^\diamond \star P^\diamond \\ \text{id} + \mu_P \star \text{id} \downarrow & & \downarrow \mu_P \\ P^\diamond + P^\diamond \star P & \xrightarrow{[\text{id}, \text{in} \circ \iota_2]} & P^\diamond \\ \text{id} + \theta \star \text{id} \downarrow & & \downarrow \theta \\ P^\diamond + P^\square \star P & \xrightarrow{[\theta, \mu' \circ (\text{id} \star u')]} & P^\square \end{array}$$

For $\mu' \circ (\theta \star \theta)$, it helps to treat both parts of the coproduct separately. In the case of $\mu' \circ (\theta \star \theta) \circ \text{in} \circ \iota_1 = \theta$ we have the following.

$$\begin{array}{ccccc}
P^\diamond & \xleftarrow{\iota_1} & P^\diamond + (P^\diamond \star P^\diamond) \star P & \xrightarrow{\text{in}} & \\
\downarrow \rho^{-1} & & \downarrow \text{distr} & & \downarrow \\
P^\diamond \star \mathcal{I} & \xrightarrow{\text{id} \star \iota_1} & P^\diamond \star (\mathcal{I} + P^\diamond \star P) & \xrightarrow{\text{id} \star \text{in}} & P^\diamond \star P^\diamond \\
\downarrow \theta \star \text{id} & \searrow \text{id} \star \eta_P & & \searrow \theta \star \theta & \downarrow \\
P^\square \star \mathcal{I} & \xrightarrow{\text{id} \star \eta'} & & & P^\square \star P^\square \\
& \searrow \rho & & \swarrow \mu' & \\
& & P^\square & &
\end{array}$$

For $\mu' \circ (\theta \star \theta) \circ \text{in} \circ \iota_2 = \mu' \circ ((\mu_P \circ (\theta \star \theta)) \star u')$ we have

$$\begin{array}{ccccc}
(P^\diamond \star P^\diamond) \star P & \xleftarrow{\iota_2} & P^\diamond + (P^\diamond \star P^\diamond) \star P & & \\
\downarrow \alpha & & \downarrow \text{in} & & \\
P^\diamond \star (P^\diamond \star P) & \xrightarrow{\text{id} \star (\text{in} \star \iota_2)} & P^\diamond \star P^\diamond & & \\
\downarrow \theta \star (\theta \star \text{id}) & \xrightarrow{(\dagger)} & \downarrow \theta \star \theta & & \\
P^\square \star (P^\square \star P) & \xrightarrow{\text{id} \star (\text{id} \star u')} & P^\square \star (P^\square \star P^\square) & \xrightarrow{\text{id} \star \mu'} & P^\square \star P^\square \\
\downarrow \alpha^{-1} & & \downarrow \alpha^{-1} & & \downarrow \mu' \\
(P^\square \star P^\square) \star P & \xrightarrow{\text{id} \star u'} & (P^\square \star P^\square) \star P^\square & & \\
\downarrow \mu' \star \text{id} & & \downarrow \mu' \star \text{id} & & \\
P^\square \star P & \xrightarrow{\text{id} \star u'} & P^\square \star P^\square & \xrightarrow{\mu'} & P
\end{array}$$

where $(\dagger)$ follows from the functoriality of $\star$ and the definition of θ and the top square comes from [Theorem 5.3.11](#).

Then by the universal property of coproducts, $\mu' \circ (\theta \star \theta) \circ \text{in} = [\theta, \mu' \circ (\text{id} \star u')] \circ ((\mu' \circ (\theta \star \theta)) \star \text{id})$, as required.

θ is unique Suppose we have a monoid morphism $\theta' : P^\diamond \rightarrow P^\square$ satisfying $u' = \theta' \circ u$, then we want to check that it is a $(\mathcal{I} + - \star P)$ -algebra morphism from the initial algebra to $[\eta', \mu \circ (\text{id} \star u')]$. By a short calculation using [Lemma B.11](#), we have

$$\begin{aligned}
\theta' \circ \text{in} \circ \iota_2 &= \theta' \circ \mu \circ (\text{id} \star u) \\
&= \mu' \circ (\theta' \star \theta') \circ (\text{id} \star u) \\
&= \mu' \circ (\theta' \star (\theta' \circ u)) \\
&= \mu' \circ (\theta' \star u') \\
&= \mu' \circ (\text{id} \star u') \circ (\theta' \star \text{id}).
\end{aligned}$$

Taken together with $\eta' = \theta' \circ \eta = \theta' \circ \text{in} \circ \iota_2$, we obtain $\theta' \circ \text{in} = [\eta', \mu \circ (\text{id} \star u')] \circ (\text{id} + (\theta' \star \text{id}))$, as required. Since there is a unique $(\mathcal{I} + - \star P)$ -algebra morphism from $\text{in}_{(\mathcal{I} + - \star P)}$ to $[\eta', \mu \circ (\text{id} \star u')]$, we have $\theta' = \theta$. $\square$

As a corollary, we can obtain the free monad on a polynomial, which will be a polynomial functor by construction.

Corollary B.12. ² $\llbracket P^\diamond \rrbracket$ is the free monad on $\llbracket P \rrbracket$ in $\text{Poly}(\mathcal{C})$.

² Gambino and Kock, ‘Polynomial functors and polynomial monads’, §4.2 – Theorem 4.5.

The final thing in this section is to check that, as is well known, the free monoid construction is itself a monad on containers.

Proposition B.13. The construction $P \mapsto P^\diamond$ is a monad on $\text{Cont}(\mathcal{C})$.

Proof. Recall that $(P^\diamond)^\diamond$ is the initial algebra for the functor $F_{P^\diamond}$, and note that $(P^\diamond, [\eta_P, \mu_P])$ is also an algebra for $F_{P^\diamond}$, and so there is a unique algebra morphism $m_P : (P^\diamond)^\diamond \rightarrow P^\diamond$ between them. We will take the collection $(m_P)_{P \in \text{Cont}(\mathcal{C})}$ to be the monad multiplication for $(-)^\diamond$. For the monad unit, we will take the collection $(u_P)_{P \in \text{Cont}(\mathcal{C})}$, where u_P is as in [Definition B.10](#). In both cases, it is easy to check we have natural transformations.

It remains to check the coherence conditions.

Associativity To establish $m_P \circ (m_P)^\diamond = m_P \circ m_{P^\diamond}$, we will prove that both are $(\mathcal{I} + - \star P^{\diamond\diamond})$ -algebra morphisms from the initial algebra in to $[\eta_P, \mu_P] \circ (\text{id} + \text{id} \star m_P)$.

On the left side, we have the following diagram, where the top square holds by the definition of the functor $(-)^\diamond$, and the bottom right square by definition of m_P .

$$\begin{array}{ccccc}
 \mathcal{I} + P^{\diamond\diamond} \star P^{\diamond\diamond} & \xrightarrow{\quad \text{in} \quad} & P^{\diamond\diamond\diamond} & & \\
 \text{id} + (m_P)^\diamond \star \text{id} \downarrow & & \downarrow (m_P)^\diamond & & \\
 \mathcal{I} + P^{\diamond\diamond} \star P^{\diamond\diamond} & \xrightarrow{\text{id} + \text{id} \star m_P} \mathcal{I} + P^{\diamond\diamond} \star P^\diamond & \xrightarrow{\quad \text{in} \quad} & P^{\diamond\diamond} & \\
 \text{id} + m_P \star \text{id} \downarrow & \text{id} + \text{id} \star m_P \downarrow & & \downarrow m_P & \\
 \mathcal{I} + P^\diamond + P^{\diamond\diamond} & \xrightarrow{\text{id} + \text{id} \star m_P} \mathcal{I} + P^\diamond + P^\diamond & \xrightarrow{[\eta_P, \mu_P]} & P^\diamond &
 \end{array}$$

For the right hand side, we have the following diagram where the top square commutes by definition, and the bottom is the cotupling of the equations $m_{P^\diamond} \circ \eta_{P^\diamond} = \eta_P$ and $m_P \circ \mu_{P^\diamond} = \mu_P \circ (m_P \star m_P)$, which hold by the definition of m and naturality of μ respectively.

$$\begin{array}{ccc}
 \mathcal{I} + P^{\diamond\diamond} \star P^{\diamond\diamond} & \xrightarrow{\quad \text{in} \quad} & P^{\diamond\diamond\diamond} \\
 \text{id} + m_{P^\diamond} \star \text{id} \downarrow & & \downarrow m_{P^\diamond} \\
 \mathcal{I} + P^{\diamond\diamond} \star P^{\diamond\diamond} & \xrightarrow{[\eta_{P^\diamond}, \mu_{P^\diamond}]} & P^{\diamond\diamond} \\
 \text{id} + m_P \star \text{id} \downarrow & & \downarrow m_P \\
 \mathcal{I} + P^\diamond \star P^{\diamond\diamond} & \xrightarrow{\text{id} + \text{id} \star m_P} \mathcal{I} + P^\diamond \star P^\diamond & \xrightarrow{[\eta_P, \mu_P]} P^\diamond
 \end{array}$$

Identity Laws To establish that $m_P \circ u_{P^\diamond} = \text{id}_{P^\diamond} = m_P \circ (u_P)^\diamond$, we will prove that all three maps are $(\mathcal{I} + - \star P)$ -algebra morphisms from the initial algebra to itself. For $\text{id}_{P^\diamond}$ this is immediate.

For $m_P \circ u_{P^\diamond}$, we only need to use that $(-)^{\diamond}$ is a functor and [Definition B.10](#).

$$\begin{array}{ccc}
 \mathcal{I} + P^\diamond \star P & \xrightarrow{\text{in}} & P^\diamond \\
 \downarrow \text{id} + (u_P)^\diamond \star \text{id} & & \downarrow (u_P)^\diamond \\
 \mathcal{I} + P^{\diamond\diamond} \star P & \xrightarrow{\text{id} + \text{id} \star u_P} \mathcal{I} + P^{\diamond\diamond} \star P^\diamond \xrightarrow{\text{in}} & P^{\diamond\diamond} \\
 \downarrow \text{id} + m_P \star \text{id} & \downarrow \text{id} + m_P \star \text{id} & \downarrow m_P \\
 \mathcal{I} + P^\diamond \star P & \xrightarrow{\text{id} + \text{id} \star u_P} \mathcal{I} + P^\diamond \star P^\diamond \xrightarrow{[\eta_P, \mu_P]} & P^\diamond \\
 & \searrow \text{in} & \uparrow
 \end{array}$$

For $m_P \circ (u_P)^\diamond$, we can establish that it is the identity directly. First, the following diagram commutes.

$$\begin{array}{ccccccc}
 P^\diamond & & & & & & \\
 \downarrow \lambda^{-1} & & & & \searrow u_{P^\diamond} & & \\
 \mathcal{I} \star P^\diamond & & & & & & \\
 \downarrow \iota_2 \star \text{id} & \searrow \eta_{P^\diamond} \star \text{id} & & & & & \\
 (\mathcal{I} + P^{\diamond\diamond} \star P^\diamond) \star P^\diamond & \xrightarrow{\text{in} \star \text{id}} & P^{\diamond\diamond} \star P^\diamond & \xleftarrow{\iota_1} & \mathcal{I} + P^{\diamond\diamond} \star P^\diamond & \xrightarrow{\text{in}} & P^{\diamond\diamond} \\
 \downarrow (\text{id} + m_P \star \text{id}) \star \text{id} & & \downarrow m_P \star \text{id} & & \downarrow \text{id} + m_P \star \text{id} & & \downarrow m_P \\
 (\mathcal{I} + P^\diamond \star P^\diamond) \star P^\diamond & \xrightarrow{[\eta_P, \mu_P] \star \text{id}} & P^\diamond \star P^\diamond & \xleftarrow{\iota_1} & \mathcal{I} + P^\diamond \star P^\diamond & \xrightarrow{[\eta_P, \mu_P]} & P^\diamond \\
 & & \searrow \mu_P & & & & \uparrow
 \end{array}$$

Then using the laws for monoids, the outer paths $P^\diamond \rightarrow P^\diamond$ prove that

$$\begin{aligned}
 m_P \circ u_{P^\diamond} &= \mu \circ ([\eta, \mu] \star \text{id}) \circ (\iota_1 \star \text{id}) \circ \lambda^{-1} \\
 &= \mu \circ (\eta \star \text{id}) \circ \lambda^{-1} \\
 &= \lambda \circ \lambda^{-1} \\
 &= \text{id}.
 \end{aligned}$$

□

Conjecture B.14. *It seems likely that, appropriately defined, $\langle\langle P^\diamond \rangle\rangle$ is the free fibred polynomial monad on a polynomial, but that is left for future work.*

Bibliography

- Abbott, Michael Gordon (2003). ‘Categories of Containers’. PhD thesis. University of Leicester. URL: <https://hdl.handle.net/2381/30102> (visited on 09/10/2026) (cit. on pp. 4, 36, 39).
- Abbott, Michael Gordon, Thorsten Altenkirch and Neil Ghani (2003). ‘Categories of Containers’. In: *FOSSACS 2003 proceedings*. Ed. by Andrew D. Gordon. Vol. 2620, pp. 23–38. DOI: [10.1007/3-540-36576-1_2](https://doi.org/10.1007/3-540-36576-1_2) (cit. on pp. iii, 2, 4, 5, 36, 39, 47).
- (2005). ‘Containers: Constructing strictly positive types’. In: *Theoretical Computer Science* 342.1, pp. 3–27. DOI: [10.1016/j.tcs.2005.06.002](https://doi.org/10.1016/j.tcs.2005.06.002) (cit. on p. 5).
- Abbott, Michael Gordon, Thorsten Altenkirch, Neil Ghani and Conor McBride (2003). ‘Derivatives of Containers’. In: *Typed Lambda Calculi and Applications*. Ed. by Martin Hofmann, pp. 16–30. DOI: [10.1007/3-540-44904-3_2](https://doi.org/10.1007/3-540-44904-3_2) (cit. on p. 110).
- Abou Samra, Jean and David Alexander Madore (2026). *An order-reversing embedding of Turing degrees into Arthur-Nimue-Merlin degrees*. arXiv: [2603.19946](https://arxiv.org/abs/2603.19946) (cit. on p. 109).
- Ahman, Danel and Andrej Bauer (2025). ‘Comodule representations of second-order functionals’. In: *Journal of Logical and Algebraic Methods in Programming* 146, p. 101071. DOI: [10.1016/j.jlamp.2025.101071](https://doi.org/10.1016/j.jlamp.2025.101071) (cit. on p. 7).
- (2026). *Sheaves as oracle computations*. arXiv: [2602.22135](https://arxiv.org/abs/2602.22135) (cit. on p. 7).
- Altenkirch, Thorsten, Neil Ghani, Peter Hancock, Conor McBride and Peter Morris (2015). ‘Indexed containers’. In: *Journal of Functional Programming* 25, e5. DOI: [10.1017/S095679681500009X](https://doi.org/10.1017/S095679681500009X) (cit. on p. 39).
- Altenkirch, Thorsten, Paul Levy and Sam Staton (2010). ‘Higher-Order Containers’. In: *Programs, Proofs, Processes*. Ed. by Fernando Ferreira, Benedikt Löwe, Elvira Mayordomo and Luís Mendes Gomes, pp. 11–20. DOI: [10.1007/978-3-642-13962-8_2](https://doi.org/10.1007/978-3-642-13962-8_2) (cit. on p. 63).
- Bauer, Andrej (2000). ‘The realizability approach to computable analysis and topology’. PhD thesis. Carnegie Mellon University. URL: <https://math.andrej.com/2000/09/20/the-realizability-approach-to-computable-analysis-and-topology/> (visited on 09/10/2026) (cit. on pp. 4, 13, 15, 22, 113).

- Bauer, Andrej (2022a). ‘Instance reducibility and Weihrauch degrees’. In: *Logical Methods in Computer Science* 18.3. DOI: [10.46298/LMCS-18\(3:20\)2022](https://doi.org/10.46298/LMCS-18(3:20)2022) (cit. on pp. 4, 6, 42, 43, 49).
- (2022b). *Notes on Realizability*. Lecture notes for the Midlands Graduate School, Nottingham, UK. URL: <https://www.andrej.com/zapiski/MGS-2022/notes-on-realizability.pdf> (visited on 09/10/2026) (cit. on pp. 4, 21).
- Van den Berg, Benno and Federico De Marchi (2007). ‘Non-well-founded trees in categories’. In: *Annals of Pure and Applied Logic* 146.1, pp. 40–59. DOI: [10.1016/j.apal.2006.12.001](https://doi.org/10.1016/j.apal.2006.12.001) (cit. on p. 5).
- Bergeron, François, Gilbert Labelle and Pierre Leroux (1998). *Combinatorial species and tree-like structures*. Trans. by Margaret Readdy. Encyclopedia of Mathematics and its Applications 67. Cambridge University Press. DOI: [10.1017/CB09781107325913](https://doi.org/10.1017/CB09781107325913) (cit. on p. 2).
- Bonchi, Filippo, Daniela Petrişan, Damien Pous and Jurriaan Rot (2014). ‘Coinduction up-to in a fibrational setting’. In: *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. DOI: [10.1145/2603088.2603149](https://doi.org/10.1145/2603088.2603149) (cit. on pp. 5, 113).
- Braithwaite, Dylan, Matteo Capucci, Bruno Gavranović, Jules Hedges and Eigil Fjeldgren Rischel (2021). *Fibre optics*. arXiv: [2112.11145](https://arxiv.org/abs/2112.11145) (cit. on p. 6).
- (2024). ‘Fibre optics II’. Unpublished draft (cit. on pp. iii, 6, 85–87).
- Brattka, Vasco (2003). ‘Computability over Topological Structures’. In: *Computability and Models: Perspectives East and West*. Ed. by S. Barry Cooper and Sergey S. Goncharov, pp. 93–136. DOI: [10.1007/978-1-4615-0755-0_5](https://doi.org/10.1007/978-1-4615-0755-0_5) (cit. on p. 61).
- (2021). ‘Stashing And Parallelization Pentagons’. In: *Logical Methods in Computer Science* 17, 20 (4). DOI: [10.46298/lmcs-17\(4:20\)2021](https://doi.org/10.46298/lmcs-17(4:20)2021) (cit. on p. 109).
- (2022). ‘Weihrauch Complexity and the Hagen School of Computable Analysis’. In: *60 Jahre DVMLG*. Ed. by Deniz Sarikaya and Benedikt Löwe. URL: <https://www.math.uni-hamburg.de/spag/ml/DVMLG/dvmlg60.html> (visited on 09/10/2026) (cit. on p. 30).
- (2026a). ‘Loops, Inverse Limits and Non-Determinism’. In: *Logical Methods in Computer Science* Volume 22, Issue 4, 3. DOI: [10.46298/lmcs-22\(4:3\)2026](https://doi.org/10.46298/lmcs-22(4:3)2026) (cit. on pp. 5, 73, 76).
- (2026b). *Bibliography on Weihrauch Complexity*. URL: <https://cca-net.de/publications/weibib.php> (visited on 24/10/2026) (cit. on p. 1).
- Brattka, Vasco and Guido Gherardi (2011a). ‘Effective Choice and Boundedness Principles in Computable Analysis’. In: *The Bulletin of Symbolic Logic* 17.1, pp. 73–117. DOI: [10.2178/bsl/1294186663](https://doi.org/10.2178/bsl/1294186663) (cit. on pp. 17, 31, 33, 34).

- (2011b). ‘Weihrauch Degrees, Omniscience Principles and Weak Computability’. In: *Journal of Symbolic Logic* 76 (1), pp. 143–176. DOI: [10.2178/jsl/1294170993](https://doi.org/10.2178/jsl/1294170993). (Visited on 11/10/2026) (cit. on pp. [59](#), [62](#), [95](#)).
- Brattka, Vasco, Guido Gherardi and Alberto Marcone (2012). ‘The Bolzano–Weierstrass Theorem is the jump of Weak König’s Lemma’. In: *Annals of Pure and Applied Logic* 163.6, pp. 623–655. DOI: [10.1016/j.apal.2011.10.006](https://doi.org/10.1016/j.apal.2011.10.006) (cit. on p. [34](#)).
- Brattka, Vasco, Guido Gherardi and Arno Pauly (2021). ‘Weihrauch Complexity in Computable Analysis’. In: *Handbook of Computability and Complexity in Analysis*. Ed. by Vasco Brattka and Peter Hertling, pp. 367–417. DOI: [10.1007/978-3-030-59234-9_11](https://doi.org/10.1007/978-3-030-59234-9_11) (cit. on pp. [iii](#), [1](#), [2](#), [4](#), [5](#), [8](#), [30](#), [49](#), [59](#), [109](#)).
- Brattka, Vasco and Arno Pauly (2018). ‘On the algebraic structure of Weihrauch degrees’. In: *Logical Methods in Computer Science* 14.4. DOI: [10.23638/LMCS-14\(4:4\)2018](https://doi.org/10.23638/LMCS-14(4:4)2018) (cit. on pp. [50](#), [53](#), [108](#)).
- Brattka, Vasco and Tahina Rakotoniana (2017). ‘On the Uniform Computational Content of Ramsey’s Theorem’. In: *The Journal of Symbolic Logic* 82.4, pp. 1278–1316. DOI: [10.1017/jsl.2017.43](https://doi.org/10.1017/jsl.2017.43) (cit. on p. [34](#)).
- Brattka, Vasco and Hendrik Smischlaew (2025). ‘Computability of Initial Value Problems’. In: *Crossroads of Computability and Logic: Insights, Inspirations, and Innovations*. Ed. by Arnold Beckmann, Isabel Oitavem and Florin Manea, pp. 191–206. DOI: [10.1007/978-3-031-95908-0_14](https://doi.org/10.1007/978-3-031-95908-0_14) (cit. on p. [34](#)).
- Carboni, Aurelio, Stephen Lack and Robert F. C. Walters (1993). ‘Introduction to extensive and distributive categories’. In: *Journal of Pure and Applied Algebra* 84.2, pp. 145–158. DOI: [10.1016/0022-4049\(93\)90035-R](https://doi.org/10.1016/0022-4049(93)90035-R) (cit. on pp. [47](#), [48](#), [63](#), [64](#), [96](#)).
- Carboni, Aurelio and Giuseppe Rosolini (2000). ‘Locally cartesian closed exact completions’. In: *Journal of Pure and Applied Algebra* 154.1, pp. 103–116. DOI: [10.1016/S0022-4049\(99\)00192-9](https://doi.org/10.1016/S0022-4049(99)00192-9) (cit. on p. [25](#)).
- Carl, Merlin (2016). ‘Generalized Effective Reducibility’. In: *Pursuit of the Universal*. Ed. by Arnold Beckmann, Laurent Bienvenu and Nataša Jonska, pp. 225–233. DOI: [10.1007/978-3-319-40189-8_23](https://doi.org/10.1007/978-3-319-40189-8_23) (cit. on p. [107](#)).
- (2026). ‘Complexities of Effective Reductions with Ordinal Turing Machines’. In: *Timeless Machines: Computability Across Eras*. Ed. by Vasco Brattka, Henning Fernau and Lorenzo Galeotti, pp. 178–192. DOI: [10.1007/978-3-032-31348-5_12](https://doi.org/10.1007/978-3-032-31348-5_12) (cit. on p. [107](#)).
- Cipriani, Vittorio and Arno Pauly (2025). ‘Embeddability of graphs and Weihrauch degrees’. In: *Journal of Mathematical Logic*, p. 2550011. DOI: [10.1142/S0219061325500114](https://doi.org/10.1142/S0219061325500114) (cit. on p. [109](#)).
- Day, Brian (1970). ‘On closed categories of functors’. In: *Reports of the Midwest Category Seminar IV*. Ed. by S. MacLane et al., pp. 1–38. DOI: [10.1007/BFb0060438](https://doi.org/10.1007/BFb0060438) (cit. on p. [110](#)).

- Dorais, François, Damir D. Dzhafarov, Jeffrey Hirst, Joseph Mileti and Paul Shafer (2015). ‘On uniform relationships between combinatorial problems’. In: *Transactions of the American Mathematical Society* 368.2, pp. 1321–1359. DOI: [10.1090/tran/6465](https://doi.org/10.1090/tran/6465) (cit. on pp. 1, 4, 34).
- Dzhafarov, Damir D. (2019). ‘Joins in the strong Weihrauch degrees’. In: *Mathematical Research Letters* 26.3, pp. 749–767. DOI: [10.4310/MRL.2019.v26.n3.a5](https://doi.org/10.4310/MRL.2019.v26.n3.a5) (cit. on pp. 6, 96, 104, 109).
- Dzhafarov, Damir D. and Carl Mummert (2022). *Reverse Mathematics. Problems, Reductions, and Proofs*. Theory and Applications of Computability. Springer Cham. DOI: [10.1007/978-3-031-11367-3](https://doi.org/10.1007/978-3-031-11367-3) (cit. on p. 1).
- Dzhafarov, Damir D., Reed Solomon and Keita Yokoyama (2024). ‘On the first-order parts of problems in the Weihrauch degrees’. In: *Computability* 13.3–4, pp. 363–375. DOI: [10.3233/COM-230446](https://doi.org/10.3233/COM-230446) (cit. on p. 109).
- Fiore, Marcelo, Zeinab Galal and Farzad Jafarrahmani (2023). ‘Fixpoint constructions in focused orthogonality models of linear logic’. In: vol. 3, 8. DOI: [10.46298/entics.12302](https://doi.org/10.46298/entics.12302) (cit. on p. 5).
- Freyd, Peter John and Scedrov Andre (1990). *Categories, Allegories*. North Holland. ISBN: 9780444703682 (cit. on p. 84).
- Friedberg, Richard M. and Hartley Rogers jr. (1959). ‘Reducibility and Completeness for Sets of Integers’. In: *Mathematical Logic Quarterly* 5.7–13, pp. 117–125. DOI: <https://doi.org/10.1002/malq.19590050703> (cit. on p. 107).
- Friedman, Harvey (1971). ‘Higher set theory and mathematical practice’. In: *Annals of Mathematical Logic* 2.3, pp. 325–357. DOI: [10.1016/0003-4843\(71\)90018-0](https://doi.org/10.1016/0003-4843(71)90018-0) (cit. on p. 110).
- (1975). ‘Some systems of second order arithmetic and their use’. In: *Proceedings of the International Congress of Mathematicians (Vancouver, B.C., 1974)*, Vol. 1, pp. 235–242. URL: <https://www.mathunion.org/icm/proceedings> (visited on 09/10/2026) (cit. on p. 1).
- Fujiwara, Makoto (2021). ‘Weihrauch and constructive reducibility between existence statements’. In: *Computability* 10.1, pp. 17–30. DOI: [10.3233/COM-190278](https://doi.org/10.3233/COM-190278) (cit. on p. 108).
- Galeotti, Lorenzo and Hugo Nobrega (2017). ‘Towards Computable Analysis on the Generalised Real Line’. In: *Unveiling Dynamics and Complexity*. Ed. by Jarkko Kari, Florin Manea and Ion Petre, pp. 246–257. DOI: [10.1007/978-3-319-58741-7_24](https://doi.org/10.1007/978-3-319-58741-7_24) (cit. on p. 107).
- Gambino, Nicola and Martin Hyland (2004). ‘Wellfounded Trees and Dependent Polynomial Functors’. In: *Types for Proofs and Programs*. Ed. by Stefano Berardi, Mario Coppo and Ferruccio Damiani, pp. 210–225. DOI: [10.1007/978-3-540-24849-1_14](https://doi.org/10.1007/978-3-540-24849-1_14). (Visited on 30/07/2025) (cit. on pp. 63, 69).
- Gambino, Nicola and Joachim Kock (2012). ‘Polynomial functors and polynomial monads’. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 152.1, pp. 1–14. DOI: [10.1017/S0004972611000488](https://doi.org/10.1017/S0004972611000488) (cit. on p. 107).

- ical Society* 154.1, pp. 153–192. DOI: [10.1017/S0305004112000394](https://doi.org/10.1017/S0305004112000394) (cit. on pp. [iii](#), [2](#), [4](#), [5](#), [39](#), [40](#), [53](#), [57](#), [63](#), [64](#), [73](#), [121](#), [126](#)).
- Gherardi, Guido and Alberto Marcone (2009). ‘How Incomputable is the Separable Hahn–Banach Theorem?’ In: *Notre Dame Journal of Formal Logic* 50.4, pp. 393–425. DOI: [10.1215/00294527-2009-018](https://doi.org/10.1215/00294527-2009-018) (cit. on p. [1](#)).
- Goh, June Le, Arno Pauly and Manlio Valenti (2021). ‘Finding Descending Sequences Through Ill-Founded Linear Orders’. In: *The Journal of Symbolic Logic* 86.2, pp. 817–854. DOI: [10.1017/jsl.2021.15](https://doi.org/10.1017/jsl.2021.15) (cit. on p. [109](#)).
- Hamkins, Joel David and Andy Lewis (2000). ‘Infinite Time Turing Machines’. In: *The Journal of Symbolic Logic* 65.2, pp. 567–604. DOI: [10.2307/2586556](https://doi.org/10.2307/2586556). (Visited on 22/10/2026) (cit. on p. [107](#)).
- Hasuo, Ichiro, Toshiki Kataoka and Kenta Cho (2018). ‘Coinductive predicates and final sequences in a fibration’. In: *Mathematical Structures in Computer Science* 28.4, pp. 562–611. DOI: [10.1017/S0960129517000056](https://doi.org/10.1017/S0960129517000056) (cit. on p. [5](#)).
- Hedges, Jules (2018). *Lenses for philosophers*. URL: <https://julesh.com/posts/2018-08-16-lenses-for-philosophers.html> (visited on 20/08/2026) (cit. on p. [2](#)).
- (2025). *We solved dependent optics!* Recorded seminar talk. URL: <https://www.youtube.com/watch?v=yhxxUnWKK2I> (visited on 23/03/2026) (cit. on p. [6](#)).
- Higuchi, Kojiro and Arno Pauly (2013). ‘The degree-structure of Weihrauch-reducibility’. In: *Logical Methods in Computer Science* 9.2. DOI: [10.2168/LMCS-9\(2:2\)2013](https://doi.org/10.2168/LMCS-9(2:2)2013) (cit. on p. [109](#)).
- Hinman, Peter G. (2012). ‘A Survey of Mučnik and Medvedev Degrees’. In: *The Bulletin of Symbolic Logic* 18.2, pp. 161–229. DOI: [10.2178/bsl/1333560805](https://doi.org/10.2178/bsl/1333560805). (Visited on 08/06/2026) (cit. on p. [58](#)).
- Hirsch, Michael David (1990). ‘Applications of topology to lower bound estimates in computer science’. PhD thesis. University of California, Berkeley (cit. on pp. [iii](#), [2](#)).
- Hua, Joseph and Yiming Xu (2026). *Polynomial functors in π -clans for the semantics of type theory*. arXiv: [2602.05689](https://arxiv.org/abs/2602.05689) (cit. on p. [54](#)).
- Jacobs, Bart (1999). *Categorical Logic and Type Theory*. Studies in logic and the foundations of mathematics. Elsevier. ISBN: 978-0444501707 (cit. on pp. [8](#), [36](#), [116](#), [118](#), [119](#)).
- Jay, C. Barry and J. R. B. Cockett (1994). ‘Shapely types and shape polymorphism’. In: *Programming Languages and Systems — ESOP ’94*. Ed. by Donald Sannella, pp. 302–316. DOI: [10.1007/3-540-57880-3_20](https://doi.org/10.1007/3-540-57880-3_20) (cit. on p. [2](#)).
- Johnstone, Peter T. (2002). *Sketches of an Elephant: A Topos Theory Compendium*. Oxford University Press. DOI: [10.1093/oso/9780198515982.001.0001](https://doi.org/10.1093/oso/9780198515982.001.0001) (cit. on pp. [20](#), [84](#), [99](#), [101](#), [111](#)).

- De Jong, Tom (2024). *Categorical Realizability*. Lecture notes and exercises for the Midlands Graduate School, 8 – 12 April 2024, Leicester, UK. URL: <https://github.com/tomdjong/MGS-categorical-realizability> (visited on 09/10/2026) (cit. on p. 4).
- Joyal, André (1981). ‘Une Théorie Combinatoire des Séries Formelles’. In: *Advances in Mathematics* 42.1, pp. 1–82. DOI: [10.1016/0001-8708\(81\)90052-9](https://doi.org/10.1016/0001-8708(81)90052-9) (cit. on p. 2).
- (2017). *Notes on Clans and Tribes*. arXiv: [1710.10238](https://arxiv.org/abs/1710.10238) (cit. on p. 54).
- Kainen, Paul C (1971). ‘Weak adjoint functors’. In: *Mathematische Zeitschrift* 122, pp. 1–9. DOI: [10.1007/BF01113560](https://doi.org/10.1007/BF01113560) (cit. on p. 24).
- Kajikawa, Akihito, Masamori Kaku, Takayuki Kihara and Satoshi Nakata (2026). *Modified realizability subtoposes and total Weihrauch reducibility*. arXiv: [2608.02133](https://arxiv.org/abs/2608.02133) (cit. on pp. 108, 109).
- Kawamura, Akitoshi and Stephen Cook (2012). ‘Complexity Theory for Operators in Analysis’. In: *ACM Transactions on Computation Theory* 4.2. DOI: [10.1145/2189778.2189780](https://doi.org/10.1145/2189778.2189780) (cit. on p. 108).
- Kawamura, Akitoshi and Arno Pauly (2014). ‘Function Spaces for Second-Order Polynomial Time’. In: *Language, Life, Limits*. Ed. by Arnold Beckmann, Erzsébet Csuhaj-Varjú and Klaus Meer, pp. 245–254. DOI: [10.1007/978-3-319-08019-2_25](https://doi.org/10.1007/978-3-319-08019-2_25) (cit. on p. 108).
- Kelly, G Max (1980). ‘A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on’. In: *Bulletin of the Australian Mathematical Society* 22.1, pp. 1–83. DOI: [10.1017/S0004972700006353](https://doi.org/10.1017/S0004972700006353) (cit. on pp. 2, 5, 73).
- Kihara, Takayuki (2023). ‘Lawvere–Tierney topologies for computability theorists’. In: *Transactions of the American Mathematical Society Series B* 10.2, pp. 48–85. DOI: [10.1090/btran/134](https://doi.org/10.1090/btran/134) (cit. on pp. 8, 107, 109).
- (2024). *Rethinking the notion of oracle: A prequel to Lawvere–Tierney topologies for computability theorists*. arXiv: [2202.00188](https://arxiv.org/abs/2202.00188) (cit. on pp. 8, 43, 109).
- Kmetz, Edward and contributors (2012). *lens: Lenses, Folds and Traversals*. URL: <https://hackage.haskell.org/package/lens> (visited on 24/10/2026) (cit. on pp. iii, 2).
- Koepeke, Peter (2005). ‘Turing Computations on Ordinals’. In: *The Bulletin of Symbolic Logic* 11.3, pp. 377–397. DOI: [10.2178/bsl/1122038993](https://doi.org/10.2178/bsl/1122038993). (Visited on 22/10/2026) (cit. on p. 107).
- Kohlenbach, Ulrich (2008). *Applied Proof Theory. Proof Interpretations and their Use in Mathematics*. Springer Monographs in Mathematics. Springer. DOI: [10.1007/978-3-540-77533-1](https://doi.org/10.1007/978-3-540-77533-1) (cit. on p. 8).
- Kozen, Dexter and Alexandra Silva (2012). ‘Left-Handed Completeness’. In: *Relational and Algebraic Methods in Computer Science*. Ed. by Wolfram Kahl and Timothy G. Griffin, pp. 162–178. DOI: [10.1007/978-3-642-33314-9_11](https://doi.org/10.1007/978-3-642-33314-9_11) (cit. on p. 76).

- Lambek, Joachim (1968). ‘A fixpoint theorem for complete categories’. In: *Mathematische Zeitschrift* 103.2, pp. 151–161. DOI: [10.1007/BF01110627](https://doi.org/10.1007/BF01110627) (cit. on p. 112).
- Mac Lane, Saunders (1978). *Categories for the working mathematician*. Vol. 5. Graduate Texts in Mathematics. Springer New York, NY. DOI: [10.1007/978-1-4757-4721-8](https://doi.org/10.1007/978-1-4757-4721-8) (cit. on pp. 9, 111).
- Mac Lane, Saunders and Ieke Moerdijk (1994). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Universitext. Springer. DOI: [10.1007/978-1-4612-0927-0](https://doi.org/10.1007/978-1-4612-0927-0). (Visited on 15/05/2026) (cit. on p. 18).
- Maschio, Samuele and Davide Trotta (2026). ‘A topos for extended Weihrauch degrees’. In: *Annals of Pure and Applied Logic* 177.9. DOI: [10.1016/j.apal.2026.103781](https://doi.org/10.1016/j.apal.2026.103781) (cit. on p. 8).
- Medvedev, Yuri Tikhonovich (1955). ‘Degrees of difficulty of the mass problem’. In: *Doklady Akademii Nauk SSSR* 104, pp. 501–504. URL: <https://search.rsl.ru/ru/record/01005892789> (visited on 12/10/2026) (cit. on p. 58).
- Moerdijk, Ieke and Erik Palmgren (2000). ‘Wellfounded trees in categories’. In: *Annals of Pure and Applied Logic* 104.1, pp. 189–218. DOI: [10.1016/S0168-0072\(00\)00012-9](https://doi.org/10.1016/S0168-0072(00)00012-9) (cit. on p. 5).
- Moskovich, Daniel (2013). *What is the source of this famous Grothendieck quote?* MathOverflow. URL: <https://mathoverflow.net/q/133020> (visited on 22/10/2026) (cit. on p. 108).
- Neumann, Eike and Arno Pauly (2018). ‘A topological view on algebraic computation models’. In: *Journal of Complexity* 44, pp. 1–22. DOI: [10.1016/j.jco.2017.08.003](https://doi.org/10.1016/j.jco.2017.08.003) (cit. on pp. 2, 73).
- Neumann, Eike, Arno Pauly and Cécilia Pradic (2024). ‘The equational theory of the Weihrauch lattice with multiplication’. In: arXiv: [2403.13975](https://arxiv.org/abs/2403.13975) (cit. on p. 52).
- Nguyễn, Lê Thành Dũng, Camille Noûs and Cécilia Pradic (2020). ‘Implicit automata in typed λ -calculi II: streaming transducers vs categorical semantics’. In: arXiv: [2008.01050](https://arxiv.org/abs/2008.01050) (cit. on p. vii).
- (2021). ‘Comparison-Free Polyregular Functions’. In: *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Ed. by Nikhil Bansal, Emanuela Merelli and James Worrell. Vol. 198, 139:1–139:20. DOI: [10.4230/LIPIcs.ICALP.2021.139](https://doi.org/10.4230/LIPIcs.ICALP.2021.139) (cit. on p. vii).
- Nguyễn, Lê Thành Dũng and Cécilia Pradic (2020). ‘Implicit automata in typed λ -calculi I: aperiodicity in a non-commutative logic’. In: *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8–11, 2020, Saarbrücken, Germany (Virtual Conference)*. Ed. by Artur Czumaj, Anuj Dawar and Emanuela Merelli. Vol. 168, 135:1–135:20. DOI: [10.4230/LIPIcs.ICALP.2020.135](https://doi.org/10.4230/LIPIcs.ICALP.2020.135) (cit. on p. vii).
- Niu, Nelson and David I. Spivak (2025). *Polynomial Functors: A Mathematical Theory of Interaction*. Cambridge University Press. DOI: [10.1017/](https://doi.org/10.1017/)

9781009576734. URL: <https://topos.site/poly-book.pdf> (visited on 09/10/2026) (cit. on pp. vii, 4, 5, 53, 64, 108).
- Nobrega, Hugo and Arno Pauly (2019). ‘Game characterizations and lower cones in the Weihrauch degrees’. In: *Logical Methods in Computer Science* Volume 15, Issue 3, 11. DOI: [10.23638/LMCS-15\(3:11\)2019](https://doi.org/10.23638/LMCS-15(3:11)2019) (cit. on p. 8).
- Van Oosten, Jaap (2008). *Realizability: An Introduction to its Categorical Side*. Vol. 152. Studies in Logic and the Foundations of Mathematics. Elsevier. ISBN: 978-0-444-51584-1 (cit. on pp. 4, 7, 11, 12, 15, 17, 22, 42, 108, 117, 120).
- Van Oosten, Jaap and Tingxiang Zou (2016). ‘Classical and Relative Realizability’. In: *Theory and Applications of Categories* 31.22, pp. 571–593. URL: <https://tac.mta.ca/tac/volumes/31/22/31-22abs.html> (visited on 09/10/2026) (cit. on p. 13).
- De Paiva, Valeria (1991). *The Dialectica categories*. Tech. rep. UCAM-CL-TR-213. University of Cambridge. DOI: [10.48456/tr-213](https://doi.org/10.48456/tr-213) (cit. on p. 8).
- Pauly, Arno (2010). ‘How Incomputable is Finding Nash Equilibria?’ In: *Journal of Universal Computer Science* 16.18, pp. 2686–2710. DOI: [10.3217/jucs-016-18-2686](https://doi.org/10.3217/jucs-016-18-2686) (cit. on p. 61).
- (2016). ‘On the topological aspects of the theory of represented spaces’. In: *Computability* 5.2, pp. 159–180. DOI: [10.3233/COM-150049](https://doi.org/10.3233/COM-150049) (cit. on pp. 4, 28, 29).
- (2017). ‘Many-one reductions and the category of multivalued functions’. In: *Mathematical Structures in Computer Science* 27.3, pp. 376–404. DOI: [10.1017/S0960129515000262](https://doi.org/10.1017/S0960129515000262) (cit. on pp. 6, 29).
- Pauly, Arno and Matthew de Brecht (2015). ‘Descriptive Set Theory in the Category of Represented Spaces’. In: *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 438–449. DOI: [10.1109/LICS.2015.48](https://doi.org/10.1109/LICS.2015.48) (cit. on p. 7).
- Pradic, Cécilia (2025). *The equational theory of the Weihrauch lattice with (iterated) composition*. arXiv: [2408.14999](https://arxiv.org/abs/2408.14999) (cit. on pp. vii, 76).
- Pradic, Cécilia and Ian Price (2024). ‘Implicit automata in λ -calculi III: affine planar string-to-string functions’. In: *Electronic Notes in Theoretical Informatics and Computer Science* Volume 4 – Proceedings of MFPS XL, 19. DOI: [10.46298/entics.14804](https://doi.org/10.46298/entics.14804) (cit. on p. vii).
- (2025). ‘Weihrauch Problems as Containers’. In: *Crossroads of Computability and Logic: Insights, Inspirations, and Innovations*. Ed. by Arnold Beckmann, Isabel Oitavem and Florin Manea, pp. 395–409. DOI: [10.1007/978-3-031-95908-0_28](https://doi.org/10.1007/978-3-031-95908-0_28) (cit. on pp. vii, 4, 41).
- (2026). ‘Problems with Fixpoints of Polynomials of Polynomials’. In: *41st Annual Symposium on Logic in Computer Science (LICS 2026)*. Ed. by Claudia Faggian and Joost-Pieter Katoen. Vol. 380, 77:1–77:27. DOI: [10.4230/LIPIcs.LICS.2026.77](https://doi.org/10.4230/LIPIcs.LICS.2026.77) (cit. on pp. vii, 5, 7, 8, 79, 106).

- Price, Ian (2024). *Weihrauch Reducibility as a Lens*. URL: <https://www.countingishard.org/blog/weihrauch-reducibility-as-a-lens> (visited on 07/10/2026) (cit. on p. vii).
- (2025). *Strong Reducibility as an Adaptor*. URL: <https://www.countingishard.org/blog/strong-reducibility-as-an-adaptor> (visited on 23/03/2026) (cit. on p. 6).
- Rauzy, Emmanuel (2025). ‘Multi-representation associated to the numbering of a subbasis and formal inclusion relations’. In: *Logical Methods in Computer Science* 21, 29 (3). DOI: [10.46298/lmcs-21\(3:29\)2025](https://doi.org/10.46298/lmcs-21(3:29)2025) (cit. on p. 109).
- Riehl, Emily (2008). *Factorization Systems*. Notes for the Algebraic Topology and Category Theory Proseminar at the University of Chicago. URL: <https://emilyriehl.github.io/files/factorization.pdf> (visited on 22/10/2026) (cit. on p. 115).
- Rijke, Egbert, Michael Shulman and Bas Spitters (2020). ‘Modalities in homotopy type theory’. In: *Logical Methods in Computer Science* 16.1. DOI: [10.23638/LMCS-16\(1:2\)2020](https://doi.org/10.23638/LMCS-16(1:2)2020) (cit. on p. 7).
- Rogers, Hartley (1987). *Theory of Recursive Functions and Effective Computability*. MIT Press. ISBN: 9780262680523 (cit. on pp. 58, 107).
- Sanders, Sam (2026). *A Notion of Reducibility for Third-Order Arithmetic*. Presentation at *Computability and Complexity in Analysis 2026*. URL: <https://cca-net.de/cca2026/slides/CCA-2026-3.2b-Sanders.pdf> (visited on 22/10/2026) (cit. on p. 108).
- Schröder, Matthias (2002). ‘Admissible Representations for Continuous Computations’. PhD thesis. University of Hagen. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00000764 (visited on 23/10/2026) (cit. on p. 109).
- Scott, Dana (1976). ‘Data Types as Lattices’. In: *SIAM Journal on Computing* 5.3, pp. 522–587. DOI: [10.1137/0205037](https://doi.org/10.1137/0205037) (cit. on pp. 13, 107).
- Seely, Robert A. G. (1984). ‘Locally cartesian closed categories and type theory’. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 95.1, pp. 33–48. DOI: [10.1017/S0305004100061284](https://doi.org/10.1017/S0305004100061284) (cit. on p. 39).
- Simpson, Stephen G. (1999). *Subsystems of second order arithmetic*. Perspectives in Mathematical Logic. Cambridge University Press, pp. xiv+445. DOI: [10.1017/CB09780511581007](https://doi.org/10.1017/CB09780511581007) (cit. on pp. 1, 17).
- Sipser, Michael (1997). *Introduction to the Theory of Computation*. PWS Publishing Company. ISBN: 9780534947286 (cit. on p. 9).
- Soldà, Giovanni and Manlio Valenti (2023). ‘Algebraic properties of the first-order part of a problem’. In: *Annals of Pure and Applied Logic* 174.7, p. 103270. DOI: [10.1016/j.apal.2023.103270](https://doi.org/10.1016/j.apal.2023.103270) (cit. on p. 109).
- Soskova, Mariya and Alice Vidrine (2024). *Enumeration Weihrauch Reducibility*. Presentation at Victoria University Logic Seminar. URL: <https://>

- people.math.wisc.edu/~soskova/talks/EnumerationWeihrauch.pdf (visited on 28/10/2026) (cit. on p. 108).
- Spivak, David I. (2022). *Generalized Lens Categories via functors $\mathcal{C}^{\text{op}} \rightarrow \text{Cat}$* . arXiv: 1908.02202 (cit. on pp. 6, 41).
- (2024). *A reference for categorical structures on Poly*. arXiv: 2202.00534 (cit. on pp. 40, 48, 50, 51, 57, 58, 108, 110).
- Steinberg, Florian (2017). ‘Complexity theory for spaces of integrable functions’. In: *Logical Methods in Computer Science* 13, 21 (3). DOI: 10.23638/LMCS-13(3:21)2017 (cit. on p. 108).
- Streicher, Thomas (2023). *Fibered Categories à la Jean Bénabou*. arXiv: 1801.02927 (cit. on pp. 116, 119).
- Swan, Andrew W. (2023). *Intuition behind Kleene’s “second algebra” $\mathcal{K}_2$* . MathOverflow. URL: <https://mathoverflow.net/q/439062> (visited on 14/10/2026) (cit. on p. 14).
- (2024). ‘Oracle modalities’. In: arXiv: 2406.05818 (cit. on pp. 7, 107).
- Trotta, Davide, Manlio Valenti and Valeria de Paiva (2025). ‘Categorifying computable reducibilities’. In: *Logical Methods in Computer Science* 21.1, 15. DOI: 10.46298/lmcs-21(1:15)2025 (cit. on pp. 6, 8).
- Univalent Foundations Program, The (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. URL: <https://homotopytypetheory.org/book> (visited on 09/10/2026) (cit. on pp. 7, 71).
- Weber, Mark (2015). ‘Polynomials in categories with pullbacks’. In: *Theory and Applications of Categories* 30.16, pp. 533–598. URL: <http://www.tac.mta.ca/tac/volumes/30/16/30-16abs.html> (visited on 09/10/2026) (cit. on pp. 4, 39).
- Weihrauch, Klaus (1992). *The TTE-interpretation of three hierarchies of omniscience principles*. Informatik Berichte 130. FernUniversität Hagen. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001147 (visited on 09/10/2026) (cit. on p. 31).
- (2000). *Computable Analysis: An Introduction*. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-56999-9 (cit. on pp. 27–30).
- Westrick, Linda (2021). ‘A note on the diamond operator’. In: *Computability* 10.2, pp. 107–110. DOI: 10.3233/COM-200295 (cit. on pp. 5, 53, 75).
- Yamada, Rinta (2026). *Effectful Toposes and Their Lawvere-Tierney Topologies*. arXiv: 2602.23086 (cit. on p. 8).
- Yanofsky, Noson S. (2022). *Theoretical Computer Science for the Working Category Theorist*. Elements in Applied Category Theory. Cambridge University Press. DOI: 10.1017/9781108872348 (cit. on p. iii).