UNIVERSITY OF STRATHCLYDE

TYPES CATEGORIES ACTIONS

by

Timothy Revell

Submitted for the degree of
Doctor of Philosophy

Faculty of Science
Computer and Information Sciences

February 2016

DECLARATION OF AUTHORSHIP

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

I, Timothy Revell, declare that this thesis titled, *Types, Categories, Actions* and the work presented in it is my own, except where explicitly stated.

Signed: Revell

Date: February 23, 2016

"It's gonna be a really tough project. You're gonna have to use your head, your brain, and your mind, too."

Dewey Finn

UNIVERSITY OF STRATHCLYDE

Abstract

Faculty of Science
Computer and Information Sciences

Doctor of Philosophy

by Timothy Revell

This thesis explores relational parametricity using fibrations. We present a complementary view of Reynolds's relational parametricity using the relations fibration. This approach allows us to uncover some of the hidden categorical structure present in Reynolds's original definitions and results, leading to new insights in the study of parametricity. In a similar vain we provide an alternative parametric model of System F using group actions, which has some novel differences to the standard relational model. We then alter the type system leading to a general categorical framework for type systems with dimension types. We develop some informative models of this type theory, including a model based on group actions that captures invariance under scaling.

CREDITS

This thesis could not have been possible without the help and support of the people around me. I would like to thank my supervisor, Neil Ghani, for his guidance throughout my PhD. His knowledge and experience has been invaluable, but it is his enthusiasm and passion for mathematics that I have required (and admired) most. He has constantly inspired me to search for beautiful mathematics and has made my time as a PhD student incredibly enjoyable.

I would like to thank everyone in the MSP (Mathematically Structured Programming) group. Our regular MSP 101 talks provided me with the opportunity to learn, discuss, and present new ideas, as well as creating a wonderful working atmosphere. I am particularly thankful to Bob Atkey, Fredrik Nordvall Forsberg, Patricia Johann, Federico Orsanigo and Sam Staton for detailed conversations and joint work. I am also incredibly grateful for the comments on the early versions of this thesis by Fredrik Nordvall Forsberg and Neil Ghani.

I would like to thank the members of theGIST (Glasgow Insight into Science and Technology) for improving my writing abilities beyond recognition and for their social support. Without regular discussions with GISTers and their strictly regimented feedback, the quality of the writing in this thesis would have been substantially lower.

I would like to thank my family for their love and support throughout my PhD. I am especially grateful to my mother and father for their contributions to my mental (and financial) well-being, as well as to my sister for her love and friendship. I would also like to give a special thanks to my grandmother for always being able to provide a smile, a story and a lunch, and for her exuberantly positive attitude to life.

Finally, I would like to thank Emilie Steinmark for her love, support, and encouragement. Emilie, du er den eneste ene.

FOR THE LAYPERSON

Let's play a game. I'll give you an item and you can do whatever you like with it before returning it to me. Th only additional rule is that you have to tell me what you will do with the mystery item *before* I give it to you. This means that you will *never know* what the item is going to be *before* you have to say what you will do with it. I win if you can't carry out your action, otherwise you win. So let's play.

What are you going to do with the item? You scratch your head for a bit, and then reply "I'm going to rotate it by 90 degrees". I ponder your response and then decide to give you a plot of land, which you can't rotate, so I win. Let's play again.

You think for a moment and then say "OK, this time I will divide the item in two." If I give you the plot of land again, you can just put up a fence through the middle to the win the game. But I realise this and give you a fundamental particle, which cannot be divided in two, so I win again. "Shall we play once more?" I say.

Again you have a little think and then proclaim "Aha! I'm going to add a sticker to it. Whatever you give me I can always place a sticker on it. I win!" I think for a moment, and then give you an idea. Unfortunately for you, you can't add a sticker to an idea, so I win again.

"OK. One more game," you say. "I'm going to do nothing to the item, and then I will return it". There's no way for me to win, you can *always* do nothing. And so I give you a thesis, which you return in it's original condition, and you become the winner.

The trick to winning this game was to come up with an action for returning the item that could be done in every situation. If the action relied in any way on something specific about an item, for example it's ability to be rotated, divided, or have something added to it, then an item could always be chosen that wouldn't have the required property. There is only one action that can be chosen to win this game and it's the do-nothing approach.

In computer science there is an analogous situation. Sometimes, we wish to write computer programs that work for any possible input without any knowledge of the input itself. We call such programs *parametrically polymorphic*, and programs of this type are one of the main focuses of this thesis.

Sparing you the details (for those look in the thesis!) parametrically polymorphic functions arise in many different places throughout computer science and so it is important to study them and to understand them. By using a subject called *category theory* and a concept called *fibrations*, this is exactly what this thesis has contributed to. So if you wish, please read on. Hopefully you will agree that studying the do-nothing approach required quite a lot of work.

CONTENTS

NOTATION

Algebras	k_{A}, k_{B}, k_{R}, in
Contexts	Γ, Δ
Categories	$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{E}$, Rel, Set, $[G, \mathsf{Set}]$
Elements of a Group	π, σ, τ, e, g, h
Functors	F, F_0, F_1, p, q, T, T_0, T_1
G-Sets	$(A, \cdot_A)$, $(B, \cdot_B)$, A, B, ϕ, ψ
Groups	G, H
Morphisms	f, g, h, $fold$, id, u
Natural Transformations	η, t_0, t_1
n-Tuples	$\mathbf{A}$, $\mathbf{B}$, $\mathbf{R}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{r}$
Objects	A, B, C, D, E, X, Y, Z,
Relations	R, R'
Sets	X, Y, A, A_i, B, C, I, J
Terms	t, u
Types	T, T_1, T_2, U, $\mathsf{Quantity}(X)$

CHAPTER 1

INTRODUCTION

Let's play a game. I give you a term $t : T$, and you can do whatever you like with the term before returning it to me provided that its type remains the same. The only other rule of this game is that you have to tell me what you will do with the mystery term *before* I give it to you. This means that you will *never know* what the term is going to be *before* you have to say what you will do with it. I win if you can't carry out your action, otherwise you win. So let's play.

I begin by asking you "what are you going to do with the term?" You scratch your head for a bit, and then reply "I'm going to divide it by two to give $(x/2) : X$". I ponder your response and then decide to give you the list $[1, 2, 3] : \mathsf{List}(\mathbb{N})$, which you cannot divide by 2, because $([1, 2, 3]/2) : \mathsf{List}(\mathbb{N})$ hasn't been assigned a meaning. You could have defined a function $\_/2 : \mathsf{List}(\mathbb{N}) \to \mathsf{List}(\mathbb{N})$ and then used that, but you would have needed to have defined this function *before* I gave you the term. Let's play again.

You think for a moment and then say "OK, this time I will add the term to itself to give $x + x : X$, and if you give me a list '+' will stand for concatenation." If I give you $[1, 2, 3] : \mathsf{List}(\mathbb{N})$, you can easily calculate $[1, 2, 3] + [1, 2, 3] = [1, 2, 3, 1, 2, 3]$ to win this round. But I realise this and so instead give you just the character $a : Char$, which you do not know how to add to itself and so I win.

"OK. One more game," you say. "I'm going to do nothing to the term, and then I will return it". I think about this for a moment and soon realise that there is no way for me to win, you can *always* do nothing. And so I give you $thesis : String$, which you return in its original condition, and you become the winner.

The trick was to come up with an action for returning the term that could be done whatever the situation. If the action relied in any way on something specific about the term to define a function in an *ad-hoc* way, for example its ability to be divided by 2 or added to itself, then a term could always be chosen that would *not* have the required property. There is only one action that can be chosen to win this game and it's the do-nothing approach.

This game in computer science corresponds to writing programs that work for all possible inputs without any knowledge of the input itself. We call such programs *parametrically polymorphic*, and programs of this type will be one of the main focuses of this thesis.

When writing a parametrically polymorphic function, one is severely restricted by the generality in which you must work. This means there is only one parametrically polymorphic function of type $\forall X.X \to X$ — the polymorphic identity function $\Lambda X.\lambda x : X.x$. In our game, this corresponds to the do-nothing approach. However, mathematically proving that the polymorphic identity function $\forall X.X \to X$ is the only parametric function of this type isn't so simple. Though by using a technique called *relational parametricity* it can be done.

In this thesis we will look to understand relational parametricity with insights from category theory by using fibrations. This will require a working knowledge of fibrations and so we will present an introduction to fibrations as well as some additional background material in Chapter 2.

In Chapter 3 we will look at relational parametricity for System F. This is the scenario in which relational parametricity was originally formulated by Reynolds [1]. When types are interpreted solely as sets without any relational constraints, the interpretation of forall types does not represent parametric polymorphism. This means that ad-hoc functions can inhabit the type $\forall X.X \to X$. Reynolds noticed that if we instead interpret types as relations and terms as relation homomorphisms, then we reach a *parametric model.*

Instead of reproducing Reynolds's original paper we will present an alternative viewpoint using fibrations. This approach has lead to a general fibrational description of relational parametricity [2] by the author of this thesis in a collaborative paper. To maintain our focus we will not describe the model in its full generality here, but instead use it to guide our exploration of parametricity. Towards the end of Chapter 3 in Corollary 42, we will prove that in this model the polymorphic identity function is the only inhabitant of the type $\forall X.X \to X$.

In Chapter 4 we will look at System F polymorphism in the context of G-sets, where G is a group. This situation is not covered by the general approach described in [2] as the category of G-sets is not well-pointed, which is a requirement of the general framework. This means that results such as there being only one function of type $\forall X.X \to X$ are no longer true, in fact there is a function for every element of G. In terms of the term-game this corresponds to introducing a new rule (which makes the game seem a lot easier). The new rule says that with every term you have a "G-action" which can be used whenever you like. This means that there is a winning strategy for every element of G, and in Theorem 78 we prove this concretely.

Finally, in Chapter 5 we will look at *dimension polymorphism*. Up until this point in the thesis, we will have concentrated on polymorphism of type variables, but dimension variables will behave differently, and so will the parametricity that we see. To account for dimension polymorphism we will work with a different type system to System F that is able to represent dimensions (e.g. length, time) and units (e.g. metres, seconds), as well as being able to define dimension polymorphic functions. For example, the term

$$(\Lambda X.\, \lambda x : \mathsf{Quantity}(X).\, x + x) :\; \forall X.\, \mathsf{Quantity}(X) \to \mathsf{Quantity}(X) \tag{1.1}$$

takes as input a dimension X, a quantity of that dimension $x : \mathsf{Quantity}(X)$, and returns its double $x + x$ of the same dimension. Extending the term-game-analogy, Chapter 5 corresponds to finding a winning strategy that is invariant under scaling. Your strategy must be appropriately scaled if I were to scale the term before handing it to you.

In our study of dimension polymorphism we will define a general categorical model for such type systems, as well as giving a general description of relational parametricity in this setting. Along the way we will spend some time looking at a model that uses group actions and prove many theorems that usually require a separate relational semantics, but in the group actions model can be proven directly. This leads to some slick and elegant proofs and we will explore whether adding parametricity here actually gives us any extra power.

We will try to avoid assuming a large quantity of knowledge by the reader, in an attempt to make this thesis as accessible as possible. As a minimum, we will expect the reader to have a passing familiarity with category theory and of type theory.

Throughout this thesis we will look at the interaction between category theory and type theory in a parametric setting, whilst also exploring the role of group actions. The main contributions of this work are as follows.

- We present Reynolds's relational parametricity in a new light using fibrations. This approach allows us to uncover some of the hidden categorical structure present in Reynolds's original definitions and results, leading to new insights in the study of parametricity.
- We show how initial algebra theorems fit into the fibrational framework.
- We give a parametric model of System F using group actions.
- We show that in the group actions model of System F a generalised version of the standard inital algebra theorem holds.
- We give a general categorical semantics for type systems with dimension polymorphism.
- We explore a semantics for dimension polymorphism using group actions, and show how this can be used to prove parametricity-like theorems.
- We give a general relational model for type systems with dimension types, and compare this to the group actions model.

The material presented in Chapter 2 is well-known and given here for completeness. The material in Chapter 3 is entirely the work of John Reynolds [1] but presented in a way that highlights its fibrational nature. This is made possible by insights given from the general fibrational approach developed by the author of this thesis and others in [2]. Chapter 4 contains material developed by the author of this thesis with support and guidance from Neil Ghani and Fredrik Nordvall Forsberg. Finally, the material in Chapter 5 is from a joint paper [3]. My specific contributions are as follows.

- I developed the categorical formulation of Reynolds's Relational Parametricity presented in Chapter 3, including the fibrational presentation of the Abstraction Theorem (Theorem 39) and the categorical proof of the Graph Lemma (Theorem 48).
- I formulated and proved all of the results in Chapter 4, except where explicitly stated. This included showing that G-sets give a parametric model of System F in Section 4.3 that satisfies the Identity Extension Lemma (Theorem 73) and the Abstraction Theorem (Theorem 75), characterising the interpretation of $\forall X.X \to X$ in the G-set model (Theorem 78), and proving a generalised initial algebra theorem (Theorem 85).
- The categorical model of dimension types in Chapter 5 was developed as part of collaborative work. I helped formulate the model, proved the majority of the results in the group actions model in Section 5.6 (Theorem 100 – 105), and developed the relational models (Example 107 and Example 108) in Section 5.7.

In only three words this thesis is about types, categories, and actions. In one sentence this thesis is about using fibrations to study polymorphism and parametricity. And in one thesis this thesis is about...

CHAPTER 2

FIBRATIONS

Throughout mathematics and computer science there are many examples of indexed collections of “stuff”. The stuff might be sets, monoids, groups, categories, types or something completely different, but these collections often share two similar properties. The first is that each collection is indexed by some mathematical object, and the second is that there exists a reindexing operation.

For example, consider a collection of sets $\{A_i\}_{i \in I}$ indexed by another set I, called the indexing set, so that for each element of i we have a set A_i. We can capture this situation with a function $\phi : A \to I$, with $A = \sqcup_{i \in I} A_i$, defined by $\phi(i, a) = i$. In other words, for each element i of I, we have a set given by the preimage $\phi^{-1} i = A_i$.

Suppose further that $f : J \to I$ is a function. By taking the pullback of ϕ along f we obtain a new function $f^*\phi : f^*A \to J$.

$$\begin{tikzcd}
f^*A \arrow[r] \arrow[d, "f^*\phi"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "\phi"] \\
J \arrow[r, "f"'] & I
\end{tikzcd}$$

Here f^*A is given by the set $\{(j,x) \mid fj = \phi x\}$, which we can be rewritten as,

$$\begin{aligned} f^*A &= \{(j,x) \mid x \in \phi^{-1}(fj)\} \\ &= \coprod_{j\in J} A_{fj} \ . \end{aligned}$$

Hence, the function $f^*\phi$ defines a reindexed collection of sets $\{A_{fj}\}_{j\in J}$, which is now indexed by the set J instead of I.

Putting this together we see that a collection of sets can be viewed as a function, where reindexing is given via pullback along a function between indexing sets. The universal property of pullbacks guarantees that reindexing gives the “best possible substitution”. By generalising this example to categories and functors we reach the notion of a *fibration* [4, 5]. We will spell out the relationship between fibrations and indexed sets in Example 12, but first we need a couple of definitions.

Definition 1 (Cartesian Morphism). Let $p : \mathcal{E} \to \mathcal{B}$ be a functor. Suppose that $f : A \to B$ is a morphism in $\mathcal{B}$ and E is an object of $\mathcal{E}$ such that $pE = B$. Then $h : X \to E$ is said to be a *Cartesian morphism above* f if the following conditions hold.

$$\begin{tikzcd} Z \arrow[d, dashed, "\exists!\tilde{u}"'] \arrow[rd, "g"] & & \mathcal{E} \arrow[ddd, "p"] \\ X \arrow[r, "h"'] & E & \\ pZ \arrow[d, "u"'] \arrow[rd, "pg"] & & \\ A \arrow[r, "f"'] & B & \mathcal{B} \end{tikzcd}$$

- $ph = f$
- For any morphism $g : Z \to E$ in $\mathcal{E}$ such that $f \circ u = pg$ for some morphism $u : pZ \to A$ in $\mathcal{B}$, there exists a unique morphism $\tilde{u} : Z \to X$ in $\mathcal{E}$ such that $h \circ \tilde{u} = g$ and $p\tilde{u} = u$.

Using Cartesian morphisms, we now define fibrations.

Definition 2 (Fibration). A functor $p : \mathcal{E} \to \mathcal{B}$ is a *fibration* if for every morphism $f : A \to B$ in $\mathcal{B}$ and object E in $\mathcal{E}$ such that $pE = B$, there exists a Cartesian morphism above f with codomain E, denoted $f^{\S}_{E} : f^*E \to E$.

We will denote $f^{\S}_{E}$ simply by $f^{\S}$ when the subscript can be inferred from context. We now introduce some standard fibrational terminology.

Definition 3 (Fibration Terminology). Let $p : \mathcal{E} \to \mathcal{B}$ be a fibration. We call $\mathcal{E}$ the *total category* and $\mathcal{B}$ the *base category.* We say that an object E in the total category is *over* an object B in the base category if $pE = B$, and similarly for morphisms. We call a morphism of the total category a *vertical morphism* if it is over an identity morphism id_B, for some object B in the base. We call the subcategory of $\mathcal{E}$ consisting of objects over B and morphisms over id_B the *fibre above* B denoted $\mathcal{E}_B$.

A functor $p : \mathcal{E} \to \mathcal{B}$ is an *opfibration* if $p^{op} : \mathcal{E}^{op} \to \mathcal{B}^{op}$ is a fibration, and is a *bifibration* if it is simultaneously a fibration and an opfibration. We do not spell out the details here but instead refer the reader to [4] for more information, and note that the examples in Section 2.1 and Section 2.2 are all bifibrations. We will use both Cartesian and opcartesian maps in Chapters 3 and 4 to define the graph of a relation, and hence the introduction of opfibrations here. In the following examples, definitions and comments we will mostly consider fibrations for ease of presentation, but analogous statements can be made about opfibrations (and hence bifibrations) as well.

It is often useful to have fibrations that are equipped with a specific choice of Cartesian morphisms, such a fibration is called *cloven* and we call the choice of Cartesian morphisms a *cleavage.* Assuming the Axiom of Choice it is always possible to construct a particular cleavage for a fibration.

Next, we introduce the reindexing functor.

Theorem 4 (Reindexing Functor). *Let $p : \mathcal{E} \to \mathcal{B}$ be a cloven fibration and $f : A \to B$ a morphism in the base $\mathcal{B}$. Then the assignment of objects E in the fibre above B to f^*E is a functor $\mathcal{E}_B \to \mathcal{E}_A$. We denote this functor f^* and call it the* reindexing functor along f.

To see that $f^* : \mathcal{E}_B \to \mathcal{E}_A$ is actually a functor, let $g : E \to E'$ be a morphism in $\mathcal{E}_B$. Then we define the morphism $f^*g : f^*E \to f^*E'$ as the unique morphism that makes the following diagram commute.

$$\begin{tikzcd}
f^*E \arrow[r, "f^\S_E"] \arrow[d, dashed, "f^*g"'] & E \arrow[d, "g"] \\
f^*E' \arrow[r, "f^\S_{E'}"'] & E'
\end{tikzcd}$$

The existence and uniqueness of f^*g is guaranteed by the universal property of the Cartesian morphism $f^\S_{E'}$.

In Chapter 5 we will require fibrations where the reindexing functor satisfies the following definition.

Definition 5 (Split Fibration). A fibration $p : \mathcal{E} \to \mathcal{B}$ is *split* if for any two morphisms $f : A \to B$, $h : B \to C$ in the base $\mathcal{B}$, and any object E in the total category $\mathcal{E}$ the following two equalities hold.

$$(\mathrm{id}_{pE})^* = \mathrm{id}_E \qquad\qquad (h \circ f)^* = f^* \circ h^*$$

A very basic theorem about fibrations is that they are closed under composition.

Theorem 6. *Suppose that $p : \mathcal{E} \to \mathcal{B}$ and $q : \mathcal{B} \to \mathcal{A}$ are both fibrations. The $q \circ p$ is also a fibration.*

Proof. Let $f : A \to A'$ be a morphism in $\mathcal{A}$, and let E be an object in $\mathcal{E}$ such that $(q \circ p)E = A'$. Then to show that $q \circ p$ is a fibration, we must show that there exists a Cartesian morphism $f^\S_E$ above f, i.e., such that $(q \circ p)f^\S_E = f$.

Firstly, notice that since $q : \mathcal{B} \to \mathcal{A}$ is a fibration and pE is an object of $\mathcal{B}$ that is above A', there exists a Cartesian morphism $f^\S_{pE} : f^*(pE) \to pE$ in $\mathcal{B}$, such that $qf^\S_{pE} = f$. But $p : \mathcal{E} \to \mathcal{B}$ is also a fibration and E is an object in $\mathcal{E}$ above pE, so there exists a Cartesian morphism $(f^\S_{pE})^\S_E : f^*E \to E$ such that $p(f^\S_{pE})^\S_E = f^\S_{pE}$. Hence, $(f^\S_{pE})^\S_E$ is a Cartesian morphism above f as required. $\square$

This theorem allows us to combine two fibrations to form another, but there are some other ways to produce fibrations. We recall two methods here, taking the discrete fibration and taking the product.

Theorem 7. *Recall that for any category $\mathcal{C}$ the* discrete category $|\mathcal{C}|$ *consists of the objects of $\mathcal{C}$ and only the identity morphisms. Then any functor $p : \mathcal{E} \to \mathcal{B}$ induces another functor $|p| : |\mathcal{E}| \to |\mathcal{B}|$ defined as the restriction of p to $|\mathcal{E}|$. Moreover, $|p|$ is trivially a fibration, which we call the* discrete fibration on p.

Further, suppose that p is a fibration, and n is a natural number. Then the functor $p^n : \mathcal{E}^n \to \mathcal{B}^n$ defined by $p^n(E_1, \ldots, E_n) = (pE_1, \ldots, pE_2)$, is also a fibration.

Proof. It is trivial to see that $|p|$ is a fibration since the only morphisms in the base and the total category are identity morphisms. Hence, the condition of being a fibration collapses to being a functor. To see that p^n is a fibration, simply compute Cartesian morphisms component-wise. □

Before we look at some examples, we make one more observation about fibrations — they form a 2-category.

Definition 8 (Fibred Functor and Fibred Natural Transformation). Suppose that $p : \mathcal{E} \to \mathcal{B}$ and $p' : \mathcal{E}' \to \mathcal{B}'$ are fibrations. Then a *fibred functor* $F : p \to p'$ is given by two functors $F_0 : \mathcal{B} \to \mathcal{B}'$ and $F_1 : \mathcal{E} \to \mathcal{E}'$ such that $p' F_1 = F_0 p$ and Cartesian morphisms are preserved, i.e., if $f^{\S}$ is a Cartesian morphism in $\mathcal{E}$ over f in $\mathcal{B}$ then $F_1 f$ is a Cartesian morphism in $\mathcal{E}'$ over $F_0 f$ in $\mathcal{B}'$. We will often denote a fibred functor F by (F_1, F_0).

If $F' : p \to p'$ is another fibred functor, then a *fibred natural transformation* $\eta : F \to F'$ is given by two natural transformations $\eta_0 : F_0 \to F_0'$ and $\eta_1 : F_1 \to F_1'$ such that $p' \eta_1 = \eta_0 p$, in other words η_1 is over η_0.

$$\begin{tikzcd}
\mathcal{E} \arrow[r, bend left, "F_1"] \arrow[r, bend right, "F_1'"'] \arrow[r, phantom, "\Downarrow \eta_1"] \arrow[d, "p'"'] & \mathcal{E}' \arrow[d, "p'"] \\
\mathcal{B} \arrow[r, bend left, "F_0"] \arrow[r, bend right, "F_0'"'] \arrow[r, phantom, "\Downarrow \eta_0"] & \mathcal{B}'
\end{tikzcd}$$

Just like the collection of all (small) categories forms a 2-category consisting of categories, functors and natural transformations, the collection of fibrations, fibred functors and fibred natural transformations forms a 2-category, which we denote Fib.

2.1 EXAMPLES

Let us now look at a few examples of fibrations. All of these fibrations are well-known and more detailed descriptions can be found in [4], [5], [6], [7], and [8], as well as in many other places.

Example 9 (Subobject Fibration). *Let $\mathcal{C}$ be a category and let $\mathsf{Sub}(\mathcal{C})$ denote the category consisting of subobjects of $\mathcal{C}$ and commuting squares. If $\mathcal{C}$ has pullbacks of monomorphisms along arbitrary morphisms, then the functor $p : \mathsf{Sub}(\mathcal{C}) \to \mathcal{C}$, which sends each subobject to its codomain $\mathcal{C}$, is a fibration called the* subobject fibration*.*

To see that p is a fibration let $f : X \to Y$ be a morphism in the base $\mathcal{C}$, and let $s : A \rightarrowtail Y$ be (a representative of) a subobject of Y. Then the pullback $f^*(s) : f^*(A) \to X$ of s along f is also a monomorphism, see [9], and so the equivalence class of $f^*(s)$ gives a subobject.

$$\begin{tikzcd}
f^*(A) \arrow[r, "s^*(f)"] \arrow[d, tail, "f^*(s)"'] & A \arrow[d, tail, "s"] \\
X \arrow[r, "f"'] & Y
\end{tikzcd}$$

Moreover, $s^*(f) : f^*(A) \to A$ is a Cartesian morphism above f by the universal property of pullbacks.

The case where $\mathcal{C}$ is equal to Set is used in Chapter 3 and Chapter 5, so we recall its specific details.

Example 10 (Subset Fibration). *The subobjects in the category Set are given by subsets, and so we call the functor $p : \mathsf{Sub}(\mathsf{Set}) \to \mathsf{Set}$, which takes a subset $A \subseteq X$ to X, the* subset fibration*.*

For a function $f : X \to Y$ in the base and a subset $B \subseteq Y$, the Cartesian morphism $f^{\S}_B : f^*B \to B$ above f is given by the restriction of f to the preimage $f^{-1}B$.

For the rest of the fibrations in this section, we will omit the proofs that they are actually fibrations and instead just describe their basic structure. For more information see the introductory texts mentioned at the start of this section.

The next example is a generalisation of Example 9. Provided that the category $\mathcal{C}$ has enough structure, then the functor $cod : \mathcal{C}^{\rightarrow} \rightarrow \mathcal{C}$ which returns the codomain of a morphism, is a fibration.

Example 11 (Codomain Fibration). *For any category $\mathcal{C}$ we denote the category of morphisms of $\mathcal{C}$ by $\mathcal{C}^{\rightarrow}$. If $\mathcal{C}$ has pullbacks then the codomain functor $cod : \mathcal{C}^{\rightarrow} \rightarrow \mathcal{C}$ is a fibration. The Cartesian morphisms of the codomain fibration are given by pullback squares, and the universal property of Cartesian morphisms is guaranteed by the universal property of pullbacks.*

The example at the beginning of this chapter (on page 6) of indexed sets can be seen as an example of the codomain fibration, by taking $\mathcal{C}$ to be Set.

Example 12 (Indexed Sets). *Consider the fibration $p : \mathsf{Set}^{\rightarrow} \rightarrow \mathsf{Set}$ defined by $p(\phi : A \rightarrow I) = I$. Then for any function $f : J \rightarrow I$ in Set and indexed set $\phi : A \rightarrow I$ in $\mathsf{Set}^{\rightarrow}$ above I, we have that $f^*(\phi) : f^*(A) \rightarrow J$ is given by taking the pullback of ϕ along f.*

Another useful fibration is the families fibration.

Example 13 (Families Fibration). *Let $\mathcal{C}$ be a category. We define category the $\mathsf{Fam}(\mathcal{C})$, which has objects given by pairs (I, f), where I is a set and f is a function $I \rightarrow |\mathcal{C}|$, and morphisms $(u, \alpha) : (I, f) \rightarrow (J, g)$ are given by a function $u : I \rightarrow J$ in Set and a family of morphisms $\alpha_i : f(i) \rightarrow g(u(i))$. Pictorially,*

$$\begin{tikzcd}
I \arrow[rr, "u"] \arrow[dr, "f"'] & \arrow[d, phantom, "\overset{\alpha}{\Longrightarrow}"] & J \arrow[dl, "g"] \\
& \mathcal{C} &
\end{tikzcd}$$

The families fibration *for $\mathcal{C}$ is the functor $p : \mathsf{Fam}(\mathcal{C}) \rightarrow \mathsf{Set}$ defined by $p(I, f) = I$. For any morphism $u : I \rightarrow J$ in Set and any pair (J, g) in $\mathsf{Fam}(\mathcal{C})$, reindexing is given by $u^*(J, g) = (I, g \circ u)$ and the Cartesian morphism above u is given by $(u, \mathrm{id}) : (I, g \circ u) \rightarrow (J, g)$*

2.2 EXAMPLES FROM PULLBACKS

We've seen that Cartesian morphisms can be pullbacks, such as in Example 9 and Example 11, but fibrations can be pullbacks too. In fact, the pullback of a fibration is again a fibration.

Theorem 14. *(Change-of-Base) Let $p : \mathcal{E} \to \mathcal{B}$ be a fibration and let $F : \mathcal{A} \to \mathcal{B}$ be a functor. Then the pullback of p along F in* Cat*, denoted $F^*(p) : F^*(\mathcal{E}) \to \mathcal{A}$ is a fibration.*

$$\begin{tikzcd} F^*(\mathcal{E}) \arrow[r, "p^*(F)"] \arrow[d, "F^*(p)"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \mathcal{E} \arrow[d, "p"] \\ \mathcal{A} \arrow[r, "F"'] & \mathcal{B} \end{tikzcd}$$

Proof. The proof is left as an exercise or alternatively a proof can be found in [4]. □

A fibration that is constructed via the pullback of another fibration is said to arise via *change-of-base*. Two of the fibrations that we will focus heavily on in this thesis both arise via change-of-base— the homogeneous relations fibration and the heterogeneous relations fibration.

As a forewarning, we mention at this point that we will use the same notation for the domain of both the homogeneous relations fibration and the heterogeneous relations fibration, but their uses will be separated by chapters. The heterogenous relations fibration appears in Chapters 3 and 4, whereas the homogeneous relations fibration appears in Chapter 5.

Example 15 (Homogeneous Relations Fibration). *Suppose that $p : \mathcal{E} \to \mathcal{B}$ is a fibration and $\_ \times \_$ is a functor $\mathcal{B} \to \mathcal{B}$ defined by $(\_ \times \_)A = A \times A$ on objects, and similarly for morphisms. Then the* homogeneous relations fibration *is given by taking the pullback of p along $\_ \times \_$.*

$$\begin{tikzcd} \mathsf{Rel}(\mathcal{E}) \arrow[r, "p^*F"] \arrow[d, "F^*p"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \mathcal{E} \arrow[d, "p"] \\ \mathcal{B} \arrow[r, "\_ \times \_"'] & \mathcal{B} \end{tikzcd}$$

The objects in the fibre above an object B are given by pairs (B, R), where B is an object of $\mathcal{B}$ and R is in the fibre $\mathcal{E}_{B \times B}$ above $B \times B$.

One of the main examples of a relations fibration uses the subset fibration $\mathsf{Sub}(\mathsf{Set}) \to \mathsf{Set}$*. In this case the objects of* $\mathsf{Rel}(\mathsf{Sub}(\mathsf{Set}))$ *are given by a pair (A, R), where R is a subset of $A \times A$, i.e., R is a relation on $A \times A$. For a morphism $f : A \to A'$ in the base and a relation $R' \subseteq A' \times A'$, reindexing is given by the relation $f^*(R') = \{(a, b) \mid (fa, fb) \in R'\}$*

on $A \times A$. Later in this thesis we will also be interested in the relations fibration as a bifibration and so we also note that for a relation $R \subseteq A \times A$ opreindexing is given by the relation $\Sigma_f = \{(fa, fb) \mid (a,b) \in R\}$ on $A' \times A'$.

We can also describe a heterogeneous version of the homogeneous relations bifibration, and again this arises via change-of-base.

Example 16 (Heterogeneous Relations Fibration). *Suppose that $p : \mathcal{E} \to \mathcal{B}$ is a fibration and $\_ \times \_$ is the product functor $\mathcal{B} \times \mathcal{B} \to \mathcal{B}$ defined by $(\_ \times \_)(A, B) = A \times B$ and similarly for morphisms. Then the* heterogeneous relations fibration *is given by change-of-base along $\_ \times \_$.*

$$\begin{tikzcd}
\mathsf{Rel}(\mathcal{E}) \arrow[r, "p^*F"] \arrow[d, "F^*p"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{B} \times \mathcal{B} \arrow[r, "\_ \times \_"'] & \mathcal{B}
\end{tikzcd}$$

Again instantiating this to the subset fibration gives the archetypal example, where objects in the relations fibration above a pair of sets (A, B) are given by heterogeneous relations $((A, B), R)$, where $R \subseteq A \times B$. For a morphism $(f, g) : (A, B) \to (A', B')$ in the base and a relation $R' \subseteq A' \times B'$, reindexing is given by the relation $(f, g)^*(R') = \{(a, b) \mid (fa, gb) \in R'\}$ on $A \times B$, and for a relation $R \subseteq A \times B$ opreindexing is given by $\Sigma_{(f,g)}R = \{(fa, gb) \mid (a, b) \in R\}$.

We will normally call the homogeneous relations fibration and the heterogeneous relations fibrations simply the *relations fibration*, only saving the words homogeneous and heterogeneous for when it is not clear from context which we mean or an emphasis is required.

2.3 FIBRATIONS WITH STRUCTURE

Throughout this thesis we will use fibrations to give the semantics of type theories and to provide descriptions of parametricity in each setting. This will require fibrations with extra structure, and so we introduce a few more basic definitions.

Definition 17 (Bicartesian Closed). Let $p : \mathcal{E} \to \mathcal{B}$ be a fibration. We say that p is *bicartesian closed* if for every object B in the base $\mathcal{B}$, the fibre $\mathcal{E}_B$ is a Cartesian closed category with finite coproducts, and each reindexing functor preserves products, coproducts and exponentials.

To be explicit, we say that each reindexing functor *preserves the bicartesian closed structure* if for any morphism $f : A \to B$ in the base $\mathcal{B}$, and any two objects E, E' in the fibre $\mathcal{E}_B$ above B, that $f^*(E \times E') \cong f^*E \times f^*E'$, $f^*(E \Rightarrow E') \cong f^*E \Rightarrow f^*E'$, and $f^*(E + E') \cong f^*E + f^*E'$, where the $\times$, $\Rightarrow$, and $+$ on the left-hand side denote the bicartesian closed structure in $\mathcal{E}_B$, and on the right-hand side the bicartesian closed structure in $\mathcal{E}_A$.

A bicartesian closed fibration with finite products in the base category can be used to give semantics to a Simply Typed λ-Calculus with coproducts and type variables. Type contexts are interpreted using the finite products in the base, and the types and terms are interpreted in the fibres above the contexts in which they are defined using the fibred bicartesian closed structure. This correspondence is well-known and more details can be found in, for example [10].

To extend the interpretation of the Simply Typed λ-Calculus to a calculus with polymorphism, we must be able to interpret the universal quantification of types. One categorical description uses simple products, which has the Beck-Chevalley condition as part of its definition.

Definition 18 (Beck-Chevalley for Products). Let $p : \mathcal{E} \to \mathcal{B}$ be a fibration with products and suppose that we have this pullback square on the right. Further, denote the right adjoints of h^* and g^* by $h^* \dashv \forall_h$ and $g^* \dashv \forall_g$. Then we say that the *Beck-Chevalley condition* is satisfied if there is a natural isomorphism $\forall_h f^* \cong j^* \forall_g$.

$$\begin{tikzcd}
A \arrow[r, "h"] \arrow[d, "f"'] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d, "j"] \\
B \arrow[r, "g"'] & D
\end{tikzcd}$$

Pictorially, we can describe the Beck-Chevalley condition for products as follows.

$$\begin{tikzcd}
\mathcal{E}_D \arrow[r, "j^*"] \arrow[d, bend right=40, "g^*"'] & \mathcal{E}_C \arrow[d, bend right=40, "h^*"'] \\
\mathcal{E}_B \arrow[r, "f^*"'] \arrow[u, bend right=40, "\forall_g"'] \arrow[u, phantom, "\vdash"] & \mathcal{E}_A \arrow[u, bend right=40, "\forall_h"'] \arrow[u, phantom, "\vdash"]
\end{tikzcd}$$

Definition 19 (Products). Let $p : \mathcal{E} \to \mathcal{B}$ be a fibration with products in the base category. For an object B in the base $\mathcal{B}$, we say that p has *B-products* if for any object X in the base $\mathcal{B}$, reindexing by the projection morphism $\pi_X : X \times B \to X$ has a right adjoint $\pi_X^* \dashv \forall_X$ satisfying the Beck-Chevalley condition. If p has B-products for all objects B in $\mathcal{B}$, then we say that p has *simple products*. And finally, we say that p has *products* if there exists a right adjoint $f^* \dashv \forall_f$ for all morphisms $f : A \to B$ in the base $\mathcal{B}$, that satisfies the Beck-Chevalley condition.

Finally, by combining a few of these definitions we reach the important notion of a $\lambda\forall$-fibration.

Definition 20 (λ∀-Fibration). Let $p : \mathcal{E} \to \mathcal{B}$ be a fibration. We say that p is a *$\lambda\forall$-fibration* if it is a bicartesian closed fibration with simple products.

All of the fibrations of interest in this thesis will be $\lambda\forall$-fibrations, as they contain all of the structure that we will need to produce categorical models. It is well-known that $\lambda\forall$-fibrations give a categorical model of the fragment of first-order logic without existential quantifiers [4].

Let's have a look at a $\lambda\forall$-fibration in action by revisiting the subset fibration.

Example 21. *The subset fibration $p : \mathsf{Sub(Set)} \to \mathsf{Set}$ is a $\lambda\forall$-fibration.*

First notice that for any set X every morphism in the fibre above X is a restriction of the identity function. Hence, for two subsets $A \subseteq X$ and $B \subseteq X$ we have that there exists a morphism $f : A \to B$ in the fibre above X if, and only if, $A \subseteq B$. With this in mind, let X be a set, then the bicartesian closed structure of $\mathsf{Sub(Set)}_X$ is given as follows.

Terminal and Initial Objects: *The terminal object is given by the subset $X \subseteq X$. Clearly any object in the fibre above X is a subset of X by definition, and all morphisms are unique. The initial object is given by the empty subset $\emptyset \subseteq X$.*

Products and Coproducts: *For two subsets $A \subseteq X$ and $B \subseteq X$ the product in the fibre $\mathsf{Sub(Set)}_X$ is given by $(A \subseteq X) \underset{\mathsf{Sub(Set)}_X}{\times} (B \subseteq X) = (A \cap B) \subseteq X$. We have projection morphisms since $A \cap B \subseteq A$ and $A \cap B \subseteq B$, and the universal property of products is* *satisfied since any set that is a subset of both A and B is also a subset of $A \cap B$. By making the dual arguments we see that the coproduct of two subsets $A \subseteq X$ and $B \subseteq X$ in the fibre $\mathsf{Sub}(\mathsf{Set})_X$ is given by $(A \subseteq X) \underset{\mathsf{Sub}(\mathsf{Set})_X}{+} (B \subseteq X) = (A \cup B) \subseteq X$.*

Exponential Objects: *For two subsets $A \subseteq X$ and $B \subseteq X$ their exponential in $\mathsf{Sub}(\mathsf{Set})_X$ is given by $(A \subseteq X) \underset{\mathsf{Sub}(\mathsf{Set})_X}{\Rightarrow} (B \subseteq X) = \big((X \setminus A) \cup B\big) \subseteq X$. To see that this is really an exponential we must show that $Hom(A \cap B, C) \cong Hom(A, (X \setminus B) \cup C)$ in the fibre above X. Or equivalently, that $A \cap B \subseteq C$ if, and only if, $A \subseteq (X \setminus B) \cup C$. To this end, first suppose that $A \cap B \subseteq C$. Then by taking a union on both sides we have that $(X \setminus B) \cup (A \cap B) \subseteq (X \setminus B) \cup C$, and clearly $A \subseteq (X \setminus B) \cup (A \cap B)$. Hence, $A \subseteq (X \setminus B) \cup C$ by transitivity. Conversely, suppose that $A \subseteq (X \setminus B) \cup C$. Then $A \cap B \subseteq \big((X \setminus B) \cup C\big) \cap B$, and clearly $\big((X \setminus B) \cup C\big) \cap B = C \cap B$ which is a subset of C. Hence by transitivity $A \cap B \subseteq C$ as required.*

Finally, since Set has finite products it only remains to show the existence of simple products.

Simple Products: *Suppose that $\pi_X : X \times Y \to X$ is a projection morphism, and that A is a subset of $X \times Y$. Then the right adjoint to reindexing along π_X, denoted $\forall_\pi$, is given by $\forall_\pi(A) = \{x \mid \forall y \in Y \;\; (x, y) \in A\}$. To see this is actually a right adjoint we refer the reader to [11].*

Though we won't use this fact its also worth noting that there exists a left adjoint $\exists_\pi$, which corresponds to existential quantification $(\exists_\pi)A = \{x \mid \exists y \in Y \;\; (x, y) \in A\}$.

The subobject fibration is such that provided the base category has enough structure then it is always a $\lambda\forall$-fibration.

Theorem 22. *Suppose that $\mathcal{B}$ is a bicartesian closed category with finite limits. Then the subobject fibration $\mathsf{Sub}(\mathcal{B}) \to \mathcal{B}$ has simple products.*

It is also worth noting that the families fibration over Set is a $\lambda\forall$-fibration.

Theorem 23. *The families fibration $p : \mathsf{Fam}(\mathsf{Set}) \to \mathsf{Set}$ is a $\lambda\forall$-fibration.*

Proofs to both Theorem 22 and Theorem 23 can be found in e.g. [4]. We will finish this section with a few theorems about fibrations with structure. First we look at the composition of two bicartesian closed fibrations, which is bicartesian closed fibration if there exists appropriate simple products.

Theorem 24. *Suppose that $q : \mathcal{A} \to \mathcal{B}$ and $p : \mathcal{B} \to \mathcal{C}$ are bicartesian closed fibrations and let $u : \mathcal{A} \to \mathcal{C}$ denote the composite $p \circ q$ (hence u is also a fibration). Then if q has simple products, u is bicartesian closed.*

$$\begin{tikzcd} \mathcal{A} \arrow[r, "q"] \arrow[dr, "u"'] & \mathcal{B} \arrow[d, "p"] \\ & \mathcal{C} \end{tikzcd}$$

Proof. Let C be an object in the base $\mathcal{C}$, and let A and A' be objects in the fibre $\mathcal{A}_C$, i.e., $uA = C$ and $uA' = C$, then the bicartesian structure is given as follows.

Terminal and Initial Objects: To find the terminal object in $\mathcal{A}_C$, first let $1_{\mathcal{B}_C}$ denote the terminal object in the fibre $\mathcal{B}_C$, which exists since p is a bicartesian closed fibration. Then the terminal object in $\mathcal{A}_C$ is given by the terminal object in the fibre above $1_{\mathcal{B}_C}$, which exists since q is bicartesian closed. The initial object is similarly given by the initial object in the fibre above $\emptyset_{\mathcal{B}_C}$.

Products and Coproducts: Since p is a bicartesian closed fibration, we have the fibred product $qA \times qA'$, and the projection morphisms $\pi_1 : qA \times qA' \to qA$ and $\pi_2 : qA \times qA' \to qA'$, in the fibre $\mathcal{B}_C$. The fibred product of A and A' is given by $\pi_1^* A \times \pi_2^* A'$ in the fibre $\mathcal{A}_{qA \times qA'}$, which exists since q is a bicartesian closed fibration. Similarly, the fibred coproduct of A and A' is given by $\iota_1^* A + \iota_2^* A'$, where ι_1 and ι_2 denote the injection morphisms $\iota_1 : qA \to qA + qA'$ and $\iota_2 : qA' \to qA + qA'$ respectively.

Exponential Objects: Since p is a bicartesian closed fibration, we have the fibred exponential $qA \Rightarrow qA'$ in $\mathcal{B}_C$, the evaluation morphism $\mathsf{ev} : (qA \Rightarrow qA') \times qA \to qA'$, and the projection morphism $\pi_2 : (qA \Rightarrow qA') \times qA \to qA$. Note also that since q has simple products, reindexing by the projection $\pi_1 : (qA \Rightarrow qA') \times qA \to (qA \Rightarrow qA')$ has a right adjoint $\pi_1^* \dashv \forall_{\pi_1}$. Putting this all together, the fibred exponential $A \Rightarrow A'$, is given by $\forall_{\pi_1}(\pi_2^* A \Rightarrow \mathsf{ev}^* A')$, which exists since q is a bicartesian closed fibration with simple products. $\square$

$\lambda\forall$-fibrations are not closed under composition in general either, we need a stronger condition on one of the fibrations to ensure the existence of simple products for a composition.

Theorem 25. *Suppose that $q : \mathcal{A} \to \mathcal{B}$ and $p : \mathcal{B} \to \mathcal{C}$ are fibrations, and let $u : \mathcal{A} \to \mathcal{C}$ denote the composite $u = p \circ q$, as in Theorem 24. Suppose further that p has simple products. Then u has simple products that are preserved by q if and only if for any projection morphism $\pi : p(B) \times Y \to p(B)$ in $\mathcal{C}$, the reindexing functor $(\pi^{\S}_{p(B)})^* : \mathcal{A}_B \to \mathcal{A}_{\pi^*(B)}$ has right adjoints for all $B \in \mathcal{B}$, satisfying the Beck-Chevalley condition.*

Proof. This theorem is proven by using the factorisation and lifting properties of the 2-category Fib as outlined by Hermida in [12]. □

Unlike composition, change-of-base of fibrations with structure is easy. Both bicartesian closed fibrations, and fibrations with simple products are closed under change-of-base along a product preserving functor.

Theorem 26. *Suppose that $p : \mathcal{E} \to \mathcal{B}$ is a $\lambda\forall$-fibration, $\mathcal{A}$ is a category with finite products, and $F : \mathcal{A} \to \mathcal{B}$ is a product preserving functor. Then the pullback of p along F, denoted $F^*(p) : F^*(\mathcal{E}) \to \mathcal{A}$, is a $\lambda\forall$-fibration.*

$$\begin{tikzcd}
F^*(\mathcal{E}) \arrow[r, "p^*(F)"] \arrow[d, "F^*(p)"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{A} \arrow[r, "F"'] & \mathcal{B}
\end{tikzcd}$$

Proof. Recall that $F^*(\mathcal{E})_A \cong \mathcal{E}_{FA}$. Since each fibre in $F^*(\mathcal{E})$ is a fibre in $\mathcal{E}$ it is bicartesian closed and the structure is preserved by reindexing. By definition $\mathcal{A}$ has finite products and so all that remains to show is the existence of simple products.

Let $\pi_A : A \times B \to A$ be a projection morphism in $\mathcal{A}$. Then the right-adjoint to reindexing along π_A, denoted $\forall_A : (F^*\mathcal{E})_{A\times B} \to (F^*\mathcal{E})_A$, is given by $\forall_A(A \times B, E) = (A, \forall_{F(A)} E)$, where $\forall_{F(A)} : \mathcal{E}_{F(A)\times F(B)} \to \mathcal{E}_{F(A)}$ is the right adjoint to reindexing along $F(\pi_A) : F(A) \times F(B) \to F(A)$, which exists because p has simple products. Notice that we used product preservation to ensure that $F(\pi_A)$ is also a projection morphism. □

An interesting class of fibrations with simple products has functors as objects in the total category. Let $\mathcal{S}$ be a category (typically $\mathcal{S} = $ Set), and consider the category Cat$//\mathcal{S}$. The objects of $\mathsf{Cat}//\mathcal{S}$ are pairs $(\mathcal{C}, P : \mathcal{C} \to \mathcal{S})$, where $\mathcal{C}$ is a small category and $P : \mathcal{C} \to \mathcal{S}$ a functor. Morphisms $(F, \phi) : (\mathcal{C}, P) \to (\mathcal{D}, Q)$ are pairs of a functor $F : \mathcal{C} \to \mathcal{D}$ and a natural transformation $\phi : P \to Q \circ F$. The projection functor $(\mathcal{C}, P) \mapsto \mathcal{C}$ is a fibration $\mathsf{Cat}//\mathcal{S} \to \mathsf{Cat}$. The fibre over a small category $\mathcal{C}$ is the category $[\mathcal{C}, \mathcal{S}]$ of functors $\mathcal{C} \to \mathcal{S}$ and natural transformations between them. Reindexing is given by precomposition of functors.

Theorem 27. *If $\mathcal{S}$ has all small limits then the fibration $\mathsf{Cat}//\mathcal{S} \to \mathsf{Cat}$ has simple products.*

This result appears in the literature (see e.g. Lawvere [13, end of §3], Melliès and Zeilberger [14]), but its construction is important in Chapter 5 so we will sketch it here.

Proof Sketch. For any functor $F : \mathcal{C} \to \mathcal{D}$, the reindexing functor $F^* : \mathcal{S}^{\mathcal{D}} \to \mathcal{S}^{\mathcal{C}}$ has a right adjoint $F_* : \mathcal{S}^{\mathcal{C}} \to \mathcal{S}^{\mathcal{D}}$ given by the right Kan extension along F ,[1] which exists when $\mathcal{S}$ has limits. For simple products, we are only interested in a right adjoint to weakening, i.e. in the functor $\forall_{\mathcal{C}} : \mathcal{S}^{\mathcal{C} \times \mathcal{D}} \to \mathcal{S}^{\mathcal{C}}$ which is the right Kan extension along the projection functor $\pi_{\mathcal{C}} : \mathcal{C} \times \mathcal{D} \to \mathcal{C}$. Expanding the definitions, we see that $\forall_{\mathcal{C}}(P) : \mathcal{C} \to \mathcal{S}$ is a point-wise limit:

$$(\forall_{\mathcal{C}} P)(c) = \lim_{d \in \mathcal{D}} P(c, d) \quad . \tag{2.1}$$

The Beck-Chevalley condition requires that the canonical map $F^* \forall_{\mathcal{C}'} \to \forall_{\mathcal{C}} (F \times \mathrm{id}_{\mathcal{D}})^*$ is a natural isomorphism for all functors $F : \mathcal{C} \to \mathcal{C}'$, which is satisfied since for any functor $P : \mathcal{C} \times \mathcal{D} \to \mathcal{S}$ and any object $c \in \mathcal{C}$,

$$\begin{aligned}(F^*(\forall_{\mathcal{C}'} P))(c) = (\forall_{\mathcal{C}} P)(F(c)) \cong \lim_{d \in \mathcal{D}} P(F(c), d) = \lim_{d \in \mathcal{D}} (((F \times \mathrm{id}_{\mathcal{D}})^*(P))(c, d)) \\ \cong (\forall_{\mathcal{C}} ((F \times \mathrm{id}_{\mathcal{D}})^*(P))(c) \text{ as required.}\end{aligned}$$

□

[1]For a refresher on Kan extensions see [9].

2.4 GROUP ACTIONS

To finish this chapter we look will recall some basic properties about group actions, as well as introducing one more fibration in Section 2.5. This section will provide the prerequisite knowledge of group actions that we will require in Chapter 4 and Chapter 5.

Definition 28 (Group Action). Let A be a set, and G be a group. A *group action* on A is given by a function $\cdot_A : G \times A \to A$ such that the following two conditions hold for every element a in the set A, and every pair of elements σ and π in the group G.

$$e \cdot_A a = a \qquad \text{(Identity)}$$

$$\sigma \cdot_A (\pi \cdot_A a) = (\sigma\pi) \cdot_A a \qquad \text{(Compatibility)}$$

Note that in the compatibility equation $\sigma\pi$ denotes the multiplication in the group G of π by σ. We call a set A equipped with a group action $\cdot_A : G \times A \to A$ a G-*set*, and denote it by $(A, \cdot_A)$, or simply A. We say that a function $f : A \to B$ between G-sets $(A, \cdot_A)$ and $(B, \cdot_B)$ is an *equivariant map* if the group action is preserved, i.e., $f(\pi \cdot_A a) = \pi \cdot_B (fb)$ for any elements a in A, and π in G.

Theorem 29. *Let G be a group. The collection of G-sets and equivariant maps between them forms a category.*

An alternative view of G-sets and equivariant maps can be given as follows. Given a group G, we write $\mathcal{G}$ (with a different font) for the corresponding one-element category, which has morphisms given by elements of G and composition given by group multiplication. Then a G-set is simply a functor $\phi : \mathcal{G} \to \mathsf{Set}$ and an equivariant map is a natural transformation. The collection of functors $\phi : \mathcal{G} \to \mathsf{Set}$ and natural transformations between them forms the functor category $[G, \mathsf{Set}]$, which is a presheaf category and hence comes equipped with structure such as Cartesian closure. This category is isomorphic to the category of G-sets above.

In this thesis we will jump between these two views of G-sets. We will denote $\phi(\star)$ by $|\phi|$ for the underlying carrier set of a G-set $\phi : \mathcal{G} \to \mathsf{Set}$, and we will use $[G, \mathsf{Set}]$ to denote both the category of G-sets viewed as functors as well as the category of G-sets viewed as sets with group actions.

Since we will make use of the bicartesian closed structure of $[G, \mathsf{Set}]$, we spell it out here.

Terminal and Initial Object: The terminal object in $[G, \mathsf{Set}]$ is given by the one object set 1 and the trivial action $\pi \cdot_1 * = *$, where $*$ denotes the canonical element of 1. The initial object in $[G, \mathsf{Set}]$ is given by the empty set $\emptyset$.

Products and Coproducts: Let $(A, \ \cdot_A)$ and $(B, \ \cdot_B)$ be G-sets, then their product $(A \times B, \ \cdot_{A\times B})$ is given by taking the product in Set and assigning the point-wise group action, i.e., for any π in G and (a, b) in $A \times B$, the action is given by $\pi \cdot_{A\times B} (a, b) = (\pi \cdot_A a, \pi \cdot_B b)$. For any two G-sets $(A_1, \ \cdot_{A_1})$ and $(A_2, \ \cdot_{A_2})$ the coproduct $(A_1 + A_2, \ \cdot_{A_1+A_2})$ is given by the disjoint union $A_1 + A_2 = \{(i, a) \mid i \in \{1, 2\} \textit{ and } a \in A_i\}$ with the action $\pi \cdot_{A_1+A_2} (i, a) = (i, \pi \cdot_{A_i} a)$.

Set-Indexed Products: Suppose that $\{(A_i, \ \cdot_{A_i})\}_{i\in I}$ is an I-indexed collection of G-sets, for some set I. Then the product $\prod_{i\in I}(A_i, \ \cdot_{A_i})$ is given by taking the product in Set and the action is given point-wise, i.e., for for any π in G and f in $\prod_{i\in I} A_i$ the action is given by $\pi \cdot_{\prod} f = \lambda i.(\pi \cdot_{A_i} f_i)$.

Exponential Objects: Let $(A, \ \cdot_A)$ and $(B, \ \cdot_B)$ be G-sets, then the exponential is given by the set $A \Rightarrow B = \{f : A \to B\}$ and the action $\pi \cdot_{\Rightarrow} f = \lambda a.\ \pi \cdot_B f(\pi^{-1} \cdot_A a)$. Notice that the elements of $A \Rightarrow B$ are not equivariant maps, but are merely functions i.e., morphisms in Set. Every equivariant map is an element of the exponential, but not every element in the exponential is equivariant.

Though the category of G-sets has lots of nice properties, one that it is missing is well-pointedness. Recall the following definition.

Definition 30 (Well-Pointed Category). Let $\mathcal{C}$ be a category with a terminal object 1. We say that $\mathcal{C}$ is *well-pointed* if for any pair of *non-equal* morphisms $f, g : X \to Y$ in $\mathcal{C}$, there exists a morphism $u : 1 \to X$ such that $f \circ u \neq g \circ u$.

To see that the category of G-sets is *not* well-pointed, notice that by equivariance any morphism $u : 1 \to A$ satisfies $\pi \cdot_A u(*) = u(*)$ for all π in G. This means that there must be an element a of A such that $\pi \cdot_A a = a$ for all π. In many situations this does not occur and there are no morphisms $1 \to A$. For example, consider the integers under addition.

2.5 THE GRP-SET FIBRATION

By collecting all of the categories of G-sets together we can define the category $\mathsf{Grp}/\!/\mathsf{Set}$.

Definition 31 (Grp-Set Category). The category $\mathsf{Grp}/\!/\mathsf{Set}$ has as objects pairs (G, A) where G is a group and A is a G-set. A morphism $(G, A) \to (H, B)$ in $\mathsf{Grp}/\!/\mathsf{Set}$ is given by a group homomorphism $\phi : G \to H$ and a function $f : A \to B$ such that for any $\pi \in G$ and $a \in A$ we have $f(\pi \cdot_A a) = (\phi\pi) \cdot_B (fa)$.

Definition 32 (Grp-Set Fibration). Let Grp be the category of groups and homomorphisms. Then there is a projection functor $\mathsf{Grp}/\!/\mathsf{Set} \to \mathsf{Grp}$ defied by $(G, A) \mapsto G$. This functor is $\lambda\forall$-fibration, which we call the $\mathsf{Grp}/\!/\mathsf{Set}$ fibration.

Theorem 33. *The* $\mathsf{Grp}/\!/\mathsf{Set}$ *fibration is a* $\lambda\forall$*-fibration.*

Proof. For any group G, the fibre above G is the category $[G, \mathsf{Set}]$ and hence each fibre is bicartesian closed with the structure outlined on Page 22. Reindexing preserves the bicartesian closed structure by definition. There is a product-preserving, full and faithful functor $\mathsf{Grp} \to \mathsf{Cat}$, taking a group to the corresponding one-object category, and the fibration $\mathsf{Grp}/\!/\mathsf{Set} \to \mathsf{Grp}$ is given by the pullback of the fibration $\mathsf{Cat}/\!/\mathsf{Set} \to \mathsf{Cat}$ along this embedding $\mathsf{Grp} \to \mathsf{Cat}$. Therefore, by Theorem 26 and Theorem 27, $\mathsf{Grp}/\!/\mathsf{Set} \to \mathsf{Grp}$ has simple products. □

Now that we have introduced and studied a good collection of fibrations, we have the categorical knowledge required for the rest of this thesis.

CHAPTER 3

PARAMETRICITY

This chapter looks at relational parametricity for System F using some basic fibrational category theory. We will take Reynolds's relational semantics as originally stated in [1] and express it in terms of fibrations. This will allow us to make use of general categorical properties to help understand the underlying structure. Reynolds's constructions fit perfectly within the relations fibration and so do the main results of parametricity, namely *The Abstraction Theorem* and *The Identity Extension Lemma*.

One general formulation of Reynolds's relational parametricity has been developed by this author and his collaborators in [2]. In that paper, we provided a bifibrational framework for parametric models of System F, and then showed that we could derive expected properties such as the existence of initial algebras and final coalgebras. The results in that paper mean that the mathematics in this chapter is simply a concrete example of a bigger and more general picture. However, to introduce all of the machinery required to tell that story would detract from the main narrative of this thesis. Instead, we will focus solely on the concrete example of the relations fibration, which will provide the necessary insights for treading the unfamiliar path of parametricity with G-sets in Chapter 4. We begin by recalling System F.

3.1 SYSTEM F

In this section we introduce the syntax of System F. Everything that we introduce in this section is standard and well-known, and so we will not dwell on the details, instead we refer the reader to Pierce's Types and Programming Languages for a comprehensive introduction [15, Chapter 18].

System F was developed independently by Girard in 1972 [16] and Reynolds in 1974 [17]. It has the same basic constructions as the Simply Typed λ-Calculus, but with the addition of polymorphism, which is introduced using the binder "$\forall$". The type $\forall X.T$, where T is also a type which may refer to X, contains terms that can be instantiated at types, which we call *polymorphic terms.*

One example is the identity function. In the Simply Typed λ-Calculus there is no way to uniformly define this function for *all* types, instead we have to define identity functions for *each* type. In System F however, we can write $\Lambda X.\lambda x : X.x : \forall X.X \to X$ to denote the *polymorphic identity function.* The symbol X represents a type variable, which we can instantiate at a type to obtain the identity function for that type, i.e.,

$$(\Lambda X \lambda x : X.x : \forall X.X \to X)[T] = \lambda x : T.x : T \to T,$$

where T is a type. We use square brackets to emphasise that we are performing *type application* and not *term application.*

Formally, the syntax of System F is given as follows.

Type Contexts and Types: A *type context* Δ is a finite list of distinct type variables $X_1, \ldots, X_n$. A well-formed *type* is given by a judgement of the form $\Delta \vdash T\ \mathsf{Type}$, where Δ is a type context. The type judgements are generated by the following rules.

SYSTEM F TYPES

$$X_1, \ldots, X_n \vdash X_i \text{ Type} \quad for\ 1 \leq i \leq n$$ **TYPE VARIABLES**

$$\frac{\Delta \vdash T_1 \ \mathsf{Type} \qquad \Delta \vdash T_2 \ \mathsf{Type}}{\Delta \vdash T_1 \to T_2 \ \mathsf{Type}}$$ **ARROW TYPES**

$$\frac{\Delta, X \vdash T \ \mathsf{Type}}{\Delta \vdash \forall X.T \ \mathsf{Type}}$$ **FORALL TYPES**

Base types or other type constants $\Delta \vdash C$ Type can be added to the language if desired. We consider two types equivalent if they are α-convertible, i.e. if they are equal after a renaming of bound variables (in a capture-avoiding way).

Term Contexts and Terms: Well-formed *term contexts* are given by judgements $\Delta \vdash \Gamma$, where Δ is a type context, Γ is of the form $x_1 : T_1, \ldots, x_n : T_n$, and there is a well-formed typing judgement $\Delta \vdash T_i$ Type for every i. Well-formed *terms* are given by judgements $\Delta; \Gamma \vdash t : T$, where $\Delta \vdash \Gamma$ is a well-formed term context and $\Delta \vdash T$ Type is a well-formed type. The term judgements are generated by the rules below.

SYSTEM F TERMS

$$\Delta; x_1 : T_1, \ldots, x_n : T_n \vdash x_i : T_i \quad for\ 1 \leq i \leq n$$ **VARIABLES**

$$\frac{\Delta; \Gamma, x : T_1 \vdash t : T_2}{\Delta; \Gamma \vdash \lambda x.t : T_1 \to T_2}$$ **ABSTRACTION**

$$\frac{\Delta; \Gamma \vdash f : T_1 \to T_2 \qquad \Delta; \Gamma \vdash t : T_1}{\Delta; \Gamma \vdash ft : T_2}$$ **APPLICATION**

$$\frac{\Delta, X; \Gamma \vdash t : T}{\Delta; \Gamma \vdash \Lambda X.t : \forall X.T} \quad x \notin \Gamma$$ **TYPE ABSTRACTION**

$$\frac{\Delta; \Gamma \vdash t : \forall X.T \qquad \Delta; \Gamma \vdash U}{\Delta; \Gamma \vdash t[U] : T[U/X]}$$ **TYPE APPLICATION**

Type abstraction requires that X does not appear in Γ. Capture-free substitution of the type U for the free occurrences of X in the type T is denoted $T[U/X]$. Term constants $\Delta;\Gamma \vdash c : C$ can be added if desired. We have standard notions of α and β-conversion, and additionally we have η and ξ-rules so that we can derive extensionality for functions and type abstractions.

SYSTEM F CONVERSION RULES

$$\frac{}{\Gamma;\Delta \vdash \lambda x.\, t = \lambda y.\, t[y/x] : T_1 \to T_2}\,(\alpha_\lambda) \qquad \frac{}{\Gamma;\Delta \vdash \Lambda X.\, t = \Lambda Y.\, t[Y/X] : \forall X.T}\,(\alpha_\Lambda)$$

$$\frac{}{\Gamma;\Delta \vdash (\lambda x.\, t)\, s = t[s/x] : T_2}\,(\beta_\lambda) \qquad \frac{}{\Gamma;\Delta \vdash (\Lambda X.\, t)[A] = t : T[A/X]}\,(\beta_\Lambda)$$

$$\frac{x \notin FV(t)}{\Gamma;\Delta \vdash t = \lambda x.\, t\, x : T_1 \to T_2}\,(\eta_\lambda) \qquad \frac{X \notin FTV(t)}{\Gamma;\Delta \vdash t = \Lambda X.\, t\, X : \forall X.T}\,(\eta_\Lambda)$$

$$\frac{\Gamma;\Delta, \vdash t_1 = t_2 : T_1 \to T_2 \qquad \Gamma;\Delta, \vdash s_1 = s_2 : T_1}{\Gamma;\Delta \vdash t_1\, s_1 = t_2\, s_2 : T_2}\,\mathrm{cong}_\lambda$$

$$\frac{\Gamma;\Delta, \vdash t_1 = t_2 : \forall X.T}{\Gamma;\Delta \vdash t_1\, A = t_2\, A : T[A/X]}\,\mathrm{cong}_\Lambda$$

$$\frac{\Gamma;\Delta, x : T_1 \vdash t_1 = t_2 : T_2}{\Gamma;\Delta \vdash \lambda x.\, t_1 = \lambda x.\, t_2 : T_1 \to T_2}\,(\xi_\lambda) \qquad \frac{\Gamma, X;\Delta \vdash t_1 = t_2 : T}{\Gamma;\Delta \vdash \Lambda X.\, t_1 = \Lambda X.\, t_2 : \forall X.T}\,(\xi_\Lambda)$$

$$\frac{}{\Gamma;\Delta \vdash t = t : T}\,(\mathrm{refl}) \qquad \frac{\Gamma;\Delta \vdash t = s : T}{\Gamma;\Delta \vdash s = t : T}\,(\mathrm{sym})$$

To finish this section we give the typing derivation of the polymorphic identity type.

Example 34. *The polymorphic identity function is typed using the variable, λ-abstraction and type abstraction rules.*

$$\frac{\dfrac{X; x : X \vdash x : X}{X \vdash \lambda x.x : X \to X}}{\vdash \Lambda X.\lambda x.x : \forall X.X \to X}$$

3.2 SEMANTICS

For the semantics of System F we want to capture a uniform notion of polymorphism (forall types). As first described by Strachey [18], there are two type of polymorphism — *parametric* polymorphism and *ad-hoc* polymorphism. Intuitively, a parametrically polymorphic program should act the same way for all data types, and an ad-hoc one does not have this constraint.

For example, the constant function $K : \forall X.\forall Y.X \to Y \to X$, which is defined by $K[X][Y](x)(y) = x$, can be given with no specific knowledge of X and Y. The program never needs to inspect the data types — it does not matter if X or Y are integers, floats, colours or letters, the program still acts in the same way.

In contrast, the addition function $+$ may be definable for all data types in your language, but its definition (usually) depends on the data that is being "added". The expression $x + y$ for integers can be defined to mean "numerically add x to y", but for strings may mean "append y on to the the end of x". By defining a function $+ : \forall X.X \to X \to X$ that adds integers numerically, appends strings, and is constant elsewhere (i.e. $\Lambda.\lambda x.\lambda y$), we would have a polymorphic function $\forall X.X \to X \to X$. However, this function does not act uniformly for all data types, and so defines an ad-hoc polymorphic program, not a parametric one.

Intuitively, parametric polymorphism is easy to understand, but not so easy to semantically characterise. Reynolds's crucial insight was to notice that parametrically polymorphic programs preserve relations, and hence relational parametricity was born [1]. By requiring that the semantics of forall types satisfies a uniformity condition expressed using relations, Reynolds was able to capture semantic properties about parametrically polymorphic functions.

In Section 3.2.1 and Section 3.2.2 we give a relational semantics to System F using the heterogeneous relations fibration $\mathsf{Rel}(p) : \mathsf{Rel} \to \mathsf{Set} \times \mathsf{Set}$. The definitions that we provide are analogous to those given by Reynolds but have a fibrational flavour. This means that they generalise beyond sets and relations to fibrations with sufficient structure [2].

It's worth noting that after Reynolds's original paper he published the result that *Polymorphism is Not Set Theoretic* [19]. This is due to a contradiction that arises similar in nature to Cantor's Paradox. But just like with Cantor's paradox it is possible avoid paradoxical issues by using the (intuitionistic) internal language of [20] or using the Calculus of Constructions [21] with impredicative Set. We leave that choice to the reader.

3.2.1 THE SEMANTICS OF TYPES

This subsection introduces the semantics of types. This will require the equality relation, so we recall it now.

Definition 35 (Equality Relation on Sets). Let A be a set. We define the *equality relation on* A as the relation given by $\mathsf{Eq}\, A = \{(a, a) \mid a \in A\} \subseteq A \times A$.

The semantics of types is given as follows. Let $\Delta = X_1, \ldots, X_n$ be a type context. Further, let $\mathbf{A}$ and $\mathbf{B}$ denote n-tuples of sets $(A_1, ..., A_n)$ and $(B_1, ..., B_n)$ in $|\mathsf{Set}|^n$ and let $\mathbf{R}$ denote an n-tuple of relations $(R_1, ..., R_n)$ in $|\mathsf{Rel}|^n$ such that $R_i \subseteq A_i \times B_i$ for $i \in \{1, ..., n\}$. We denote this relationship by $\mathbf{R} \subseteq \mathbf{A} \times \mathbf{B}$ and write the collection of relations on $\mathbf{A} \times \mathbf{B}$ by $\mathsf{Rel}^n(\mathbf{A}, \mathbf{B})$.

We give the interpretation of types as fibred functors

$$(\llbracket \Delta \vdash T \ \mathsf{Type} \rrbracket_r,\ \llbracket \Delta \vdash T \ \mathsf{Type} \rrbracket_o \times \llbracket \Delta \vdash T \ \mathsf{Type} \rrbracket_o) : |\mathsf{Rel}(p)|^{|\Delta|} \to \mathsf{Rel}(p),$$

which we denote by $\llbracket \Delta \vdash T \ \mathsf{Type} \rrbracket$. We will often refer to $\llbracket \Delta \vdash T \ \mathsf{Type} \rrbracket_o$ as the *standard semantics* and $\llbracket \Delta \vdash T \ \mathsf{Type} \rrbracket_r$ as the *relational semantics*. Before we give the definition of this fibred functor, first note that since the domain of $\llbracket \Delta \vdash T \ \mathsf{Type} \rrbracket$ is a discrete category, requiring that $\llbracket \Delta \vdash T \ \mathsf{Type} \rrbracket$ is a fibred functor amounts simply to requiring that

Diagram 3.1 commutes, i.e., no preservation of Cartesian morphisms is required.

$$\begin{tikzcd}
|\mathsf{Rel}|^{|\Delta|} \arrow[rr, "{[\![\Delta \vdash T\ \mathsf{Type}]\!]_r}"] \arrow[d, "{|\mathsf{Rel}(p)|^{|\Delta|}}"'] & & \mathsf{Rel} \arrow[d, "{\mathsf{Rel}(p)}"] \\
|\mathsf{Set}|^{|\Delta|} \times |\mathsf{Set}|^{|\Delta|} \arrow[rr, "{[\![\Delta \vdash T\ \mathsf{Type}]\!]_o \times [\![\Delta \vdash T\ \mathsf{Type}]\!]_o}"'] & & \mathsf{Set} \times \mathsf{Set}
\end{tikzcd} \tag{3.1}$$

The use of discrete categories is reflected in Reynolds's original approach [1], since he does not give a functorial action of types on morphisms. Parametricity treats the action on morphisms via graph relations, which we will see in Section 3.3.

Returning to the definition, the fibred functor $[\![\Delta \vdash T\ \mathsf{Type}]\!] : |\mathsf{Rel}(p)|^{|\Delta|} \to \mathsf{Rel}(p)$ is given inductively as follows.

Type Variables: The type variables are interpreted as projections, so that we have $[\![\Delta \vdash X_i\ \mathsf{Type}]\!]_o \mathbf{A} = A_i$ and $[\![\Delta \vdash X_i\ \mathsf{Type}]\!]_r \mathbf{R} = R_i$.

Arrow Types: Both the standard semantics and the relational semantics of arrow types are given by exponential objects, i.e., $[\![\Delta \vdash T \to U]\!]_o \mathbf{A} = [\![\Delta \vdash T]\!]_o \mathbf{A} \Rightarrow [\![\Delta \vdash U]\!]_o \mathbf{A}$, and $[\![\Delta \vdash T \to U]\!]_r \mathbf{R} = [\![\Delta \vdash T]\!]_r \mathbf{R} \Rightarrow [\![\Delta \vdash U]\!]_r \mathbf{R}$. Recall that the exponential $[\![\Delta \vdash T]\!]_r \mathbf{R} \Rightarrow [\![\Delta \vdash U]\!]_r \mathbf{R}$ is given by $\{(f,g) \mid \forall (a,b) \in [\![\Delta \vdash T]\!]_r \mathbf{R},\ (fa, gb) \in [\![\Delta \vdash U]\!]_r \mathbf{R}\}$. Hence, we see a fundamental idea of Reynolds's relational parametricity, that related inputs map to related outputs.

Forall Types: Inhabitants of forall types must be parametrically polymorphic, and so we see that a uniformity condition is imposed. For a function f in $\prod_{S \in Set} FS$ where FS is some expression, we denote the application of f to a set A by f_A. The standard semantics $[\![\Delta \vdash \forall X.T]\!]_o \mathbf{A}$ is given by

$$\{f : \prod_{S \in \mathsf{Set}} [\![\Delta, X \vdash T]\!]_o (\mathbf{A}, S) \mid \forall R \subseteq A \times B,\ (f_A, f_B) \in [\![\Delta \vdash T]\!]_r (\mathsf{Eq}^n \mathbf{A}, R)\},$$

and the relational semantics $[\![\Delta \vdash \forall X.T]\!]_r \mathbf{R} \subseteq [\![\Delta \vdash \forall X.T]\!]_o \mathbf{A} \times [\![\Delta \vdash \forall X.T]\!]_o \mathbf{B}$ is defined by

$$[\![\Delta \vdash \forall X.T]\!]_r \mathbf{R} = \{(f,g) \mid \forall R \subseteq A \times B,\ (f_A, f_B) \in [\![\Delta \vdash T]\!]_r (\mathbf{R}, R)\}.$$

In other words, two parametric functions are related if they map related inputs to related outputs. The standard and relational semantics rely crucially on each other, and must be defined simultaneously. The definition of $[\![\Delta \vdash \forall X.T]\!]_o\mathbf{A}$ contains $[\![\Delta \vdash T]\!]_r(\mathsf{Eq}^n\mathbf{A}, R)$ and the definition of $[\![\Delta \vdash \forall X.T]\!]_r$ is a relation on $[\![\Delta \vdash \forall X.T]\!]_o\mathbf{A} \times [\![\Delta \vdash \forall X.T]\!]_o\mathbf{B}$. This means that instead of providing one set-based semantics and one relational semantics, we really are defining a single semantics based on the relations fibration $\mathsf{Rel} \to \mathsf{Set} \times \mathsf{Set}$.

Theorem 36. *The interpretation of any type judgement* $\Delta \vdash T$ Type *defines a fibred functor* $[\![\Delta \vdash T\ \mathsf{Type}]\!] : |\mathsf{Rel}(p)|^{|\Delta|} \to \mathsf{Rel}(p)$, *in other words Diagram 3.1 commutes.*

Proof. We prove this theorem by structural induction on the type judgements. Each case requires showing that for any relation $\mathbf{R} \subseteq \mathbf{A} \times \mathbf{B}$, we have that $[\![\Delta \vdash T\ \mathsf{Type}]\!]_r\mathbf{R}$ is a relation on $[\![\Delta \vdash T\ \mathsf{Type}]\!]_o\mathbf{A} \times [\![\Delta \vdash T\ \mathsf{Type}]\!]_o\mathbf{B}$, which is true by construction. □

Types do not just define fibred functors, they in fact define *equality preserving* fibred functors. We add an equality subscript $[\![\Delta \vdash T\ \mathsf{Type}]\!] : |\mathsf{Rel}(p)|^{|\Delta|} \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)$ to emphasise that the type $\Delta \vdash T$ Type defines an equality preserving functor between the fibrations $|\mathsf{Rel}(p)|^{|\Delta|}$ and $\mathsf{Rel}(p)$.

Theorem 37 (Identity Extension Lemma). *If* $\Delta \vdash T$ Type *then* $[\![\Delta \vdash T\ \mathsf{Type}]\!]$ *is an equality preserving functor* $|\mathsf{Rel}(p)|^{|\Delta|} \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)$, *in other words* $[\![\Delta \vdash T\ \mathsf{Type}]\!]_r \circ |\mathsf{Eq}|^{|\Delta|} = \mathsf{Eq} \circ [\![\Delta \vdash T\ \mathsf{Type}]\!]_o$.

Proof. The Identity Extension Lemma is proven by induction on the type judgement. To give a flavour of the proof we show the forall types case, i.e., that $[\![\Delta \vdash \forall X.T\ \mathsf{Type}]\!]_r \circ |\mathsf{Eq}|^{|\Delta|} = \mathsf{Eq} \circ [\![\Delta \vdash \forall X.T\ \mathsf{Type}]\!]_o$, and we leave the rest as a simple exercise.

Suppose that $\Delta \vdash \forall X.T$ Type is a type, $n = |\Delta|$, and $\mathbf{A}$ is an n-tuple of sets in $|\mathsf{Set}|^n$. To see that $\mathsf{Eq}\,([\![\Delta \vdash \forall X.T\ \mathsf{Type}]\!]_o\mathbf{A}) \subseteq [\![\Delta \vdash \forall X.T\ \mathsf{Type}]\!]_r(\mathsf{Eq}^n\mathbf{A})$, let (f, f) be an element of $\mathsf{Eq}\,([\![\Delta \vdash \forall X.T\ \mathsf{Type}]\!]_o\mathbf{A})$. Then since $f \in [\![\Delta \vdash \forall X.T\ \mathsf{Type}]\!]_o\mathbf{A}$, for any relation $R \subseteq A \times B$ we have that $(f_A, f_B) \in [\![\Delta, X \vdash T\ \mathsf{Type}]\!]_r(\mathsf{Eq}^n\mathbf{A}, R)$, and hence $(f, f) \in [\![\Delta \vdash \forall X.T\ \mathsf{Type}]\!]_r(\mathsf{Eq}^n\mathbf{A})$.

For the reverse direction suppose that (f, f') is an element of $[\![\Delta \vdash \forall X.T \text{ Type}]\!]_r \mathsf{Eq}^n \mathbf{A}$. Then by definition we have that $(f_A, f'_A) \in [\![\Delta, X \vdash T \text{ Type}]\!]_r(\mathsf{Eq}^n \mathbf{A}, \mathsf{Eq} A)$ for any set A. Hence by the induction hypothesis $(f_A, f'_A) \in \mathsf{Eq}\left([\![\Delta, X \vdash T \text{ Type}]\!]_o(\mathbf{A}, A)\right)$, or in other words $f_A = f'_A$, and since A was arbitrary $f = f'$ as required. □

To end this section we state a Substitution Lemma, which is proven by induction, since we will require it in the proof of Theorem 39.

Lemma 38 (Substitution Lemma). *Let* $\mathbf{A}$ *be an* n*-tuple of sets in* Set^n *and let* $\mathbf{R}$ *be an* n*-tuple of relations in* Rel^n*. If* $\Delta, X \vdash T$ Type *is a type and* $\Delta \vdash U$ *is a type then,*

$$[\![\Delta, X \vdash T]\!]_o(\mathbf{A}, [\![\Delta \vdash U]\!]_o \mathbf{A}) = [\![\Delta \vdash T[U/X]]\!]_o \mathbf{A},$$

and

$$[\![\Delta, X \vdash T]\!]_r(\mathbf{R}, [\![\Delta \vdash U]\!]_r \mathbf{R}) = [\![\Delta \vdash T[U/X]]\!]_r \mathbf{R}.$$

3.2.2 THE SEMANTICS OF TERMS

A term context $\Gamma = x_1 : T_1, \ldots, x_n : T_n$ is interpreted as a product of types, i.e., the standard semantics is given by $[\![\Delta \vdash \Gamma]\!]_o = [\![\Delta \vdash T_1 \text{ Type}]\!]_o \times \cdots \times [\![\Delta \vdash T_n \text{ Type}]\!]_o$ and the relational semantics is given by $[\![\Delta \vdash \Gamma]\!]_r = [\![\Delta \vdash T_1 \text{ Type}]\!]_r \times \cdots \times [\![\Delta \vdash T_n \text{ Type}]\!]_r$, which defines a fibred functor $[\![\Delta \vdash \Gamma]\!] : |\mathsf{Rel}(p)|^{|\Delta|} \to \mathsf{Rel}(p)$.

A term $\Delta; \Gamma \vdash t : T$ is interpreted as a fibred natural transformation

$$([\![\Delta; \Gamma \vdash t : T]\!]_r, [\![\Delta; \Gamma \vdash t : T]\!]_o \times [\![\Delta; \Gamma \vdash t : T]\!]_o) : [\![\Delta \vdash \Gamma]\!] \to [\![\Delta \vdash T \text{ Type}]\!],$$

which we denote by $[\![\Delta; \Gamma \vdash t : T]\!]$ and define inductively as follows.

Let $\mathbf{A}$ and $\mathbf{B}$ be two n-tuples of sets in $|\mathsf{Set}|^n$, let $\mathbf{R}$ be a relation on $\mathbf{A} \times \mathbf{B}$ in $|\mathsf{Rel}|^n$, and let $(\mathbf{a}, \mathbf{b})$ be an element of $\mathbf{R}$. In what follows below we will often omit the left-hand side of the turnstile in judgements for ease of presentation, i.e., we will write term contexts $\Delta \vdash \Gamma$ simply by Γ, types $\Delta \vdash T$ Type simply by T, and terms $\Delta; \Gamma \vdash t : T$ simply by $t : T$.

Variables: The morphism $[\![x_i : T_i]\!]_o\mathbf{A} : [\![T_1]\!]_o\mathbf{A} \times \cdots \times [\![T_n]\!]_o\mathbf{A} \to [\![T_i]\!]_o\mathbf{A}$ is given by the i^{th} projection morphism, and similarly for $[\![x_i : T_i]\!]_r\mathbf{R} : [\![T_1]\!]_r\mathbf{R} \times \cdots \times [\![T_n]\!]_r\mathbf{R} \to [\![T_i]\!]_r\mathbf{R}$.

Abstraction: Since Set is a Cartesian closed category, we have a natural isomorphism $\phi_0 : \mathsf{Set}([\![\Gamma]\!]_o\mathbf{A}\times[\![T_1]\!]_o\mathbf{A},\ [\![T_2]\!]_o\mathbf{A}) \to \mathsf{Set}([\![\Gamma]\!]_o\mathbf{A},\ [\![T_1]\!]_o\mathbf{A} \Rightarrow [\![T_2]\!]_o\mathbf{A})$, and similarly we have an isomorphism $\phi_1 : \mathsf{Rel}([\![\Gamma]\!]_r\mathbf{A} \times [\![T_1]\!]_r\mathbf{A},\ [\![T_2]\!]_r\mathbf{A}) \to \mathsf{Rel}([\![\Gamma]\!]_r\mathbf{A},\ [\![T_1]\!]_r\mathbf{A} \Rightarrow [\![T_2]\!]_r\mathbf{A})$ since Rel is Cartesian closed. Hence, we define $[\![\lambda x.t : T_1 \to T_2]\!] : [\![\Gamma]\!]_o\mathbf{A} \to ([\![T_1]\!]_o\mathbf{A} \Rightarrow [\![T_2]\!]_o\mathbf{A})$ by $\phi_0([\![\Delta;\Gamma, x : T_1 \vdash t : T_2]\!]_o\mathbf{A})$ and similarly we define the morphism $[\![\lambda x.t : T_1 \to T_2]\!]_r\mathbf{R}$ by $\phi_1([\![\Delta;\Gamma, x : T_1 \vdash t : T_2]\!]_r\mathbf{R})$.

Application: Let ev_0 denote the evaluation map $([\![T_1]\!]_o\mathbf{A} \Rightarrow [\![T_2]\!]_o\mathbf{A}) \times [\![T_1]\!]_o\mathbf{A} \to [\![T_2]\!]_o\mathbf{A}$ and let ev_1 denote the evaluation map $([\![T_1]\!]_r\mathbf{R} \Rightarrow [\![T_2]\!]_r\mathbf{R}) \times [\![T_1]\!]_r\mathbf{R} \to [\![T_2]\!]_r\mathbf{R}$. Then, we define the morphism $[\![ft : T_2]\!]_o\mathbf{A} : [\![\Gamma]\!]_o\mathbf{A} \to [\![T_2]\!]_o\mathbf{A}$ by a simple use of the evaluation map $[\![ft : T_2]\!]_o\mathbf{Aa} = \mathsf{ev}_0([\![f : T_1 \to T_2]\!]_o\mathbf{Aa},\ [\![t : T_1]\!]_o\mathbf{Aa})$ for $\mathbf{a} \in [\![\Delta \vdash \Gamma]\!]_o\mathbf{A}$. Similarly we define the relational interpretation $[\![\Delta;\Gamma \vdash ft : T_2]\!]_r\mathbf{R} : [\![\Delta \vdash \Gamma]\!]_r\mathbf{R} \to [\![\Delta \vdash T_2]\!]_r\mathbf{R}$ by $[\![\Delta;\Gamma \vdash ft : T_2]\!]_r\mathbf{Rr} = \mathsf{ev}_1([\![\Delta;\Gamma \vdash f : T_1 \to T_2]\!]_r\mathbf{Rr},\ [\![\Delta;\Gamma \vdash t : T_1]\!]_r\mathbf{Rr})$ for any $\mathbf{r} \in [\![\Delta \vdash \Gamma]\!]_r\mathbf{R}$.

Type Abstraction: First notice that from the type abstraction rule we have that $[\![\Delta, X \vdash \Gamma]\!]_o(\mathbf{A}, A) = [\![\Delta \vdash \Gamma]\!]_o\mathbf{A}$ for any set A, since X is not free in Γ. Hence, we define $[\![\Lambda X.t : \forall X.T]\!]_o\mathbf{A} : [\![\Gamma]\!]_o\mathbf{A} \to [\![\forall X.T]\!]_o\mathbf{A}$ by $([\![\Lambda X.t : \forall X.T]\!]_o\mathbf{Aa})A = \big([\![t : T]\!]_o(\mathbf{A}, A)\mathbf{a}\big)$, and the relational interpretation $[\![\Lambda X.t : \forall X.T]\!]_r\mathbf{R} : [\![\Gamma]\!]_r\mathbf{R} \to [\![\forall X.T]\!]_r\mathbf{R}$ is given by $[\![\Lambda X.t : \forall X.T]\!]_r\mathbf{Rr} = \Lambda R.\big([\![t : T]\!]_r(\mathbf{A}, A)\mathbf{a}, [\![t : T]\!]_r(\mathbf{B}, B)\mathbf{b}\big)$.

To show that the uniformity condition is satisfied in both cases requires a simple use of the induction hypothesis.

Type Application: We define $[\![t[U] : T[U/X]]\!]_o\mathbf{A} : [\![\Gamma]\!]_o\mathbf{A} \to [\![T[U/X]]\!]_o\mathbf{A}$ by $[\![t[U] : T[U/X]]\!]_o\mathbf{Aa} = ([\![t : \forall X.T]\!]_o\mathbf{Aa})([\![U]\!]_o\mathbf{Aa})$, which is an element of $[\![T]\!]_o(\mathbf{A}, [\![U]\!]_o\mathbf{Aa})$ by definition, and hence an element of $[\![T[U/X]]\!]_o\mathbf{A}$ by Lemma 38. Similarly, we define $[\![t[U] : T[U/X]]\!]_r\mathbf{R} : [\![\Gamma]\!]_r\mathbf{R} \to [\![T[U/X]]\!]_r\mathbf{R}$ by $[\![t[U] : T[U/X]]\!]_r\mathbf{Rr} = ([\![t : \forall X.T]\!]_r\mathbf{Rr})([\![U]\!]_r\mathbf{Rr})$.

In summary, we have that every term defines a fibred natural transformation, where the naturality condition is vacuous because we are working with discrete categories.

Theorem 39 (Abstraction Theorem). *Every term defines a fibred natural transformation* $(\llbracket \Delta; \Gamma \vdash t : T \rrbracket_r,\ \llbracket \Delta; \Gamma \vdash t : T \rrbracket_o \times \llbracket \Delta; \Gamma \vdash t : T \rrbracket_o) : \llbracket \Delta \vdash \Gamma \rrbracket \to \llbracket \Delta \vdash T\ \mathsf{Type} \rrbracket$*. Or pictorially*

$$\begin{tikzcd}
{|\mathsf{Rel}|^n} \arrow[r, bend left, "\Gamma_1"{name=U}] \arrow[r, bend right, "T_1"'{name=D}] \arrow[Rightarrow, from=U, to=D, "t_1"] \arrow[d, "{|\mathsf{Rel}(p)|^n}"'] & \mathsf{Set} \arrow[d, "\mathsf{Rel}(p)"] \\
{|\mathsf{Set}|^n \times |\mathsf{Set}|^n} \arrow[r, bend left, "\Gamma_0 \times \Gamma_0"{name=U2}] \arrow[r, bend right, "T_0 \times T_0"'{name=D2}] \arrow[Rightarrow, from=U2, to=D2, "t_0 \times t_0"] & \mathsf{Set} \times \mathsf{Set}
\end{tikzcd}$$

Where we denote $\llbracket \Delta \vdash \Gamma \rrbracket_o$ *by* Γ_0*,* $\llbracket \Delta \vdash T\ \mathsf{Type} \rrbracket_o$ *by* T_0*, and* $\llbracket \Delta; \Gamma \vdash t : T \rrbracket_o$ *by* t_0*, and similarly* $\llbracket \Delta \vdash \Gamma \rrbracket_r$*,* $\llbracket \Delta \vdash T\ \mathsf{Type} \rrbracket_r$*, and* $\llbracket \Delta; \Gamma \vdash t : T \rrbracket_r$ *by* Γ_1*,* T_1*, and* t_1*.*

Proof. We prove this theorem by induction on the derivation of terms $\Delta; \Gamma \vdash t : T$. Each case requires showing that for any n-tuple of relations $\mathbf{R} \subseteq \mathbf{A} \times \mathbf{B}$ the interpretation $\llbracket \Delta; \Gamma \vdash t : T \rrbracket_r \mathbf{R} : \llbracket \Delta \vdash \Gamma \rrbracket_r \mathbf{R} \to \llbracket \Delta \vdash T \rrbracket_r \mathbf{R}$ is over $\llbracket \Delta; \Gamma \vdash t : T \rrbracket_o \mathbf{A} \times \llbracket \Delta; \Gamma \vdash t : T \rrbracket_o \mathbf{B}$, which is trivial by construction. And since the domains of the fibred functors $\llbracket \Delta \vdash \Gamma \rrbracket$ and $\llbracket \Delta \vdash T \rrbracket$ are discrete, the interpretation $\llbracket \Delta; \Gamma \vdash t : T \rrbracket$ vacuously defines a fibred natural transformation $\llbracket \Delta \vdash \Gamma \rrbracket \to \llbracket \Delta \vdash T \rrbracket$. □

3.3 PARAMETRICITY, WHAT IS IT GOOD FOR?

Relational parametricity is a property about the semantics of type systems, and is a crucial technique for formal reasoning about programming languages. The list of uses of parametricity is vast and varied, but to name just a few, parametricity has been used to prove compiler correctness [22, 23], privacy guarantees [24], geometric invariance properties [25], and data type representations [26].

One of the most famous examples of parametricity is to generate "free theorems" as shown in [27]. These theorems are properties that can be shown about terms, knowing only the type and nothing more, such as the following from [27].

Theorem 40. *Suppose that π is a parametric function in $[\![\forall X.\forall Y.X \times Y \to X]\!]_o$. Then for any two sets A and B, and any two functions $f : A \to A'$ and $g : B \to B'$*

$$\pi_{A',B'} \circ (f \times g) = f \circ \pi_{A,B}.$$

Proof. First note that since $\forall X.\forall Y.X \times Y \to X$ is a closed type, the interpretation $[\![\forall X.\forall Y.X \times Y \to X]\!]$ defines a fibred functor $1 \to \mathsf{Rel}(p)$. In other words, the interpretation is given by a relation $[\![\forall X.\forall Y.X \times Y \to X]\!]_r$ on $[\![\forall X.\forall Y.X \times Y \to X]\!]_o \times [\![\forall X.\forall Y.X \times Y \to X]\!]_o$

By the semantics of forall types we have that for any two relations $R \subseteq A \times A'$ and $R' \subseteq B \times B'$, we have that $(\pi_{A,B}, \pi_{A',B'}) \in R \times R' \to R$. Let $f : A \to A'$ and $g : B \to B'$ be functions, and choose R be the graph relation $\langle f \rangle = \{(a, fa) \mid a \in A\}$ and R' to be the graph relation $\langle g \rangle = \{(b, gb) \mid b \in B\}$. Then, we have that $(\pi_{A,B}, \pi_{A',B'}) \in \langle f \rangle \times \langle g \rangle \to \langle f \rangle$. Therefore, for any $a \in A$ and $b \in B$, we have that $\big(\pi_{A,B}(a, b), \pi_{A',B'}(fa, gb)\big) \in \langle f \rangle$, or in other words that $f\big(\pi_{A,B}(a, b)\big) = \pi_{A',B'}(fa, gb)$. Hence, since a and b were arbitrary we have the required result. □

In the proof of the theorem above we used graph relations, which often turn up during proofs using parametricity, and we will use them again and again throughout this thesis. For that reason we dedicate Subsection 3.3.1 to a formal introduction of graph relations.

Another example of a free theorem involves deducing the following isomorphism. Consider the (closed) System F type $\vdash \forall X.X \to X$. We know from Example 34 that the polymorphic identity $\Lambda X.\lambda x.x$ is a term of that type, but are there any others? The intuitive answer is no. If I give you a type and a term of that type, there is only one thing that you can do without inspecting the type, and that is to return it to me (see Chapter 1). Hence, there should only be one way to define a parametrically polymorphic function of this type, and indeed our semantics proves this.

Theorem 41. *Suppose that f is a parametric function in $[\![\forall X.X \to X]\!]_o$. Then for any set A and an element $a \in A$, we have $f_A a = a$.*

Proof. By the definition of $[\![\forall X.X \to X]\!]_o$, we have that for any relation $R \subseteq A \times B$,

$$(f_A, f_B) \in R \to R. \tag{3.2}$$

Choose R to be the relation given by the graph of the function $\phi : 1 \to A$ defined by $\phi(\star) = a$, or in other words $R = \{(\star, a)\} \subseteq 1 \times A$. Then clearly the pair $(\star, a)$ is an element of R, and hence Equation 3.2 gives that $(f_1 \star, f_A a) \in R$, or in other words that $f_A a = a$, as required. □

Theorem 41 shows that there is a surjective function $1 \to [\![\forall X.X \to X]\!]_o$ defined by $\star \mapsto \Lambda A.\lambda a.a$, and moreover this function is trivially injective. Hence, we have an isomorphism.

Corollary 42. *There is an isomorphism $[\![\forall X.X \to X]\!]_o \cong 1$.*

Let's look at another example. Consider the type $\forall X.X \to (X \to X) \to X$. This contains parametric functions that take as input an element of X and a function $f : X \to X$, what could such a parametrically polymorphic function do?

Theorem 43. *Suppose that f is a parametric function in $[\![\forall X.X \to (X \to X) \to X]\!]_o$. Let A be a set, let $a \in A$, and let $g : A \to A$ be a function. Then, $f_A(a, g) = g^n a$ for some natural number n.*

Proof. The proof of this theorem is very similar to the proof of Theorem 41. Let R be the graph of the function $\phi : \mathbb{N} \to A$ defined by $\phi(n) = g^n a$. In other words $R = \{(n, y) \mid y = g^n a\} \subseteq \mathbb{N} \times A$. Then clearly $(0, a) \in R$, and the pair (succ, g) is a relation homomorphism $R \to R$, where $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$ denotes the successor function $\mathsf{succ}(n) = n + 1$. Hence, the pair $(f_{\mathbb{N}}(0, \mathsf{succ}),\ f_A(a, g))$ is related in R, and so by setting $n = f_{\mathbb{N}}(0, \mathsf{succ})$, we have $f_A(a, g) = g^n a$, as required. □

This theorem shows that we have a surjective function $\mathbb{N} \to [\![\forall X.X \to (X \to X) \to X]\!]_o$ defined by $n \mapsto \Lambda A.\lambda a.\lambda g.g^n a$. To see that this function is injective let n and m be natural numbers such that $\Lambda A.\lambda a.\lambda g.g^n a = \Lambda A.\lambda a.\lambda g.g^m a$. Then by setting $A = \mathbb{N}$, $a = \mathsf{0}$, and $g = \mathsf{succ}$, we have $\mathsf{succ}^n(\mathsf{0}) = \mathsf{succ}^m(\mathsf{0})$, or in other words $n = m$ as required. Hence, we again have an isomorphism.

Corollary 44. *There is an isomorphism* $[\![\forall X.X \to (X \to X) \to X]\!]_o \cong \mathbb{N}$.

The isomorphisms proven in Corollary 42 and Corollary 44 both allude to a more general theorem. It turns out that this isomorphism is the result of the universal property of an initial algebra, which we prove this in Theorem 61.

To reach Theorem 61 we first recall some basic properties about graph relations in Section 3.3.1 and of initial algebras in Section 4.4.2.

3.3.1 GRAPH RELATIONS

We used graph relations in both Theorem 41 and Theorem 43. These two theorems are examples of a more general result which we prove later in this chapter (Theorem 61). Before we can prove the general result we need to look at graph functors in more detail. The key result is called the Graph Lemma (Theorem 48) and is often proven by induction on type derivations. However, here we present the fibrational approach, which relies on defining the graph of a function both in terms of reindexing and opreindexing.

Definition 45 (Graph Relation on Functions, Concretely). Let $f : A \to B$ be a function in Set. Then the *graph relation on* f is defined by $\langle f \rangle = \{(a, fa) \mid a \in A\} \subseteq A \times B$.

Abstractly, we can characterise the graph relation as follows.

Theorem 46. *Let* $f : A \to B$ *be a function in* Set. *Then*

$$\langle f \rangle = (f, \mathrm{id}_B)^* \mathsf{Eq}\, B \qquad \textit{and} \qquad \langle f \rangle = \Sigma_{(\mathrm{id}_A, f)} \mathsf{Eq}\, A.$$

Proof. By Example 16 we have that $(f, \mathrm{id}_B)^* \mathsf{Eq}\, B = \{(a, b) \mid (fa, \mathrm{id}_B b) \in \mathsf{Eq}\, B\}$. Hence, $(f, \mathrm{id}_B)^* \mathsf{Eq}\, B = \{(a, b) \mid fa = b\}$ which is clearly equal to $\langle f \rangle$, and a similar argument shows that $\langle f \rangle = \Sigma_{(\mathrm{id}_A, f)} \mathsf{Eq}\, A$. $\square$

Since the graph can be defined using the reindexing functor (equivalently the opreindexing functor) we also have an action on morphisms, which is characterised by the following theorem.

Theorem 47. *Suppose that $f : A \to B$ and $g : A' \to B'$ are functions. Then two functions $\alpha : A \to A'$ and $\beta : B \to B'$ define a relation homomorphism $(\alpha, \beta) : \langle f \rangle \to \langle g \rangle$ if, and only if, the following diagram commutes.*

$$\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[d, "f"'] & A' \arrow[d, "g"] \\
B \arrow[r, "\beta"'] & B'
\end{tikzcd} \tag{3.3}$$

Proof. Suppose that (a, fa) in $\langle f \rangle$. Then we have that $(\alpha, \beta)(a, fa) = (\alpha a, \beta fa)$ is an element of $\langle g \rangle$ if, and only if, $\beta fa = g\alpha a$, as required. □

Graph relations interact very nicely with the semantics of System F, since types define graph preserving functors. This is a well-known result that is normally proven using induction, but here we are able to prove it fibrationally.

Theorem 48 (Graph Lemma). *Suppose $(T_1, T_0 \times T_0) : \mathsf{Rel}(p) \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)$ is an equality preserving fibred functor. Then for any function $f : A \to B$ we have $T_1\langle f \rangle = \langle T_0 f \rangle$.*

Proof. First note that we have a Cartesian morphism $(T_0 f, \mathrm{id}_{T_0 B}) : \langle T_0 f \rangle \to \mathsf{Eq}\, B$ by the definition of $\langle T_0 f \rangle$ and a morphism $T_1(f, \mathrm{id}_B) : T_1\langle f \rangle \to T_1(\mathsf{Eq}\, Y)$ by the definition of $\langle f \rangle$ and the functoriality of T. Additionally $\mathsf{Eq}(T_0 B) = T_1(\mathsf{Eq}\, B)$ since $(T_1, T_0 \times T_0)$ is an equality preserving functor. Then since $(T_0 f, \mathrm{id}_{T_0 B})$ is a Cartesian morphism and the following diagram commutes,

$$\begin{tikzcd}
T_0 A \times T_0 B \arrow[r, "{(T_0 f, \mathrm{id}_{T_0 B})}"] \arrow[d, "{(\mathrm{id}_{T_0 A}, \mathrm{id}_{T_0 B})}"'] & T_0 B \times T_0 B \arrow[d, "="] \\
T_0 A \times T_0 B \arrow[r, "{(T_0 f, \mathrm{id}_{T_0 B})}"'] & T_0 B \times T_0 B
\end{tikzcd}$$

we have a unique map $(\mathrm{id}_{T_0A}, \mathrm{id}_{T_0B}) : T_1\langle f\rangle \to \langle T_0 f\rangle$. Hence, $T_1\langle f\rangle \subseteq \langle T_0 f\rangle$.

By using the definition of the graph of a function using opreindexing we can dualise these arguments to conclude that $\langle T_0 f\rangle \subseteq T_1\langle f\rangle$, and hence $T_1\langle f\rangle = \langle T_0 f\rangle$, as required. $\square$

Notice that in the Graph Lemma the codomain of the fibred functor $(T_1, T_0 \times T_0)$: $\mathsf{Rel}(p) \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)$ is *not* discrete. This was needed to obtain a morphism $T_1(f, \mathrm{id}_B)$: $T_1\langle f\rangle \to T_1(\mathsf{Eq}\, Y)$ and was crucial to the proof. When we come to use the Graph Lemma later, we will ensure that we are working with types that define fibred functors on non-discrete domains by only working with functorial type expressions (Definition 57).

3.3.2 INITIAL ALGEBRAS

Throughout functional programming inductive data types can be seen in abundance. The most common examples are natural numbers, lists and trees. Categorically, the construction of inductive data types is given by defining an initial algebra.

In the literature there are several proofs of the existence of initial algebras in parametric models, see, e.g., example [28, 29] and a general fibrational approach [2] developed by the author of this thesis as part of collaborative work. We will not recount the fibrational approach in full generality but instead specialise it to within the relations fibration. But before we reach that point, we first recall some basic properties about initial algebras, beginning with the definition.

Definition 49 (Algebra). Let $F : \mathcal{C} \to \mathcal{C}$ be an endofunctor. Then an F-*algebra* is a pair (A, k) where A is an object in $\mathcal{C}$ and $k : FA \to A$ is a morphism in $\mathcal{C}$. We call A the *carrier* of the F-algebra and k the *structure map*.

Definition 50 (Algebra Homomorphism). For any two F-algebras (A, k) and (A', k'), an F-*algebra homomorphism* $(A, k) \to (A', k')$ is a morphism $f : A \to A'$ in $\mathcal{C}$ such that the following diagram commutes.

$$\begin{tikzcd} FA \arrow[r, "Ff"] \arrow[d, "k"'] & FA' \arrow[d, "k'"] \\ A \arrow[r, "f"'] & A' \end{tikzcd}$$

Definition 51 (Initial Algebra). We say that an F-algebra (A, k) is an *initial F-algebra* if for any F-algebra (A', k') there exists a unique F-algebra homomorphism $fold_{A'}\ k' : (A, k) \to (A', k')$. If for every algebra (A', k') the morphism $fold_A\ k$ merely exists (i.e. is not required to be unique), then we call (A, k) a *weak initial algebra.* We denote a (weak) initial algebra by $(\mathsf{\mu F}, in)$.

A very simple example of an initial algebra is given by a set.

Example 52. *Let A be a set and $F : \mathsf{Set} \to \mathsf{Set}$ be the functor defined on objects by $FX = A$ and on morphisms $Ff = \mathrm{id}_A$. Then there exists an initial F-algebra (A, id_A). To see this let (A', k') be an F-algebra and define $fold_{A'}\ k' : A \to A'$ by $fold_{A'}\ k' = k'$, which is trivially an F-algebra and unique.*

Initiality is a useful property and allows us to prove the following result.

Theorem 53. *Let $F : \mathcal{C} \to \mathcal{C}$ be a functor. If $(\mathsf{\mu F}, in)$ is an initial algebra, then $fold_{\mathsf{\mu F}}\ in = \mathrm{id}_{\mathsf{\mu F}}$.*

Proof. By uniqueness of the mediating morphism $fold_{\mathsf{\mu F}}\ in : \mathsf{\mu F} \to \mathsf{\mu F}$. □

A helpful theorem for understanding initial algebras is given by Lambek's Lemma (Theorem 54), which says that the structure map of an initial algebra is an isomorphism.

Theorem 54 (Lambek's Lemma). *Let $F : \mathcal{C} \to \mathcal{C}$ be a functor. If $(\mathsf{\mu F}, in)$ is an initial F-algebra then $in : F(\mathsf{\mu F}) \to \mathsf{\mu F}$ is an isomorphism, where the inverse $in^{-1} : \mathsf{\mu F} \to F(\mathsf{\mu F})$ is given by $in^{-1} = fold_{F(\mathsf{\mu F})}\ F(in)$.*

Proof. Since $F(in) : F^2(\mu\mathsf{F}) \to F(\mu\mathsf{F})$ is an F-algebra, we have that the left-hand side of the following diagram commutes by initiality and the right-hand side trivially.

$$\begin{tikzcd} F(\mu\mathsf{F}) \arrow[r, "F\left(fold_{F(\mu\mathsf{F})}\, F(in)\right)"] \arrow[d, "in"'] & F^2(\mu\mathsf{F}) \arrow[r, "F(in)"] \arrow[d, "F(in)"'] & F(\mu\mathsf{F}) \arrow[d, "in"] \\ \mu\mathsf{F} \arrow[r, "fold_{F(\mu\mathsf{F})}\, F(in)"'] & F(\mu\mathsf{F}) \arrow[r, "in"'] & \mu\mathsf{F} \end{tikzcd} \tag{3.4}$$

By initiality we have that $in \circ \left(fold_{F(\mu\mathsf{F})}\, F(in)\right) = fold_{\mu\mathsf{F}}\, in$ and hence by Theorem 53 we have that $in \circ \left(fold_{F(\mu\mathsf{F})}\, F(in)\right) = \mathrm{id}_{\mu\mathsf{F}}$. For the reverse direction note that,

$$\begin{aligned} \mathrm{id}_{F(\mu\mathsf{F})} &= F(in \circ fold_{F(\mu\mathsf{F})}\, F(in)) && \text{by functoriality,} \\ &= F(in) \circ F(fold_{F(\mu\mathsf{F})}\, F(in)) && \text{by functoriality.} \end{aligned}$$

Hence, by the left-hand side of Diagram 3.4 we have that $fold_{F(\mu\mathsf{F})}\, F(in) \circ in = \mathrm{id}_{F(\mu\mathsf{F})}$, as required. □

If an initial algebra exists for a functor, then Lambek's Lemma can be used to help work out what it will be. Let's consider the functor $F : \mathsf{Set} \to \mathsf{Set}$ defined by $FX = 1 + X$. If F has an initial algebra $(\mu\mathsf{F}, in)$ then by Theorem 54 there is an isomorphism $1 + \mu\mathsf{F} \cong \mu\mathsf{F}$. Hence, any element x of the set $\mu\mathsf{F}$ corresponds to to $in(\star)$ or $in(x')$ for some $x' \in \mu\mathsf{F}$, which looks a lot like the natural numbers. To confirm that the natural numbers is actually an initial F-algebra we still need to do the proof.

Example 55. *The initial algebra of the functor $F : \mathsf{Set} \to \mathsf{Set}$ defined by $FX = 1 + X$, is given by the pair $(\mathbb{N}, [0, \mathsf{succ}])$, where $[0, \mathsf{succ}] : 1 + \mathbb{N} \to \mathbb{N}$ is defined by $0(\star) = 0$ and $\mathsf{succ}(n) = n + 1$. To see this let $[k_0, k_1] : 1 + A \to A$ be an F-algebra, then we define $fold_{\mathbb{N}}\,[k_0, k_1] : \mathbb{N} \to A$ inductively by $(fold_{\mathbb{N}}\,[k_0, k_1])0 = k_0(\star)$ and $(fold_{\mathbb{N}}\,[k_0, k_1])(n+1) = k_1\big((fold_{\mathbb{N}}\,[k_0, k_1])n\big)$. We leave it as a simple exercise to show that $fold_{\mathbb{N}}\,[k_0, k_1]$ is an F-algebra homomorphism and that it is unique.*

Similarly, let's have a look at the functor $FX = S \times (1 + X)$ for some set S. Since Set is distributive, this functor is equivalent to $FX = (S \times 1) + (S \times X)$. By Lambek's Lemma we have that if there exists an initial algebra $\mu\mathsf{F}$ then any element x in $\mu\mathsf{F}$ corresponds to either $x = in(s, *)$ for some $s \in S$ or $x = in(s, x')$ for some $s \in S$ and $x' \in \mu\mathsf{F}$, which looks a like non-empty lists of elements of S.

Example 56. *Let S be a set and suppose that $F : \mathsf{Set} \to \mathsf{Set}$ is the functor defined by $FX = (S \times 1) + (S \times X)$. Then an initial algebra for F is given by $(\mathsf{List}^+(S), [\iota, \mathsf{cons}])$, where $\mathsf{List}^+(S)$ denotes the collection of non-empty positive lists with elements in S, the function $\iota : S \times 1 \to \mathsf{List}^+(S)$ is defined by $\iota(s, \star) = [s]$, and the function $\mathsf{cons} : S \times \mathsf{List}^+(S) \to \mathsf{List}^+(S)$ is given by $\mathsf{cons}(s, [ss]) = [s, ss]$. For any F-algebra $[k_0, k_1] : FA \to A$ the mediating morphism morphism $fold\,[k_0, k_1] : \mathsf{List}^+(S) \to A$ is given inductively by $fold\,[k_0, k_1][s] = k_0(s, \star)$ and $fold\,[k_0, k_1][s, ss] = k_1(s, fold\,[k_0, k_1][ss])$.*[1] *It is left as an exercise to show $fold$ is unique.*

3.3.3 INITIAL ALGEBRAS FOR SYSTEM F

We are now ready to show the generalisation of Corollary 44 and Corollary 42 using initial algebras. To this end we first show that the interpretation of every System F type with one free type variable that defines an endofunctor has a weakly initial algebra in the relations fibration.

To formulate this theorem we need to be careful about which types we are talking about, since we need to have an action on morphisms to talk about initial algebras. This leads us to the notion of a *functorial type*.

Definition 57 (Functorial Type). A type $\Delta \vdash T$ Type is *functorial* if there exists a term $t_{map} : \forall X.\forall Y.(X \to Y) \to TX \to TY$ such that

- $t_{map}[X][X]\mathrm{id}_X = \mathrm{id}_{TX}$ and
- $(t_{map}[Y][Z]g) \circ (t_{map}[X][Y]f) = t_{map}[X][Z](g \circ f)$.

All type expressions with one free type variable occurring only positively give rise to functorial type expressions, and in the semantics functorial type expressions give functors.

[1] We omit the subscript on $fold$ for typographic reasons.

We will now show that for any functorial type $\Delta \vdash T$ Type an initial $[\![\Delta \vdash T\ \mathsf{Type}]\!]_o$-algebra exists. We begin by showing weak initiality.

Theorem 58. *Let $X \vdash T$ Type be a functorial type and denote $[\![\Delta \vdash T\ \mathsf{Type}]\!]_o$ by T_0, $[\![\Delta \vdash T\ \mathsf{Type}]\!]_r$ by T_1, $[\![\forall X.(TX \to X) \to X]\!]_o$ by Z_0, and $[\![\forall X.(TX \to X) \to X]\!]_r$ by Z_1. Then Z_0 is a weakly initial T_0-algebra and Z_1 is a weakly initial T_1-algebra.*

Proof. We begin by constructing the mediating morphisms.

Mediating Morphisms: Suppose that A is a set and that $k_A : T_0 A \to A$ is a T_0-algebra. Then since $[\![(TX \to X) \to X]\!]_o A = (T_0 A \Rightarrow A) \Rightarrow A$, we can define the function $fold_A\, k_A : Z_0 \to A$ by $(fold_A\, k_A) f = f_A(k_A)$.[2]Similarly, for a relation R on $A \times B$ in Rel and a T_1 algebra $k_R : T_1 R \to R$ where $k_R = (k_A, k_B)$, we define the relation homomorphism $fold_R\, k_R : Z_1 \to R$ by $(fold_R\, k_R)(f, g) = (f_A k_A,\ g_B k_B)$. We see that $fold_R\, k_R$ is a relation homomorphism by the definition of Z_1, and clearly $(fold_R\, k_R)(f, g) = (fold_A\, k_A, fold_B\, k_B)$.

The Structure Maps: The morphism $in_0 : T_0 Z_0 \to Z_0$ in Set is given using $fold_A$ by $in_0\, x = \lambda A.\lambda k_A.\, k_A\big((T_0 fold_A\, k_A) x\big)$ and we similarly define $in_1 : T_1 Z_1 \to Z_1$ by $in_1(x, y) = \lambda R.\lambda k_R.\, \big(k_A(T_0 fold_A\, k_A)x,\ k_B(T_0 fold_B\, k_B)y\big)$.

Algebra Homomorphisms: All that remains to prove Theorem 58 is to show that $fold_A\, k_A$ is a T_0-algebra homomorphism for any T_0-algebra $k_A : T_0 A \to A$, and that $fold_R\, k_R$ is a T_1-algebra homomorphism for any T_1-algebra $k_R : T_1 R \to R$, which is done as follows.

$$\begin{aligned} fold_A\, k_A(in_0\, x) &= in_0\, x\, A\, k_A \quad \text{by the definition of } fold_A\,, \\ &= k_A((T_0 fold_A\ \ k_A)x) \quad \text{by definition of } in_0. \end{aligned}$$

A similar argument holds for the relational case. □

To show that Z_0 as defined above in Theorem 58 is a strong initial algebra requires the following theorem.

[2]Notice that there are two different meanings for a subscript here. The first is that algebra homomorphisms, e.g. k_A, are labelled with their codomain, and the second is the projection of a parametric function, e.g., f_A. This clash is slightly unfortunate, but by pointing this distinction out, we hope to avoid confusion.

Theorem 59. *Suppose that $k_A : T_0 A \to A$ and $k_B : T_0 B \to B$ are T_0-algebras and that $h : k_A \to k_B$ is T_0-algebra homomorphism. Then $h \circ fold_A\, k_A = fold_B\, k_B$.*

Proof. By Theorem 47 we have that (k_A, k_B) is relation homomorphism $\langle T_0 h\rangle \to \langle h\rangle$, and by the Graph Lemma (Theorem 48) we have that $\langle T_0 h\rangle = T_1\langle h\rangle$ and hence the pair $(k_A, k_B) : T_1\langle h\rangle \to \langle h\rangle$ is a T_1-algebra. Therefore, since Z_1 is weakly initial we have a relation homomorphism $fold_{\langle h\rangle}\,(k_A, k_B) : Z_1 \to \langle h\rangle$. By the Identity Extension Lemma (Theorem 37) we have that $Z_1 = \mathsf{Eq}\, Z_0$, and by a trivial calculation we have that $\mathsf{Eq}\, Z_0 = \langle \mathrm{id}_{Z_0}\rangle$. Hence, $fold_{\langle h\rangle}\,(k_A, k_B)$ is a map between graphs $\langle \mathrm{id}_{Z_0}\rangle \to \langle h\rangle$, and so by Theorem 47 we have that $h \circ fold_A\, k_A = fold_B \circ \mathrm{id}_{Z_0}$ as required. □

As a simple corollary we have the following result, reminiscent of Theorem 53.

Corollary 60. *Let Z_0, in_0, and $fold_{Z_0}$ be defined as in Theorem 59. Then*

$$fold_{Z_0}\, in_0 = \mathrm{id}_{Z_0}.$$

Proof. Let $k_A : T_0 A \to A$ be a T_0-algebra, then $fold_A\, k_A$ is a T_0-algebra homomorphism $Z_0 \to A$ by Theorem 58. Hence, by Theorem 59 we have that $fold_A\, k_A \circ fold_{Z_0}\, in_0 = fold_A\, k_A$. In other words, for any $x \in Z_0$, we have $(fold_{Z_0}\, in_0\, x) A\, k_A = x\, A\, k_A$. Hence by extensionality we have that $fold_{Z_0}\, in_0\, x = x$ as required. □

And finally we can prove the main theorem of this chapter.

Theorem 61. *The pair (Z_0, in_0) is an initial T_0-algebra.*

Proof. Suppose that $k_A : T_0 A \to A$ is T_0-algebra and h is T_0-algebra homomorphism $in_0 \to k_A$. Then by Theorem 59 we have that $h \circ fold_{Z_0}\, in_0 = fold_A\, k_A$ as required. □

To conclude this chapter, let's see this theorem in action.

Example 62. *Suppose that T is the constant type $T = A$. Then we have that $[\![\forall X.(A \to X) \to X]\!]_o$ is the initial algebra of the functor $T_0 X = A$, denoted μT_0. By Example 52 we already know that $\mu T_0 = A$ and hence we have that $[\![\forall X.(A \to X) \to X]\!]_o = A$.*

Similarly, we can show that $[\![\forall X.(1 + X \to X) \to X]\!]_o \cong \mathbb{N}$ using Example 55 and $[\![\forall X.(S \times (1 + X) \to X) \to X]\!]_o \cong \mathsf{List}^+(S)$ using Example 56.

Using Theorem 61 and Lambek's Lemma we can see why there are no set-theoretic models of System F in classical logic. Consider $TX = (X \to 2) \to 2$, which by Theorem 61 has an initial algebra μT_0. Using classical logic we also have that the interpretation of T is the double-powerset functor $\mathbb{P} \circ \mathbb{P} : \mathsf{Set} \to \mathsf{Set}$. Hence, by Lambek's Lemma we have $\mathbb{P}(\mathbb{P}(\mu T_0)) \cong \mu T_0$, i.e., $|\mu T_0| = 2^{2^{|\mu T_0|}}$, contradicting Cantor's Theorem, and hence there can be no set-theoretic models of System F in classical logic.

In this Chapter we have seen a fibrational presentation of Reynolds's relational parametricity, as well as a proof of one of the most important consequences of parametricity — the existence of initial algebras (Theorem 61). In the next chapter we will explore what happens when we change the role played by sets to G-sets.

CHAPTER 4

GROUP ACTION PARAMETRICITY

Group actions are about symmetry. They allow a departure from solely talking about symmetries of geometries and allow a discussion about symmetries of more abstract mathematical objects. This level of abstraction means that group actions arise in many different places including in combinatorics, Galois theory, quantum mechanics, representation theory, topology, and of course computer science.

Within computer science group actions arise in a large variety of subjects, ranging from the study of quotient containers [30] to variable binding [31]. Group actions often provide a handy tool for formulating invariance properties, i.e., when a structure or data type is invariant under a particular symmetry. Similarly parametricity can be used to study invariance.

Recall the "free theorem" proven in Theorem 40, which said that for any parametric function π in $[\![\forall X.\forall Y.X \times Y \to X]\!]_o$, we have that for any two functions $f : A \to A$ and $g : B \to B$,

$$\pi_{A,B} \circ (f \times g) = f \circ \pi_{A,B}. \tag{4.1}$$

This result shows that a parametric function in $[\![\forall X.\forall Y.X \times Y \to X]\!]_o$ must act uniformly. If one imagines that f and g are symmetries on A and B, then we see that Equation 4.1 says that $\pi_{A,B}$ must preserve those symmetries, i.e. is invariant. The uniformity conditions imposed on parametric functions mean that relations are always preserved, and the "free theorem" argument allows us to deduce invariance properties. This line of reasoning has allowed parametricity to be used to describe invariance of data type representations [26], geometric invariance [25] as well as giving the invariance of physical systems required to use Noether's theorem [32].

Motivated by the abundance of group actions in mathematics and computer science, and by the invariance properties of parametricity, in this chapter we extend Reynolds's relational parametricity to the context of group actions. We will begin by recalling a few basic examples of group actions in Section 4.1. We will then introduce the notion of relations on G-sets in Section 4.2, before providing a relational semantics for System F using G-sets and equivariant relations in Section 4.3. Once we have defined the relational semantics we will be able to see a surprising difference between the set-valued relational model and the G-set relational model in Sections 4.4.1 – 4.4.4.

4.1 EXAMPLES OF GROUP ACTIONS

For a group G there are plenty of ways to equip it with a group action, and here we explore three of them .[1] For each example we introduce a different notation for the underlying set even though it remains the same. Though this is may seem a little heavy handed now, in Section 4.4.2 the distinction will be necessary and using this notation will help to avoid mixing up the different G-sets.

Example 63 (Trivial Action). *Let G_T denote the underlying set of the group G. Then equip G_T with the group action $\cdot_{G_T} : G \times G_T \to G_T$ given by*

$$\sigma \cdot_{G_T} \pi = \pi.$$

[1]For a reminder of the definition of a group action see Chapter 2 Section 5.6.

Both the identity and compatibility axioms are trivially satisfied by this action, which we call $\cdot_{G_T}$ *the* trivial group action*.*

Example 64 (Group Multiplication Action). *Let* G_M *denote the underlying set of* G*. Then we equip* G_M *with the group action* $\cdot_{G_M} : G \times G_M \to G_M$ *given by*

$$\sigma \cdot_{G_M} \pi = \sigma\pi.$$

The identity axiom is satisfied by the properties of the identity element in the group, and the compatibility axiom is satisfied by the associativity axiom of the group. We call $\cdot_{G_M}$ *the* group multiplication action*.*

The group multiplication action has the property that for any G-set X, and any element x in X the function $\phi_x : G_M \to X$ defined by $\phi_x(\pi) = \pi \cdot_X x$ is equivariant, as shown by a simple calculation.

Example 65 (Conjugate Action). *Let* G_C *denote the underlying set of* G*. Then we equip* G_C *with the group action* $\cdot_{G_C} : G \times G_C \to G_C$ *given by*

$$\sigma \cdot_{G_C} \pi = \sigma\pi\sigma^{-1}.$$

We call $\cdot_{G_C}$ *the* conjugate action*. The identity axiom is satisfied since* $e\pi e^{-1} = \pi$ *for any element* π *of* G_C*. The compatibility axiom is satisfied by using the associativity of the group, as seen in the following quick calculation. For any two elements* σ *and* π *in the group* G *and any element* τ *in* G_C *we have,*

$$\begin{aligned}
\sigma \cdot_{G_C} (\pi \cdot_{G_C} \tau) &= \sigma \cdot_{G_C} (\pi\tau\pi^{-1}) && \textit{by definition of } \cdot_{G_C}, \\
&= \sigma(\pi\tau\pi^{-1})\sigma^{-1} && \textit{by definition of } \cdot_{G_C}, \\
&= (\sigma\pi)\tau(\sigma\pi)^{-1} && \textit{by associativity of } G, \\
&= (\sigma\pi) \cdot_{G_C} \tau && \textit{as required.}
\end{aligned}$$

The conjugate action has the additional property that for any G-set X, the group action $\cdot_X : G_C \times X \to X$ is equivariant.

Before we begin looking at group actions as a model for System F let's have a look at one last class of examples of G-sets, using lists.

Example 66 (Lists). *Suppose that* $(X, \cdot_X)$ *is G-set. Then we denote the set of lists of elements of X by* $\mathsf{List}(X)$*, and this induces a G-set* $(\mathsf{List}(X), \cdot_{\mathsf{List}(X)})$ *defined by*

$$\pi \cdot_{\mathsf{List}(X)} [x_1, \ldots, x_n] = [\pi \cdot_X x_1, \ldots, \pi \cdot_X x_n].$$

Similarly, for any G-set $(X, \cdot_X)$ *let* $\mathsf{List}^+(X)$ *denote the set of non-empty lists of elements of X. Then three examples of G-sets that we will see later are* $(\mathsf{List}^+(G_T), \cdot_{\mathsf{L}G_T})$, $(\mathsf{List}^+(G_M), \cdot_{\mathsf{L}G_M})$ *and* $(\mathsf{List}^+(G_C), \cdot_{\mathsf{L}G_C})$*, where*

- $\sigma \cdot_{\mathsf{L}G_T} [\pi_1, \ldots, \pi_n] = [\sigma \cdot_{G_T} \pi_1, \ldots, \sigma \cdot_{G_T} \pi_n]$,
- $\sigma \cdot_{\mathsf{L}G_M} [\pi_1, \ldots, \pi_n] = [\sigma \cdot_{G_M} \pi_1, \ldots, \sigma \cdot_{G_M} \pi_n]$, *and*
- $\sigma \cdot_{\mathsf{L}G_C} [\pi_1, \ldots, \pi_n] = [\sigma \cdot_{G_C} \pi_1, \ldots, \sigma \cdot_{G_C} \pi_n]$.

It is left as a simple exercise to show that these actions do actually define G -sets.

Not all G-sets act on groups, but all of the ones in this thesis do. Other examples of G-sets include the set $\{1, \ldots, n\}$, which is acted on by the symmetric group S_n by permuting its elements, or the set of vertices of a polyhedron is acted on by its symmetry group, to name but a few. For the interested we suggest looking in any good introductory text on group theory (e.g. [33]) for more examples.

4.2 RELATIONS ON G-SETS

In Chapter 3 we showed how a parametric model for System F can be given using the relations fibration. A similar semantics can be given using G-sets and equivariant relations, and we can describe an analogous notion of relation. We have seen how the relations fibration arises via pullback in Example 16 and we can use the same principle here to give a notion of a G-set relation. Consider the following pullback square:

$$\begin{tikzcd}
\mathsf{Rel}([G,\mathsf{Set}]) \arrow[r] \arrow[d, "\mathsf{Rel}(p)"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{Sub}([G,\mathsf{Set}]) \arrow[d, "p"] \\
{[G,\mathsf{Set}] \times [G,\mathsf{Set}]} \arrow[r, "\_ \times \_"'] & {[G,\mathsf{Set}]}
\end{tikzcd}$$

If S is a subset of A and $\pi \cdot_A s \in S$ for all π in G and s in S, then the pair $(S, \cdot_S)$ is a subobject of A in $[G, \mathsf{Set}]$, where $\cdot_S$ denotes the restriction of the group action $\cdot_A$ to S. Moreover, a morphism $S \subseteq A \to S' \subseteq A'$ in $\mathsf{Sub}([G, \mathsf{Set}])$ is given by an equivariant map $f : A \to A'$ such that for any element s of S, the element $f(s)$ is in S'.

Hence, objects of $\mathsf{Rel}([G, \mathsf{Set}])$ in the fibre above (A, B) are given by subobjects of $A \times B$, and a morphism $R \subseteq A \times B \to R' \subseteq A' \times B'$ in $\mathsf{Rel}([G, \mathsf{Set}])$ is given by a pair of equivariant maps $f : A \to A'$ and $g : B \to B'$, such that for any pair (a, b) in R, the pair (fa, gb) is in R'. This leads us to the following definition.

Definition 67 (Equivariant Subset and Equivariant Relation). Let S, A and B be G-sets. Then we call S an *equivariant subset (of A)* if S is a subobject of A, i.e., if S is a subset of A and $\pi \cdot_A s \in S$ for all π in G and s in S. Moreover, we call an equivariant subset of a product $A \times B$ an *equivariant relation (on $A \times B$)*. We will denote *equivariant* subsets by $S \subseteq A$. When we wish to talk about subsets that are not equivariant, we will make this explicit by writing $\subseteq_{\mathsf{Set}}$.

4.2.1 STRUCTURE ON RELATIONS ON G-SETS

For $\mathsf{Rel}([G, \mathsf{Set}])$ to be of any use as a model of System F it needs to be Cartesian closed and luckily, it is.

Terminal Object: The terminal equivariant relation is given by the one object equivariant subset $1_{\mathsf{Rel}([G,\mathsf{Set}])} \subseteq 1_{[G,\mathsf{Set}]} \times 1_{[G,\mathsf{Set}]}$, with the action given by the unique map $G \times 1_{\mathsf{Rel}([G,\mathsf{Set}])} \to 1_{\mathsf{Rel}([G,\mathsf{Set}])}$. We will usually just denote both $1_{\mathsf{Rel}([G,\mathsf{Set}])}$ and $1_{[G,\mathsf{Set}]}$ by 1, leaving the subscript to be inferred by context.

Products: The product of two equivariant relations $R_1 \subseteq A_1 \times B_1$ and $R_2 \subseteq A_2 \times B_2$ is given by

$$R_1 \times R_2 = \{(a_1, a_2, b_1, b_2) \mid (a_1, b_1) \in R_1 \;\; \mathit{and} \;\; (a_2, b_2) \in R_2\},$$

with the group action given component-wise, i.e., $\pi \cdot_{R_1 \times R_2} (a_1, a_2, b_1, b_2) = (\pi \cdot_{A_1} a_1,\, \pi \cdot_{A_2} a_2,\, \pi \cdot_{B_1} b_1,\, \pi \cdot_{B_2} b_2)$, which makes $R_1 \times R_2$ an equivariant relation on $(A_1 \times A_2) \times (B_1 \times B_2)$.

Exponential Objects: The exponential of two equivariant relations $R_1 \subseteq A_1 \times B_1$ and $R_2 \subseteq A_2 \times B_2$ is given by

$$R_1 \Rightarrow R_2 = \{(f, g) \mid \forall (a, b) \in R_1, \;\; (fa, fb) \in R_2\},$$

with the group action inherited from the G-set $(A_1 \Rightarrow A_2) \times (B_1 \Rightarrow B_2)$, which makes $R_1 \Rightarrow R_2$ an equivariant relation on $(A_1 \Rightarrow A_2) \times (B_1 \Rightarrow B_2)$.[2]

Additional to the Cartesian closed structure we have a canonical equality functor analogous to Definition 35.

Definition 68. The *equality functor on G-sets* $\mathsf{Eq} : [G, \mathsf{Set}] \to \mathsf{Rel}([G, \mathsf{Set}])$ is defined on objects by $\mathsf{Eq}\, A = \{(a, a) \mid a \in A\}$, where the group action is inherited from $A \times A$ and on morphisms by $\mathsf{Eq}\, f = (f, f)$.

An equality functor can be defined in many different settings (using fibrational structure) but for group actions the following important property holds.

Proposition 69. *The equality functor* $\mathsf{Eq} : [G, \mathsf{Set}] \to \mathsf{Rel}([G, \mathsf{Set}])$ *is full and faithful.*

Proof. To see that Eq is full suppose that (f, g) is an equivariant map $\mathsf{Eq}\, A \to \mathsf{Eq}\, B$. Then by definition, for all a in A, we have that $(fa,\, gb) \in \mathsf{Eq}\, Y$, i.e., $fa = gb$. Hence, $f = g$ meaning that $\mathsf{Eq}\, f = (f, g) = \mathsf{Eq}\, g$ as required.

For faithfulness, suppose that we have two equivariant maps $f, g : A \to B$ such that $\mathsf{Eq}\, f = \mathsf{Eq}\, g$. Then, $(f, f) = (g, g)$ and hence $f = g$ as required. □

[2]As remarked on Page 22, recall that an element f of the exponential $A_1 \Rightarrow A_2$ is simply a function $f : A_1 \to A_2$ and is *not* required to be equivariant. It is important to remember the distinction between *elements of the exponential* and *morphisms in the category*, as they are not the same.

In addition to the equality functor, we also have the graph functor on G-sets, which we now define. Recall that $\overrightarrow{[G, \mathsf{Set}]}$ is the arrow category on $[G, \mathsf{Set}]$ with objects given by equivariant maps $f : A \to B$ and morphisms $(\alpha, \beta) : f \to f'$ are given by commuting squares.

$$\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[d, "f"'] & A' \arrow[d, "f'"] \\
B \arrow[r, "\beta"'] & B'
\end{tikzcd}$$

Definition 70 (Graph Functor on G-Sets). The *graph functor* $\langle \_ \rangle : \overrightarrow{[G, \mathsf{Set}]} \to \mathsf{Rel}([G, \mathsf{Set}])$ is defined on objects by $\langle f \rangle = \{(a, b) \in A \times B \mid b = fa\}$, which is a relation on $A \times B$, and on morphisms by $\langle (\alpha, \beta) \rangle (a, fa) = (\alpha a, \beta f a)$.[3]

Note that the action of the graph functor on morphisms is well-defined since $\beta f a = f' \alpha a$.

Additionally, we can define the "graph" of an element of the exponential in the same way. However, it is only the case that the graph of a function is an *equivariant* relation when the function is also *equivariant*.

Lemma 71. *Suppose that $f : A \to_{\mathsf{Set}} B$ is a function. Then $\langle f \rangle = \{(a, b) \in A \times B \mid b = fa\}$ is an equivariant relation on $A \times B$ if, and only if, f is an equivariant map.*

Proof. The graph relation $\langle f \rangle$ is equivariant if, and only if, for all (a, fa) in $\langle f \rangle$ and π in G, that $(\pi \cdot_A a, \pi \cdot_B fa) \in \langle f \rangle$ holds, which is true if, and only if, $f(\pi \cdot_A a) = \pi \cdot_B (fa)$, i.e., if f is an equivariant map. $\square$

Analogous to Theorem 47 we are able to characterise when we have an element of the exponential of two graph relations.

Theorem 72. *Suppose that A, A', B, and B' are G-sets. Suppose further that f is an element of the exponential object $A \Rightarrow B$, f' is an element of the exponential object*

[3]This definition is analogous to Definition 45 and can similarly be defined in terms of reindexing. For simplicity we just give the concrete definition here.

$A' \Rightarrow B'$, and $\alpha : A \to A'$ and $\beta : B \to B'$ are equivariant maps. Then $(f, f') \in \langle\alpha\rangle \Rightarrow \langle\beta\rangle$ if, and only if,

$$\begin{tikzcd} A \arrow[r, "\alpha"] \arrow[d, "f"'] & A' \arrow[d, "f'"] \\ B \arrow[r, "\beta"'] & B' \end{tikzcd}$$

commutes in Set.

Proof. The proof is almost identical to Theorem 47. □

4.3 GROUP ACTIONS AS A MODEL OF SYSTEM F

So far in this chapter we have defined the notion of a relation on a G-set and have seen the equality and graph relations for G-sets. Additionally, we have seen that the category of equivariant relations on G-sets is Cartesian closed. It is now possible using the relations fibration for G-sets $\mathsf{Rel}(p) : \mathsf{Rel}([G, \mathsf{Set}]) \to [G, \mathsf{Set}] \times [G, \mathsf{Set}]$ to give a relational semantics for System F using G-sets.

In the G-set relational semantics for System F types are given by equality-preserving fibred functors $[\![\Delta \vdash T \ \mathsf{Type}]\!] : |\mathsf{Rel}(p)|^n \underset{\mathsf{Eq}}{\to} \mathsf{Rel}(p)$. In other words, a type is interpreted as a pair of functors $[\![\Delta \vdash T \ \mathsf{Type}]\!]_o : |[G, \mathsf{Set}]|^n \to [G, \mathsf{Set}]$ and $[\![\Delta \vdash T \ \mathsf{Type}]\!]_r : |\mathsf{Rel}([G, \mathsf{Set}])|^n \to \mathsf{Rel}([G, \mathsf{Set}])$ such that $[\![\Delta \vdash T \ \mathsf{Type}]\!]_r$ is over the product $[\![\Delta \vdash T \ \mathsf{Type}]\!]_o \times [\![\Delta \vdash T \ \mathsf{Type}]\!]_o$ and the equality functor is preserved, i.e., $\mathsf{Eq}\,([\![\Delta \vdash T \ \mathsf{Type}]\!]_o \mathbf{A}) = [\![\Delta \vdash T \ \mathsf{Type}]\!]_r (\mathsf{Eq}\,\mathbf{A})$.

Since both $[G, \mathsf{Set}]$ and $\mathsf{Rel}([G, \mathsf{Set}])$ are Cartesian closed categories we can interpret System F types built up from 1, $\times$ and $\to$ in both categories in the usual way (see Chapter 3). All that remains is the interpretation of $\forall$, which we define as follows.[4]

$$\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A} =$$
$$\{f : \prod_{A \in [G,\mathsf{Set}]} \llbracket \Gamma, X \vdash T \rrbracket_o (\mathbf{A}, A) \mid \forall R \subseteq A \times B,\ (f_A, f_B) \in \llbracket \Gamma, X \vdash T \rrbracket_r (\mathsf{Eq}\,\mathbf{A}, R)\},$$

with the action $\pi \cdot_\forall f = \lambda A.(\pi \cdot_{\llbracket T \rrbracket_o A} f_A)$ and the relational interpretation given by

$$(f, g) \in \llbracket \Gamma \vdash \forall X.T \rrbracket_r \mathbf{R} \quad \textit{iff} \quad \forall R \subseteq A \times B \ \ (f_A, g_B) \in \llbracket \Gamma, X \vdash T \rrbracket_r (\mathbf{R}, R),$$

where the action is inherited from $\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A} \times \llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{B}$. We call a function f in $\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A}$ a *parametric function.*

Equivariance of Interpretation: To see that $\llbracket \Gamma \vdash \forall X.T \rrbracket_r \mathbf{R}$ is actually an equivariant relation on $\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A} \times \llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{B}$, let (f, f') be related in $\llbracket \Gamma \vdash \forall X.T \rrbracket_r \mathbf{R}$. We then have to show that for any element π in G, the pair $\pi \cdot (f, f')$ is related in $\llbracket \Gamma \vdash \forall X.T \rrbracket_r \mathbf{R}$. By the induction hypothesis we have that $\llbracket \Gamma, X \vdash T \rrbracket_r (\mathbf{R}, R)$ is an equivariant relation, and we know that $(f_A, f'_B) \in \llbracket \Gamma, X \vdash T \rrbracket_r (\mathbf{R}, R)$, which means that $\pi \cdot (f_A, f'_B)$ is related in $\llbracket \Gamma, X \vdash T \rrbracket_r (\mathbf{R}, R)$ as required.

So far we have managed to give an interpretation of forall types, but there is still some work to do to show that it is the *correct* interpretation. In Theorem 73 we prove that this interpretation is equality preserving as part of the Identity Extension Lemma for G-sets and in Theorem 74 we show that the definition satisfies a general adjointness property for a relational semantics.

Theorem 73 (Identity Extension Lemma for G-Sets). *For any type* $\Delta \vdash T$ Type*, and any* G*-set* $\mathbf{A}$ *in* $|[G, \mathsf{Set}]|^n$*, we have that* $\mathsf{Eq}\,(\llbracket \Delta \vdash T\ \mathsf{Type} \rrbracket_o \mathbf{A}) = \llbracket \Delta \vdash T\ \mathsf{Type} \rrbracket_r (\mathsf{Eq}\,^n \mathbf{A})$.

Proof. The proof is by structural induction on T. We will prove the forall types case and leave the rest to the reader. To this end we need to show that the interpretation of $\forall$ is

[4]Similarly to Section 3.2.1, we need to be careful when writing large products. We avoid any trouble, by again insisting that we either use the (intuitionistic) internal language of [20] or using the Calculus of Constructions [21] with impredicative Set.

equality preserving, i.e.,

$$\mathsf{Eq}\,(\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A}) = \llbracket \Gamma \vdash \forall X.T \rrbracket_r (\mathsf{Eq}\,\mathbf{A}).$$

First show that we have an inclusion $\mathsf{Eq}\,(\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A}) \subseteq \llbracket \Gamma \vdash \forall X.T \rrbracket_r (\mathsf{Eq}\,\mathbf{A})$. Suppose that f is a parametric function in $\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A}$. Then the pair (f, f) is an element of $\mathsf{Eq}\,(\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A})$, and hence by the definition of $\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A}$, we have that for any equivariant relation R on $A \times B$

$$(f_A, f_B) \in \llbracket \Gamma, X \vdash T \rrbracket_r (\mathsf{Eq}\,\mathbf{A}, R).$$

Or in other words the pair (f, f) is related in $\llbracket \Gamma \vdash \forall X.T \rrbracket_r (\mathsf{Eq}\,\mathbf{A})$ as required. To complete the proof we show the reverse inclusion $\llbracket \Gamma \vdash \forall X.T \rrbracket_r (\mathsf{Eq}\,\mathbf{A}) \subseteq \mathsf{Eq}\,(\llbracket \Gamma \vdash \forall X.T \rrbracket_o \mathbf{A})$. To this end let (f, f') be an element of $\llbracket \Gamma \vdash \forall X.T \rrbracket_r (\mathsf{Eq}\,\mathbf{A})$, then by the definition of $\llbracket \Gamma \vdash \forall X.T \rrbracket_r (\mathsf{Eq}\,\mathbf{A})$, we have that for any equivariant relation R on $A \times B$,

$$(f_A, f'_B) \in \llbracket \Gamma, X \vdash T \rrbracket_r (\mathsf{Eq}\,\mathbf{A}, R). \tag{4.2}$$

In particular, for any G-set A, we have that $(f_A, f'_A) \in \llbracket \Gamma, X \vdash T \rrbracket_r (\mathsf{Eq}\,\mathbf{A}, \mathsf{Eq}\,A)$ by instantiating Equation 4.2 at $R = \mathsf{Eq}\,A$. By the induction hypothesis we have that $\llbracket \Gamma, X \vdash T \rrbracket_r (\mathsf{Eq}\,\mathbf{A}, \mathsf{Eq}\,A) = \mathsf{Eq}\,\llbracket \Gamma, X \vdash T \rrbracket_o (\mathbf{A}, A)$, and hence for all $A \in [G, \mathsf{Set}]$, we have $(f_A, f'_A) \in \mathsf{Eq}\,\llbracket \Gamma, X \vdash T \rrbracket_o (\mathbf{A}, A)$ implying $f = f'$ as required. □

Our interpretation of $\forall$ satisfies the general adjointness property of a relational semantics for System F as stated in [2, Section 4.2]. The adjoint functors are defined as follows.

Let $|\mathsf{Rel}(p)|^n \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)$ be the category whose objects are equality preserving fibred functors from $|\mathsf{Rel}(p)|^n$ to $\mathsf{Rel}(p)$ and whose morphisms are fibred natural transformations between them. Let $\pi_n : |\mathsf{Rel}(p)|^{n+1} \to \mathsf{Rel}(p)^n$ be a projection morphism and denote the pre-composition functor by $\_ \circ \pi_n : (|\mathsf{Rel}(p)|^n \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)) \to (|\mathsf{Rel}(p)|^{n+1} \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p))$. Let $\forall_n = (\forall_r, \forall_o \times \forall_o)$ be the fibred functor defined by,

$$(\forall_o T_o)\mathbf{A} = \{f : \prod_{A \in [G, \mathsf{Set}]} T_o(\mathbf{A}, A) \mid \forall R \subseteq A \times B,\ (f_A, f_B) \in T_r(\mathsf{Eq}\,\mathbf{A}, R)\},$$

with the group action given by $\pi \cdot_{\forall_o} f = \lambda X.(\pi \cdot_{T_o(\mathbf{A},A)} f_A)$. And we define the relation $(\forall_r T_r)\mathbf{R}$ by

$$(f, g) \in (\forall_r T_r)\mathbf{R} \quad \textit{iff} \quad \forall R \subseteq A \times B,\ (f_A, g_B) \in T_r(\mathbf{R}, R),$$

where the group action is inherited from $(\forall_o T_o)\mathbf{A} \times (\forall_o T_o)\mathbf{B}$.

Theorem 74. *For every projection morphism $\pi_n : |\mathsf{Rel}(p)|^{n+1} \to \mathsf{Rel}(p)^n$, the pre-composition functor has a right adjoint $\_ \circ \pi_n \dashv \forall_n$, as described above.*

Proof. To go through all of the details of this proof is rather lengthy, a little tedious, not particularly illuminating and completely standard. For this reason, we define the isomorphism and omit the remainder of the proof.

To simplify the presentation, let $\mathcal{C}_n$ denote the category $|\mathsf{Rel}(p)|^n \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)$. Then we show that the functor $\forall_n$ is the right adjoint to π_n, i.e. $(\forall_r, \forall_o \times \forall_o)$ is right adjoint to $(\pi_r, \pi_o \times \pi_o)$, by constructing an isomorphism $\mathcal{C}_{n+1}(T \circ \pi_n, S) \cong \mathcal{C}_n(T, \forall_n S)$, which is natural in T and S. We begin by defining functions ϕ and ψ

$$\begin{tikzcd}
\mathcal{C}_{n+1}(T \circ \pi_n, S) \arrow[r, bend left, "\phi"] & \mathcal{C}_n(T, \forall S). \arrow[l, bend left, "\psi"]
\end{tikzcd}$$

Defining the Isomorphism: Let $(\alpha_r, \alpha_o \times \alpha_o) : T \circ \pi_n \to S$ be any fibred natural transformation in $\mathcal{C}_{n+1}(T \circ \pi_n, S)$, let $\mathbf{A}$ be an n-tuple of G-sets in $|[G, \mathsf{Set}]|^n$, and let $\mathbf{R} \subseteq \mathbf{A} \times \mathbf{B}$ be an n-tuple of relations in $|\mathsf{Rel}([G, \mathsf{Set}])|^n$. Then we define ϕ as the pair $(\phi_r,\ \phi_o \times \phi_o) : \mathcal{C}_{n+1}(T \circ \pi_n, S) \to \mathcal{C}_n(T, \forall_n S)$ given by $(\phi_o \alpha_o)\mathbf{A} = \lambda a.\lambda A.\alpha_o(\mathbf{A}, A)a$ and $(\phi_r \alpha_r)\mathbf{R} = (\phi_o \alpha_o)\mathbf{A} \times (\phi_o \alpha_o)\mathbf{B}$.

To see that $(\phi_o \alpha_o)\mathbf{A}$ is well-defined, i.e., that for any a in $T_o\mathbf{A}$, $(\phi_o \alpha_o)\mathbf{A}a$ is actually an element of $(\forall_o S_o)\mathbf{A}$, we must show that for any relation $R \subseteq A \times B$ in $|\mathsf{Rel}([G, \mathsf{Set}])|$, the pair $\big((\phi_o \alpha_o)\mathbf{A}aA,\ (\phi_o \alpha_o)\mathbf{A}aB\big)$ is in $S_r(\mathsf{Eq}\,\mathbf{A}, R)$. To this end notice that,

$$\begin{aligned}
\big((\phi_o \alpha_o)\mathbf{A}aA, (\phi_o \alpha_o)\mathbf{A}aB\big) &= \big(\alpha_o(\mathbf{A}, A)a, \alpha_o(\mathbf{A}, B)a\big) && \text{by the definition of } \phi_o, \\
&= \alpha_r(\mathsf{Eq}\,\mathbf{A}, R)(a, a) && \text{by the definition of } \alpha_r.
\end{aligned}$$

Hence, since $(a, a) \in \mathsf{Eq}\,(T_o\mathbf{A})$, and $\mathsf{Eq}\,(T_o\mathbf{A}) = T_r(\mathsf{Eq}\,\mathbf{A}) = T_r\pi_r(\mathsf{Eq}\,\mathbf{A}, R)$, we have that $\alpha_r(\mathsf{Eq}\,\mathbf{A}, R)(a, a)$ is in $S_r(\mathsf{Eq}\,\mathbf{A}, R)$, and therefore $\big((\phi_o\alpha_o)\mathbf{A}aA, (\phi_o\alpha_o)\mathbf{A}aB\big)$ is in $S_r(\mathsf{Eq}\,\mathbf{A}, R)$, as required. An analogous argument shows that $(\phi_r\alpha_r)\mathbf{R}(a, b)$ is actually an element of $(\forall_r S_r)\mathbf{R}$.

For the definition of ψ, let $(\beta_r,\ \beta_o \times \beta_o)$ be any fibred natural transformation in $\mathcal{C}_n(T, \forall_n S)$, let $(\mathbf{A}, A)$ and $(\mathbf{B}, B)$ be $(n+1)$-tuples of G-sets in $|[G, \mathsf{Set}]|^{n+1}$, and let $(\mathbf{R}, R)$ be an $(n+1)$-tuple of relations in $|\mathsf{Rel}([G, \mathsf{Set}])|^{n+1}$ on $(\mathbf{A}, A) \times (\mathbf{B}, B)$. Then we define ψ to be the pair $(\psi_r, \psi_o \times \psi_o) : \mathcal{C}_n(T, \forall_n S) \to \mathcal{C}_{n+1}(T\pi_n, S)$ given by $(\psi_o\beta_0)(\mathbf{A}, A) = \lambda a.(\beta_o\mathbf{A}aA)$ and $(\psi_r\beta_r)(\mathbf{R}, R) = (\psi_r\beta_r)(\mathbf{A}, A) \times (\psi_r\beta_r)(\mathbf{B}, B)$. To show that $(\psi_o\beta_o)(\mathbf{A}, A)$ and $(\psi_r\beta_r(\mathbf{R}, R)$ are well-defined follows a similar line of arguments to above.

To complete the proof one should then show that ϕ and ψ are fibred natural transformations, that they are mutually inverse, and that the isomorphism is natural in T and S. $\square$

We have managed to show that the types of System F can be interpreted using the relations fibration on $[G, \mathsf{Set}]$ as equality preserving fibred functors. We will not explicitly state the interpretation of terms, but a very similar process to Section 3.2.2 can be followed to yield the Abstraction Theorem for G-sets.

Theorem 75 (Abstraction Theorem). *In the relational G-set model every term defines a fibred natural transformation* $([\![\Delta; \Gamma \vdash t : T]\!]_r,\ [\![\Delta; \Gamma \vdash t : T]\!]_o \times [\![\Delta; \Gamma \vdash t : T]\!]_o) : [\![\Delta \vdash \Gamma]\!] \to [\![\Delta \vdash T\ \mathsf{Type}]\!]$.

Just like in Chapter 3 we also have that the Graph Lemma holds (using almost an identical proof), which we will need later.

Theorem 76 (Graph Lemma for G-Sets). *Suppose* $(T_1, T_0 \times T_0) : \mathsf{Rel}(p) \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)$ *is an equality preserving fibred functor. Then for any equivariant function* $f : A \to B$ *we have* $T_1\langle f\rangle = \langle T_0 f\rangle$.

4.4 INITIAL ALGEBRAS FOR SYSTEM F, AGAIN

So far, we have followed a similar approach to Chapter 3 by providing a model for System F only this time using G-sets. To continue along the same path, we might also wonder whether it is possible to prove an initial algebra theorem analogous to Theorem 61 for the G-set model, and indeed, it is. We prove this main result in Theorem 85, but we will find out that there are some crucial differences. We find the first difference by tackling the question: how many functions are there of $\forall X.X \to X$ in the G-set model?

4.4.1 THE G-SET INTERPRETATION OF ∀X.X→X

Consider the System F type $\forall X.X \to X$. In Reynolds's relational semantics this contains only the parametric identity function $\Lambda X.\lambda x.x$ and so we are able to show that there is an isomorphism $[\![\forall X.X \to X]\!]_o \cong 1$ (see Corollary 42). But in the world of G-sets, something slightly different happens.

Let's have a look at the interpretation of $\forall X.X \to X$ using the relations fibration on $[G, \mathsf{Set}]$. Since $\forall X.X \to X$ is a closed type, the interpretation $[\![\forall X.X \to X]\!]_o$ is an equality-preserving fibred functor $|[G, \mathsf{Set}]|^0 \xrightarrow[\mathsf{Eq}]{} [G, \mathsf{Set}]$, i.e., a G-set with the carrier $[\![\forall X.X \to X]\!]_o$ given by

$$\{f : \prod_{A \in [G,\mathsf{Set}]} [\![X \vdash X \to X]\!]_o A \mid \forall R \subseteq A \times B,\ (f_A, f_B) \in [\![X \vdash X \to X]\!]_r R\}.$$

Since X is a variable, for any G-set A, $[\![X \vdash X \to X]\!]_o A = A \Rightarrow A$, and so we can rewrite the above as

$$[\![\forall X.X \to X]\!]_o = \{f : \prod_{A \in [G,\mathsf{Set}]} A \Rightarrow A \mid \forall R \subseteq A \times B,\ (f_A, f_B) \in R \Rightarrow R\}.$$

Suppose that π is an element of the group G. It seems like we can define a parametric function by $f_A a = \pi \cdot_A a$, where a is an element of a G-set A. This would be parametric in the sense that it could be uniformly defined for every G-set A. But this would mean that we would have a quite different interpretation to the standard interpretation, since there is no longer only one parametric function in $[\![\forall X.X \to X]\!]_o$, but instead one for each element of G. Before we jump ahead of ourselves we need to ask a simple question: is the function f actually an element of $[\![\forall X.X \to X]\!]_o$?

Theorem 77. *Suppose that π is an element of the group G. Then the parametric function f defined for any G-set A by $f_A a = \pi \cdot_A a$ is an element of $[\![\forall X.X \to X]\!]_o$.*

Proof. We need to show that f satisfies the uniformity condition of $[\![\forall X.X \to X]\!]_o$, in other words that that for any equivariant relation R on $A \times B$ that (f_A, f_B) is related in $R \Rightarrow R$. To this end, let (a, b) be a pair of related elements in R. Then to show that $(f_A a, f_B b)$ is related in R, we must show that $(\pi \cdot_A a, \pi \cdot_B b)$ is related in R, which is true since $(\pi \cdot_A a, \pi \cdot_B b) = \pi \cdot_{A \times B} (a, b)$ and R is an equivariant relation. □

If we look at the interpretation of $\forall X.X \to X$ in $[G, \mathsf{Set}]$ we now know from the theorem above that it contains the identity function, as well as all of the functions of the form $f_A a = \pi \cdot_A a$. There are no other obvious functions that we might expect to inhabit $[\![\forall X.X \to X]\!]_o$, and so one might speculate that $[\![\forall X.X \to X]\!]_o \cong G$, and indeed it is (where G has an appropriate group action).

Theorem 78. *The closed type $\forall X.X \to X$ has the following interpretation in the G-set model*

$$[\![\forall X.X \to X]\!]_o \cong G_C.$$

Notice that in the statement of Theorem 78 we use the G-set G_C with the conjugate action. This action means that the bijective maps are actually isomorphisms (i.e. equivariant). Interestingly, to be able to appeal to parametricity we also have to use the G-set G_M to define a relation. It is this ability to be able to jump back and fourth between these two G-sets that allows the proof to go through, and so for the general initial algebras theorem for G-sets in Section 4.4.4 we will need to generalise this trick.

Proof. We show that there exists mutually inverse equivariant maps ϕ and ψ.

$$\begin{tikzcd}
\llbracket \forall X.X \to X \rrbracket_o \arrow[r, bend left, "\phi"] & G_C \arrow[l, bend left, "\psi"]
\end{tikzcd}$$

To begin, first note that for any G-set A, and any element a in A, we have an equivariant relation R_a on $G_M \times A$ defined by $R_a = \langle \_ \cdot_A a \rangle = \{(\pi, a') \mid a' = \pi \cdot_A a\}$, where $\_ \cdot_A a : G_M \to A$ denotes the function $(\_ \cdot_A a)\pi = \pi \cdot_A a$. To see that this relation is equivariant suppose that we have two related elements $(\pi, \pi \cdot_A a) \in R_a$, then for any element σ in G,

$$\begin{aligned}
\sigma \cdot_{G_M \times A} (\pi, \pi \cdot_A a) &= (\sigma \cdot_{G_M} \pi, \sigma \cdot_A (\pi \cdot_A a)) && \text{by definition of action,}\\
&= (\sigma\pi, (\sigma\pi) \cdot_A a) && \text{by compatibility and definition of action,}
\end{aligned}$$

which is clearly in the relation R_a as required. Notice here that for this relation to be equivariant we needed to use the G_M action.

Using this equivariant relation we can derive a quick parametricity result which we will need later on in the proof.

Parametricity: The pair (e, a) is related in R_a and hence for any parametric function f in $\llbracket \forall X.X \to X \rrbracket_o$ we have that $(f_{G_M} e, f_A a)$ is related in R_a. Or written equivalently as,

$$f_A a = (f_{G_M} e) \cdot_A a. \tag{4.3}$$

Defining the Isomorphism: Now we can define ϕ and ψ. We begin by defining the morphism $\phi : \llbracket \forall X.X \to X \rrbracket_o \to G_C$ by $\phi f = f_{G_M} e$, which is well-defined because an element of G_M is also an element of G_C by definition.[5] To see that ϕ is an equivariant map

[5]Recall that a morphism in $[G, \mathsf{Set}]$ is a set-valued function that is also equivariant. This means that we can construct ϕ in a non-equivariant way, i.e., as the composite of a (non-equivariant) function $\phi' : \llbracket \forall X.X \to X \rrbracket_o \to G_M$ with the (non-equivariant) inclusion $\iota : G_C \to G_M$, and then prove that this composition $\phi = \iota\phi'$ is equivariant. In the definition of ϕ above we do not make the use of the inclusion explicit.

we simply compute. Let π be an element of G_C, then,

$$\begin{aligned}
(\pi \cdot_\forall f)_{G_M} e &= (\pi \mathbin{\dot{\Rightarrow}} f_{G_M})e && \text{by definition of } \cdot_\forall, \\
&= \pi \cdot_{G_M} \left(f_{G_M} (\pi^{-1} \cdot_{G_M} e)\right) && \text{by definition of } \dot{\Rightarrow}, \\
&= \pi(f_{G_M})\pi^{-1} && \text{by definition of } \cdot_{G_M}.
\end{aligned}$$

Now by the instantiating Equation 4.3 from the parametricity argument above with $A = G_M$ and $a = \pi$, we have that $f_{G_M} \pi^{-1} = (f_{G_M} e)\pi^{-1}$. Hence, we can conclude that,

$$\begin{aligned}
(\pi \cdot_\forall f)_{G_M} e &= \pi(f_{G_M})\pi^{-1} && \text{by definition of } \cdot_\forall, \\
&= \pi(f_{G_M} e)\pi^{-1} && \text{by Equation 4.3}, \\
&= \pi \cdot_{G_C} (f_{G_M} e) && \text{by definition } \cdot_{G_C}, \\
&= \pi \cdot_{G_C} \phi(f) && \text{as required.}
\end{aligned}$$

We define the morphism $\psi : G_C \to [\![\forall X.X \to X]\!]_o$ by $\psi\pi = \Lambda A.\lambda a.\pi \cdot_A a$, which is well-defined by Theorem 77. To see that ψ is equivariant we again simply compute. Let π and σ be elements of G_C, then

$$\begin{aligned}
\sigma \cdot_\forall \psi(\pi) &= \sigma \cdot_\forall \left(\Lambda A.\lambda a.\pi \cdot_A a\right) && \text{by definition of } \psi, \\
&= \Lambda A.\sigma \mathbin{\dot{\Rightarrow}} (\lambda a.\pi \cdot_A a) && \text{by definition of } \cdot_\forall, \\
&= \Lambda A.\lambda a.\sigma \cdot_A \pi \cdot_A \sigma^{-1} \cdot_A a && \text{by definition of } \dot{\Rightarrow}, \\
&= \Lambda A.\lambda a.(\sigma \cdot_{G_C} \pi) \cdot_A a && \text{by compatibility axiom,} \\
&= \psi(\sigma \cdot_{G_C} \pi) && \text{as required.}
\end{aligned}$$

Notice here that we really do need the right-hand side of the isomorphism to be G_C and not G_M, as otherwise ψ would not be equivariant.

Mutually Inverse: Finally, to complete the proof we must show that ϕ and ψ are mutually inverse. Let's do the slightly more difficult direction first. Let f be a parametric function in $[\![\forall X.X \to X]\!]_o$, then we need to show that $\psi\phi(f) = f$. The left-hand side of this equation is given by $\psi\phi(f) = \psi(f_{G_M} e) = \Lambda A.\lambda a.(f_{G_M} e) \cdot_A a$. Hence it suffices to show that $\Lambda A.\lambda a.(f_{G_M} e) \cdot_A a = f$, or in other words for any G-set A, and any element a of A, that $f_A a = (f_{G_M} e) \cdot_A a$, which is given by Equation 4.3.

In the reverse direction we just expand the definitions. For an element π of G_C, we have that $\phi\psi(\pi) = \phi(\Lambda A.\lambda a.\pi \cdot_A a) = \pi \cdot_{G_M} e = \pi$ as required. $\square$

It's worth taking a moment to look at the usual proof to see why $[\![\forall X.X \to X]\!]_o \not\cong 1$ in the G-set model. If we were to follow exactly the proof in the usual set based case, we would define for any a in a G-set A the relation $R_a = \{(a, *)\}$ on $A \times 1$. However, this relation is not an *equivariant* relation, and hence cannot be used to deduce a parametricity property.

More abstractly, the G-set relational model does not satisfy the general initial algebra theorem from [2] since the category of G-sets $[G, \mathsf{Set}]$ is not well-pointed.

4.4.2 THE BEGINNINGS OF A GENERALISATION

The previous section described a clear difference between a parametricity result in the usual Reynolds-style relational semantics of System F and the relational semantics given by G-sets. Usually the type $\forall X.X \to X$ is interpreted as the terminal object, but instead in the world of G-sets it is interpreted as the conjugate G-set G_C. One then starts to wonder if this is a general phenomenon. Normally a functor defined by the interpretation of a functorial type of the form $X \vdash T$ Type, where T is some functorial type expression, has an initial T-algebra with the carrier given by the interpretation of $\forall X.(TX \to X) \to X$, as shown in Theorem 61. We have already seen that in the G-sets model this fails, but this doesn't necessarily mean that a similar result does not hold.

Normally, initial algebras can be thought of as trees (for strictly positive functors), and $\forall X.X \to X$ gives a tree with only one node. But in the G-set model, when we looked at the interpretation of $\forall X.X \to X$ we saw that instead of containing only the parametric identity, we found a function for every element of G. In terms of trees, we have a single node that is labelled by elements of G. By imagining that a similar pattern might happen for other types, one might guess that the interpretation of $\forall X.(TX \to X) \to X$ in $[G, \mathsf{Set}]$ has the same "structure" as the regular Reynolds-style case, but can additionally store an element of G at each "node". It turns out that the interpretation of the System F type $\forall X.(TX \to X) \to X$ is the carrier of an initial algebra, just not the usual one. It is the carrier of an initial F_0-algebra, where F_0 denotes a functor $[G, \mathsf{Set}] \to [G, \mathsf{Set}]$ defined by $F_0 A = G_M \times T_0 A$, with an action that is analogous to the conjugate action. It takes a little bit of effort to actually prove this theorem, which we do in the rest of this section and then look at some examples in Section 4.4.5.

Since the proof of the initial algebra theorem for G-sets (Theorem 85) requires a reasonable amount of work we first give a high-level overview. Instead of showing that an initial algebra for the functor F_0 exists, the theorem shows that if an initial F_0-algebra exists, then it is given by the interpretation of $\forall X.(TX \to X) \to X$. This lightens the workload a little and allows us to focus on the most interesting part — how the result differs from Theorem 61.

In what follows, the initial F_0-algebra, which we denote $\mathsf{\mu F_0}$, plays the role of G_M in Section 4.4.1. To define the G-set that plays the equivalent role to G_C we have to define the action on $|\mathsf{\mu F_0}|$ in terms of the fold map. We denote this action $\cdot_Z : G \times |\mathsf{\mu F_0}| \to |\mathsf{\mu F_0}|$ and define it by

$$\pi \cdot_Z t = \pi \cdot_{\mathsf{\mu F_0}} \left(fold_{\mathsf{\mu F_0}} \left(\alpha\, in(\pi^{-1}, \_) \right) t \right).$$

It is non-trivial to show that this action satisfies the identity and compatibility axioms, and this task constitutes Theorem 81 and Theorem 82.

Once we have shown that $(|\mathsf{\mu F_0}|, \cdot_Z)$ is actually an action, we then show that this G-set is isomorphic to $[\![\forall X.(TX \to X) \to X]\!]_o$. We do this by first showing that for any h in $|\mathsf{\mu F_0}|$, the function f defined by

$$f_A h = fold_A\, \alpha h,$$

for any G-set A and any $h \in T_0 A \Rightarrow A$, is an element of $[\![\forall X.(TX \to X) \to X]\!]_o$. This is proven in Theorem 84, and we use this to prove the general G-set initial algebra result in Theorem 85.

The first step on our journey towards Theorem 85 is to prove two technical theorems, that show how to convert an element of the exponential into a morphism.

Theorem 79. *Suppose that A and B are G-sets, $f : A \to_{\mathsf{Set}} B$ is a function in* Set, *and define $\alpha f : G_M \times A \to_{\mathsf{Set}} B$ by $\alpha f(\pi, a) = \pi \cdot_B f(\pi^{-1} \cdot_A a)$. Then αf is an equivariant map $G_M \times A \to B$.*

Proof. Let A and B be G-sets and $f : A \to_{\mathsf{Set}} B$ be a function in Set. To see that αf is an equivariant map $G_M \times A \to B$ let (π, a) be a pair in $G_M \times A$ and let σ be an element of G_M, then,

$$\begin{aligned}
\alpha f(\sigma \cdot_{G_M \times A} (\pi, a)) &= \alpha f(\sigma \cdot_{G_M} \pi, \sigma \cdot_A a) && \text{by definition of action,}\\
&= (\sigma\pi) \cdot_B f((\sigma\pi)^{-1}\sigma \cdot_A a) && \text{by definition of } \alpha,\\
&= (\sigma\pi) \cdot_B f(\pi^{-1} \cdot_A a) && \text{by definition of inverse,}\\
&= \sigma \cdot_B \big(\pi \cdot_B f(\pi^{-1} \cdot_A a)\big) && \text{by compatibility axiom,}\\
&= \sigma \cdot_B (\alpha f(\pi, a)) && \text{by definition of } \alpha \text{, as required.}
\end{aligned}$$

□

Moreover, we can use a similar setup to obtain a relation homomorphism from an element of an exponential object in $\mathsf{Rel}([G, \mathsf{Set}])$.

Theorem 80. *Suppose that $R \subseteq A \times B$ and $R' \subseteq A' \times B'$ are equivariant relations and (h_A, h_B) is an element of $R \to R'$, then $(\alpha h_A, \alpha h_B)$ is a morphism $\mathsf{Eq}\, G_M \times R \to R'$ in $\mathsf{Rel}([G, \mathsf{Set}])$.*

Proof. To see that $(\alpha h_A, \alpha h_B)$ is a relation homomorphism we need to show that for any (π, π, a, b) related in $\mathsf{Eq}\, G_M \times R$ that $(\alpha h_A, \alpha h_B)(\pi, \pi, a, b)$ is related in R'. To this end,

$$\begin{aligned}
(\alpha h_A, \alpha h_B)(\pi, \pi, a, b) &= (\pi \cdot_A h_A(\pi^{-1} \cdot_A a), \pi \cdot_B h_B(\pi^{-1} \cdot_B b)) && \text{by definition of } \alpha,\\
&= \pi \cdot_{A\times B} (h_A, h_B)(\pi^{-1} \cdot_{A\times B} (a, b)) && \text{by definition of } \pi\cdot_{A\times B}.
\end{aligned}$$

Then since $(a, b) \in R$ we have that $\pi^{-1} \cdot_{A\times B} (a, b) \in R$ because R is an equivariant relation, and hence $(h_A, h_B)(\pi^{-1} \cdot_{A\times B} (a, b)) \in R'$ as (h_A, h_B) is a relation homomorphism, and therefore $\pi \cdot_{A\times B} (h_A, h_B)(\pi^{-1} \cdot_{A\times B} (a, b)) \in R'$ as R' is an equivariant relation.

All that remains is to show that $(\alpha h_{A}, \alpha h_{B})$ is equivariant, which is trivial since both αh_{A} and αh_{B} are equivariant and the action is given componentwise. □

4.4.3 DEFINING THE INITIAL ALGEBRA ACTION

In Section 4.4.1 we saw that we have to use both the conjugation action and the multiplication action to prove that $[\![\forall X.X \to X]\!]_{o} \cong G_{C}$. These actions were simple to describe in terms of each other ($\sigma \cdot_{G_C} \pi = \sigma \cdot_{G_M} \pi \cdot_{G_M} \sigma^{-1}$), and this made it easy to use them both in the proof of Theorem 78. However, when the data type is a bit more complicated, this makes defining two actions a bit more complicated too. In Theorem 82 we describe this action, and prove Theorem 81 to help show that it satisfies the compatibility axiom.

Theorem 81. *Suppose that $T_0 : [G, \mathsf{Set}] \to [G, \mathsf{Set}]$ is an endofunctor, $F_0 : [G, \mathsf{Set}] \to [G, \mathsf{Set}]$ is an endofunctor defined by $F_0 A = G_M \times T_0 A$, and that there exists an initial F_0-algebra $\mu\mathsf{F}_0$. Let π be an element of the group G and κ_π be the F_0-algebra $\alpha\, in(\pi^{-1}, \_) : G_M \times T_0\mu\mathsf{F}_0 \to \mu\mathsf{F}_0$. Then for any function h in the exponential $T_0 A \Rightarrow A$ we have that,*

$$(fold_{A}\ \alpha h)(fold_{\mu\mathsf{F}_0}\ \kappa_\pi) = fold_{A}\ \alpha(\pi^{-1} \mathbin{\dot{\Rightarrow}} h). \tag{4.4}$$

Proof. This proof relies on several uses of initiality. Firstly, since κ_π is an F_0-algebra, the left-hand side of the following diagram commutes.

$$\begin{tikzcd}
G_M \times T_0\mu\mathsf{F}_0 \arrow[r, "\mathrm{id} \times T_0 fold_{\mu\mathsf{F}_0}\,\kappa_\pi"] \arrow[d, "in"'] & G_M \times T_0\mu\mathsf{F}_0 \arrow[r, "\mathrm{id} \times F_0 fold_{A}\,\alpha h"] \arrow[d, "\kappa_\pi"] & G_M \times F_0 A \arrow[d, "\alpha(\pi^{-1} \dot{\Rightarrow} h)"] \\
\mu\mathsf{F}_0 \arrow[r, "fold_{\mu\mathsf{F}_0}\,\kappa_\pi"'] & \mu\mathsf{F}_0 \arrow[r, "fold_{A}\,\alpha h"'] & A
\end{tikzcd} \tag{4.5}$$

If we could show that the right-hand side also commutes then we could appeal to initiality to get the final result. In other words, we need to show that for any (σ, t) in $G_M \times T_0\mu\mathsf{F}_0$,

$$(fold_{A}\ \alpha h)(\kappa_\pi(\sigma, t)) = (\alpha(\pi^{-1} \mathbin{\dot{\Rightarrow}} h))(\mathrm{id} \times T_0 fold_{A}\ \alpha h)(\sigma, t), \tag{4.6}$$

which we do by a brute force calculation. First, let's expand the left-hand side of Equation 4.6.

$$\begin{aligned}(fold_A\ \alpha h)(\kappa_\pi(\sigma, t)) &= (fold_A\ \alpha h)(\sigma \cdot_{\mathsf{\mu F_0}} in(\pi^{-1}, \sigma^{-1} \cdot_{T_0\mathsf{\mu F_0}} t)) && \text{by definition of } \kappa_\pi,\\ &= (fold_A\ \alpha h)(in(\sigma\pi^{-1}, t)) && \text{by equivariance of } in.\end{aligned}$$

Similarly, we expand the right-hand side of Equation 4.6.

$$\begin{aligned}&(\alpha(\pi^{-1} \dot{\Rightarrow} h))(\mathrm{id} \times T_0 fold_A\ \alpha h)(\sigma, t)\\ &\quad= (\alpha(\pi^{-1} \dot{\Rightarrow} h))(\sigma, (T_0 fold_A\ \alpha h)t) && \text{by definition of } \times,\\ &\quad= \sigma \cdot_A (\pi^{-1} \dot{\Rightarrow} h)(\sigma^{-1} \cdot_{T_0 A} T_0(fold_A\ \alpha h)t) && \text{by definition of } \alpha,\\ &\quad= \sigma\pi^{-1} \cdot_A h(\pi\sigma^{-1} \cdot_{T_0 A} (T_0 fold_A\ \alpha h)t) && \text{by definition of } \dot{\Rightarrow},\\ &\quad= \alpha h(\sigma\pi^{-1}, T_0 fold_A\ \alpha h t) && \text{by definition of } \alpha,\\ &\quad= \alpha h(\mathrm{id} \times T_0 fold_A\ \alpha h)(\sigma\pi^{-1}, t) && \text{by definition of } \times.\end{aligned}$$

Hence, proving that Equation 4.6 holds is equivalent to showing that

$$(fold_A\ \alpha h)(in(\sigma\pi^{-1}, t)) = \alpha h(\mathrm{id} \times F_0 fold_A\ \alpha h)(\sigma\pi^{-1}, t),$$

which is true since αh is an F_0-algebra, and $fold_A\ \alpha h$ is an F_0-algebra homomorphism, i.e., $(fold_A\ \alpha h)in = \alpha h(id \times F_0 fold_A\ \alpha h)$. Therefore by uniqueness of the mediating morphism $fold_A\ \alpha(\pi^{-1} \cdot h) : \mathsf{\mu F_0} \to A$ we have that $(fold_A\ \alpha h)(fold_{\mathsf{\mu F_0}}\ \kappa_\pi) = fold_A\ \alpha(\pi^{-1} \cdot h)$ as required. □

Now using Theorem 81 we can define a generalised version of the conjugate action.

Theorem 82. *Suppose that $F_0, T_0 : [G, \mathsf{Set}] \to [G, \mathsf{Set}]$ are both endofuctors, with $F_0 A = G_M \times T_0 A$, and that there exists an initial F_0-algebra $\mathsf{\mu F_0}$. Then we can define a G-set Z that has the same underlying set as $\mathsf{\mu F_0}$, but has it's action given by*

$$\pi \cdot_Z t = \pi \cdot_{\mathsf{\mu F_0}} (fold_{\mathsf{\mu F_0}}\ \kappa_\pi)t.$$

Proof. To see that $\cdot_Z$ does in fact define a group action we must show that it satisfies the identity and compatibility axioms.

Identity: For any $t \in Z$, we must show that $e \cdot_Z t = t$, which we do by a simple computation. First note that $\kappa_e = in$, since for any π in G and t in $T_0 Z$, we have that

$$\kappa_e(\pi, t) = \alpha\, in(e^{-1}, \_)(\pi, t) = \pi \cdot_{F_0 \mu F_0} in(e, \pi^{-1} \cdot_{\mu G} t) = in(\pi, t),$$

by the equivariance of in. Hence, the identity axiom is satisfied as follows.

$$e \cdot_Z t = e \cdot_{\mu F_0} (fold_{\mu F_0}\, \kappa_e)t = (fold_{\mu F_0}\, in)t = t \text{ by Theorem 53.}$$

Compatibility: For any element t of Z and any two elements σ and π of the group G, we must show that $\sigma \cdot_Z (\pi \cdot_Z t) = (\sigma\pi) \cdot_Z t$.

$$\begin{aligned}
\sigma \cdot_Z (\pi \cdot_Z t) &= \sigma \cdot_{\mu F_0} (fold_{\mu F_0}\, \kappa_\sigma)(\pi \cdot_{\mu F_0} (fold_{\mu F_0}\, \kappa_\pi)t) && \text{by definition of } \cdot_Z, \\
&= \sigma \cdot_{\mu F_0} \pi \cdot_{\mu F_0} (fold_{\mu F_0}\, \kappa_\sigma)(fold_{\mu F_0}\, \kappa_\pi t) && \text{by equivariance of } fold_{\mu F_0}\, \kappa_\pi, \\
&= (\sigma\pi) \cdot_{\mu F_0} (fold_{\mu F_0}\, \alpha(\pi^{-1} \dot{\Rightarrow} in(\sigma^{-1}, \_))t && \text{by Theorem 81,}^{6} \\
&= (\sigma\pi) \cdot_{\mu F_0} (fold_{\mu F_0}\, \alpha(\pi^{-1} \cdot_{\mu F_0} in(\sigma^{-1}, \pi \cdot_{T_0 \mu F_0} \_))t && \text{by definition of } \dot{\Rightarrow}, \\
&= (\sigma\pi) \cdot_{\mu F_0} (fold_{\mu F_0}\, \alpha(in(\pi^{-1}\sigma^{-1}, \_))t && \text{by equivariance of } in, \\
&= (\sigma\pi) \cdot_{\mu F_0} (fold_{\mu F_0}\, \kappa_{\sigma\pi})t && \text{by definition of } \kappa_{\sigma\pi}, \\
&= (\sigma\pi) \cdot_Z t && \text{as required.}
\end{aligned}$$

□

4.4.4 INITIAL ALGEBRA THEOREM FOR G-SETS

We are nearly ready to prove the general initial algebra theorem, but first we need to prove the following helpful result.

[6] By Theorem 81 with $h = in(\sigma^{-1}, \_)$.

Theorem 83. *Suppose that* $(F_1, F_0 \times F_0) : \mathsf{Rel}(p) \xrightarrow[\mathsf{Eq}]{} \mathsf{Rel}(p)$ *is an equality preserving fibred functor with initial* F_0 *and* F_1*-algebras given by* $(\mu\mathsf{F}_0, in_0)$ *and* $(\mu\mathsf{F}_1, in_1)$*, such that* $\mu\mathsf{F}_1 = \mathsf{Eq}\,\mu\mathsf{F}_0$ *and* $in_1 = (in_0, in_0)$*. Then for any equivariant relation* $R \subseteq A \times B$ *and any* F_1*-algebra* $(k, k') : F_1 R \to R$*, we have* $fold_R\,(k, k') = (fold_A\,k, fold_B\,k')$*.*

Proof. By applying the forgetful functor $p : \mathsf{Rel}([G, \mathsf{Set}]) \to [G, \mathsf{Set}] \times [G, \mathsf{Set}]$ and then the first projection functor $\pi_1 : [G, \mathsf{Set}] \times [G, \mathsf{Set}] \to [G, \mathsf{Set}]$, we have a morphism $\pi_1 \circ p(fold_R\,(k, k')) : \mu\mathsf{F}_0 \to A$, which makes the following diagram commute.

$$\begin{tikzcd}
F_0\mu\mathsf{F}_0 \arrow[r, "F_0\left(\pi_1 \circ p(fold_R\,(k,k'))\right)"] \arrow[d, "in_0"'] & F_0 A \arrow[d, "k"] \\
\mu\mathsf{F}_0 \arrow[r, "\pi_1 \circ p(fold_R\,(k,k'))"'] & A
\end{tikzcd}$$

Hence, by initiality of $\mu\mathsf{F}_0$ we have that $\pi_1 \circ p(fold_R\,(k, k')) = fold_A\,k$. By applying the same argument with the second projection we see that $\pi_2 p(fold_R\,(k, k')) = fold_B\,k'$, and hence $fold_R\,(k, k') = (fold_A\,k, fold_B\,k')$ as required. □

We can now prove the general initial algebra theorem, which we do in two parts analogous to Theorem 77 and Theorem 78.

Theorem 84. *Let* $X \vdash T$ Type *be a functorial type expression and denote* $[\![\Delta \vdash T\ \mathsf{Type}]\!]_o$ *by* T_0*, denote* $[\![\Delta \vdash T\ \mathsf{Type}]\!]_r$ *by* T_1*, denote* $[\![\forall X.(TX \to X) \to X]\!]_o$ *by* Z_0*, and denote* $[\![\forall X.(TX \to X) \to X]\!]_r$ *by* Z_1*. Let* $(F_1, F_0 \times F_0) : \mathsf{Rel}(p) \to \mathsf{Rel}(p)$ *denote the fibred fuctor defined by* $F_0 A = G_M \times T_0 A$ *and* $F_1 R = \mathsf{Eq}\,G_M \times T_1 R$*.*

Suppose that there exists initial F_0 *and* F_1*-algebras* $(\mu\mathsf{F}_0, in_0)$ *and* $(\mu\mathsf{F}_1, in_1)$*, such that* $\mu\mathsf{F}_1 = \mathsf{Eq}\,\mu\mathsf{F}_0$ *and* $in_1 = (in_0, in_0)$*. Then for any* t *in* $\mu\mathsf{F}_0$*, any* G*-set* A *and any element* h *of the exponential* $T_0 A \Rightarrow A$*, the parametric function* ρ *defined by*

$$\rho_A h = (fold_A\,\alpha h)t,$$

is an element of $[\![\forall X.(TX \to X) \to X]\!]_o$*.*

Proof. First note that by Theorem 80 for any equivariant relation $R \subseteq A \times B$ and any element (h, h') of the exponential $T_1 R \Rightarrow R$, we have a morphism $(\alpha h, \alpha h') : F_1 R \to R$.

Then to see that ρ is an element of $[\![\forall X.(TX \to X) \to X]\!]_o$ we need to show that for any equivariant relation $R \subseteq A \times B$ and any element of the exponential $(h, h') \in T_1 R \Rightarrow R$, that the pair $(\rho_A h, \rho_B h')$ is in R, i.e., that $\big((fold_A\, \alpha h)t, (fold_A\, \alpha h')t\big) \in R$. Firstly, we have a morphism $fold_R\, (\alpha h, \alpha h') : \mathsf{\mu F}_1 \to R$ since $(\alpha h, \alpha h') : F_1 R \to R$ is an F_1-algebra and $\mathsf{\mu F}_1$ is weakly initial. Then since $\mathsf{\mu F}_1 = \mathsf{Eq}\, \mathsf{\mu F}_0$ and $(t, t) \in \mathsf{Eq}\, \mathsf{\mu F}_0$, we have that $fold_R\, (\alpha h, \alpha h')(t, t) \in R$, which since $fold_R\, (\alpha h, \alpha h')(t, t) = \big((fold_A\, \alpha h)t, (fold_B\, \alpha h')t'\big)$ by Theorem 83, proves the result. □

We can now prove the general initial algebra theorem for G-sets.

Theorem 85. *Suppose that we have the same set up as Theorem 84. Further, let Z denote the G-set with carrier $|Z| = \mathsf{\mu F}_0$ and with the action given by $\pi \cdot_Z t = \pi \cdot_{\mathsf{\mu F}_0} (fold_{\mathsf{\mu F}_0}\, \kappa_\pi)t$.*[7]

Then there is an isomorphism $[\![\forall X.(TX \to X) \to X]\!]_o \cong Z$.

Proof. Similar to the proof of Theorem 78, we start by looking at parametricity.

Parametricity: For any parametric function f in $[\![\forall X.(TX \to X) \to X]\!]_o$ we can deduce the following parametricity property. Let A be a G-set, and h an element of the exponential $T_0 A \Rightarrow A$. Then by initiality of $\mathsf{\mu F}_0$ the following diagram commutes.

$$\begin{tikzcd}
G_M \times T_0 \mathsf{\mu F}_0 \arrow[r, "\mathrm{id} \times T_0 fold_A\, \alpha h"] \arrow[d, "in_0"'] & G_M \times T_0 A \arrow[d, "\alpha h"] \\
\mathsf{\mu F} \arrow[r, "fold_A\, \alpha h"'] & A
\end{tikzcd} \tag{4.7}$$

Further, since $fold_A\, \alpha h : \mathsf{\mu F}_0 \to A$ is an equivariant map, by Lemma 71 its graph $\langle fold_A\, \alpha h\rangle$ is an equivariant relation on $\mathsf{\mu F}_0 \times A$. Hence, by the uniformity condition on f, we have that $(f_{\mathsf{\mu F}_0}, f_A) \in (T_1 \langle fold_A\, \alpha h\rangle \Rightarrow \langle fold_A\, \alpha h\rangle) \Rightarrow \langle fold_A\, \alpha h\rangle$. By instantiating Diagram 4.7

[7]Which actually defines an action by Theorem 82

at e, and noting that $\alpha h(e, \_) = h$, we have that,

$$\begin{tikzcd}
T_0\mathsf{\mu F}_0 \arrow[r, "T_0 fold_A\, \alpha h"] \arrow[d, "in_0(e,\, \_)"'] & T_0 A \arrow[d, "h"] \\
\mathsf{\mu F}_0 \arrow[r, "fold_A\, \alpha h"'] & A
\end{tikzcd}$$

commutes in Set. Hence, by Theorem 72 we have that $(in_0(e, \_), h)$ is in the exponential $(T_1\langle fold_A\, \alpha h\rangle \Rightarrow \langle fold_A\, \alpha h\rangle)$. Therefore, putting this all together we see that $(f_{\mathsf{\mu F}_0} in_0(e, \_), f_A h) \in \langle fold_A\, \alpha h\rangle$, or in other symbols,

$$f_A h = (fold_A\, \alpha h)(f_{\mathsf{\mu F}_0} in_0(e, \_)). \tag{4.8}$$

Defining the Isomorphism: We define the morphism $\phi : [\![\forall X.(TX \to X) \to X]\!]_o \to Z$ by $\phi f = f_{\mathsf{\mu F}_0} in_0(e, \_)$, which is well defined since $|\mathsf{\mu F}_0| = Z$, and the morphism $\psi : Z \to [\![\forall X.(TX \to X) \to X]\!]_o$ by $\psi t = \Lambda A.\lambda h : T_0 A \Rightarrow A.(fold_A\, \alpha h)t$, which is well-defined by Theorem 84. To see that ϕ is equivariant let π be in G_M, then,

$$\begin{aligned}
\phi(\pi \cdot_{\forall} f) &= (\pi \cdot_{\forall} f)_{\mathsf{\mu F}_0} in(e, \_) && \text{by the definition of } \phi, \\
&= \pi \cdot_{\mathsf{\mu F}_0} (f_{\mathsf{\mu F}_0} in_0(\pi^{-1}, \_)) && \text{by the definition of } \cdot_{\forall}, \\
&= \pi \cdot_{\mathsf{\mu F}_0} (fold_{\mathsf{\mu F}_0}\, \alpha in_0(\pi^{-1}, \_))(f_{\mathsf{\mu F}} in_0(e, \_)) && \text{by Equation 4.8.} \\
&= \pi \cdot_{\mathsf{\mu F}_0} (fold_{\mathsf{\mu F}_0}\, \kappa_\pi)(\phi f) && \text{by the definition of } \kappa_\pi \text{ and } \phi, \\
&= \pi \cdot_Z (\phi f) && \text{by the definition of } \cdot_Z \text{, as required.}
\end{aligned}$$

Similarly, to see that ψ is equivariant, again let π be an element of G_M, then

$$\begin{aligned}
\psi(\pi \cdot_Z t) &= \Lambda A.\lambda h.\, (fold_A\, \alpha h)(\pi \cdot_Z t) && \text{by the definition of } \psi, \\
&= \Lambda A.\lambda h.\, (fold_A\, \alpha h)(\pi \cdot_{\mu\mathsf{F}_0} fold_{\mu\mathsf{F}_0}\, \kappa_\pi)t && \text{by the definition of } \cdot_Z\,, \\
&= \Lambda A.\lambda h.\, \pi \cdot_A (fold_A\, \alpha h)(fold_{\mu\mathsf{F}_0}\, \kappa_\pi)t && \text{by equivariance of } fold. \\
&= \Lambda A.\lambda h.\, \pi \cdot_A (fold_A\, \alpha(\pi^{-1} \cdot_{\Rightarrow} h)t) && \text{by Theorem 81 .} \\
&= \pi \cdot_\forall (\Lambda A.\lambda h.\, (fold_A\, \alpha h)t) && \text{by the definition of } \cdot_\forall\,, \\
&= \pi \cdot_\forall \psi(t) && \text{as required.}
\end{aligned}$$

Mutual Inverses: To see that the composition $\psi\phi$ gives the identity, let f be a parametric function in $[\![\forall X.(TX \to X) \to X]\!]_o$, then we have that

$$\psi\phi f = \Lambda A.\lambda h.\, (fold_A\, \alpha h)(f_{\mu\mathsf{F}_0} in_0(e, \_)) = f \text{ by Equation 4.8 .}$$

For the reverse direction, suppose that $t \in Z$, then since $\kappa_e = in_0$, we have that

$$\phi\psi t = (fold_{\mu\mathsf{F}_0}\, \alpha in_0(e, \_))t = (fold_{\mu\mathsf{F}_0}\, in_0)t = t \text{ by initiality.}$$

□

Notice that Theorem 61 is a generalisation of Theorem 85 from Chapter 3. If the group G is the trivial group 1 then we recover the original theorem.

4.4.5 INITIAL ALGEBRA THEOREM IN ACTION

Let's look at a few examples of Theorem 85 in action.

Example 86. *Let's revisit the example that we looked at in Section 4.4.1. Suppose that* $TX = 1$ *and hence* $F_0 A = 1 \times G_M$*. Then* F_0 *has an initial algebra given by* $\mu\mathsf{F}_0 = G_M$ *and the morphisms* $in : G_M \times 1 \to G_M$ *and* $fold_A\, k : G_M \to A$ *are given by* $in(\pi, \star) = \pi$ *and* $fold_A\, k\pi = k(\pi, \star)$ *respectively.*

By Theorem 85 the interpretation of $\forall X.(1 \to A) \to A$ *is given by* $(G_M, \cdot_Z)$, *where the action* $\cdot_Z$ *is given by*

$$\begin{aligned}
\pi \cdot_Z \sigma &= \pi \cdot_{G_M} (fold_{G_M}\ \kappa_\pi \sigma) && \text{by Theorem 85,}\\
&= \pi \cdot_{G_M} \big(\kappa_\pi(\sigma, \star)\big) && \text{by the definition of } fold,\\
&= \pi \cdot_{G_M} \big((\alpha\, in(\pi^{-1}, \_))(\sigma, \star)\big) && \text{by the definition of } \kappa_\pi,\\
&= \pi \cdot_{G_M} (\sigma \cdot_{G_M} in(\pi^{-1}, \sigma^{-1} \cdot_1 \star) && \text{by the definition of } \alpha,\\
&= \pi \cdot_{G_M} in(\sigma \cdot_{G_M} \pi^{-1}, \star) && \text{by equivariance of } in,\\
&= \pi \cdot_{G_M} \sigma \cdot_{G_M} \pi^{-1} && \text{by the definition of } in.
\end{aligned}$$

Hence, $\pi \cdot_Z \sigma = \pi \cdot_{G_C} \sigma$ *and so* $[\![\forall X.(1 \to X) \to X]\!]_o = (G_C, \cdot_{G_C})$ *as expected (i.e., agreeing with Theorem 78).*

Example 87. *Suppose that* $TX = 1+X$ *and hence* $F_0 A = (G_M \times 1) + (G_M \times A)$. *Then* F_0 *has an initial algebra given by* $\mathsf{List}^+(G_M)$.[8] *The morphism* $in : (G_M \times 1) + (G_M \times \mathsf{List}^+(G_M)) \to \mathsf{List}^+(G_M)$ *is given (inductively) by* $in(\sigma, \star) = [\sigma]$ *and* $in(\sigma, [\overline{\sigma}]) = [\sigma, \overline{\sigma}]$, *and the morphism* $fold_A\ k : \mathsf{List}^+(G_M) \to A$ *is given (again inductively) by* $fold\,[k_0, k_1][\sigma] = k_0(\sigma, \star)$ *and* $fold\,[k_0, k_1][\sigma_1, \dots \sigma_n] = k_1(\pi, fold\,[k_0, k_1][\sigma_1, \dots \sigma_{n-1}])$.[9]

By Theorem 85 we have that the interpretation of $\forall X.(1 + X \to X) \to X$ *is given by* $(\mathsf{List}^+(G_M), \cdot_Z)$, *where the action* $\cdot_Z$ *is given inductively as follows. Let* π *be an element of* G, *let* $[\sigma]$ *be a list of length one, an let* $[\sigma, \overline{\sigma}]$ *be a list of length* $n+1$.

$$\begin{aligned}
\pi \cdot_Z [\sigma] &= \pi \cdot_{\mathsf{L}G_M} (fold\, \kappa_\pi [\sigma]) && \text{by Theorem 85,}\\
&= \pi \cdot_{\mathsf{L}G_M} (\alpha\, in(\pi^{-1}, \_))(\sigma, \star) && \text{by the definition of } fold \text{ and } in,\\
&= \pi \cdot_{\mathsf{L}G_M} \big(\sigma \cdot in(\pi^{-1}, \sigma^{-1} \cdot \star)\big) && \text{by the definition of } \alpha,\\
&= \pi \cdot_{\mathsf{L}G_M} \big(in(\sigma \cdot_{G_M} \pi^{-1}, \star)\big) && \text{by equivariance of } in,\\
&= \pi \cdot_{\mathsf{L}G_M} [\sigma \cdot_{G_M} \pi^{-1}] && \text{by the definition of } in\\
&= [\pi \cdot_{G_M} \sigma \cdot_{G_M} \pi^{-1}] && \text{by the definition of } \cdot_{\mathsf{L}G_M}.
\end{aligned}$$

[8] For the set based case see Example 56.

[9] As in Example 56 we omit the subscript of $fold$.

To see the action $\cdot_Z$ on a list of length $n+1$, first notice that by simply expanding the definitions, we have that $(\alpha\, in(\pi^{-1}, \_))(\sigma, [\overline{\sigma}]) = [\sigma \cdot_{G_M} \pi^{-1}, \overline{\sigma}]$. Hence, we have,

$$\begin{aligned}
\pi \cdot_Z [\sigma, \overline{\sigma}] &= \pi \cdot_{\mathsf{L}G_M} (fold\, \kappa_\pi[\sigma, \overline{\sigma}]) && \text{by Theorem 85,}\\
&= \pi \cdot_{\mathsf{L}G_M} (\alpha\, in(\pi^{-1}, \_))(\sigma_n,\, fold\, \kappa_\pi[\overline{\sigma}]) && \text{by definition of } fold_{\mathsf{List}^+(G_M)}\, \kappa_\pi,\\
&= \pi \cdot_{\mathsf{L}G_M} [\sigma \cdot_{G_M} \pi^{-1},\, fold\, \kappa_\pi[\overline{\sigma}]] && \text{by definition of } \alpha \text{ and } in,\\
&= [\pi \cdot_{G_M} \sigma \cdot_{G_M} \pi^{-1},\, fold\, \kappa_\pi[\overline{\sigma}]] && \text{by definition of } \cdot_{\mathsf{L}G_M},\\
&= [\pi \cdot_{G_C} \sigma,\, fold\, \kappa_\pi[\overline{\sigma}]] && \text{by definition of } \cdot_{G_C}.
\end{aligned}$$

Therefore, putting this all together we have that $\pi \cdot_Z [\sigma_1, \ldots, \sigma_n] = [\pi \cdot_{G_C} \sigma_1, \ldots, \pi \cdot_{G_C} \sigma_n]$, and hence $[\![\forall.(1 \to X) \to X]\!]_o = (\mathsf{List}^+(G_C),\ \cdot_{\mathsf{L}G_C})$.

Example 88. *Suppose that $T_0X = 1 + (X \times X)$ and hence $F_0A = (G_M \times 1) + (G_M \times A \times A)$. Then following a similar line of arguments to Theorem 86 and Theorem 87, we have that the interpretation of $\forall X.(1 + (X \times X) \to X) \to X$ gives the collection of binary trees with an element of G_C stored at each node, and the group action on a binary tree is given by applying the group action of G_C at each node.*

4.5 NOMINAL DISCUSSION

In this chapter we have managed to show that the relations fibration on G-sets can be used to give a parametric model for System F. Instead of the usual initial algebras theorem we found that a different (but related) theorem holds. Where initial algebras normally give trees, in the world of group actions we found that each node additionally contained an element of G. This meant that the characterisation of the terminal object 1 became the G-set given by the group with the conjugate action G_C, and the characterisation of the natural numbers $\mathbb{N}$ became the collection of non-empty lists $\mathsf{List}^+(G_C)$ containing elements of G_C.

One might think that a natural progression of this work would be to look at nominal sets. Nominal sets are $\mathsf{Perm}\,\mathbb{A}$-sets that are finitely supported, where $\mathsf{Perm}\,\mathbb{A}$ denotes the set of finite bijections of a countably infinite set $\mathbb{A}$. The term *finite support* means that for an element x of a nominal set X, there is a set $A \subseteq \mathbb{A}$ such that, if for all $\pi \in \mathsf{Perm}\,\mathbb{A}$, we have $(\forall a \in A,\ \pi a = a) \implies \pi \cdot_X x = x$.

The restriction of $\mathsf{Perm}\,\mathbb{A}$-sets to finitely supported ones is a very useful one, and the study of nominal sets has been very fruitful (see e.g. [34] for an overview). However, it turns out that by requiring $\mathsf{Perm}\,\mathbb{A}$-sets to be finitely supported we make it a little more tricky to apply the results proven in this chapter.

The first immediate stumbling block is given when trying to define the $\mathsf{Perm}\,\mathbb{A}$-set with the multiplication action (the equivalent of G_M). This set is a $\mathsf{Perm}\,\mathbb{A}$-set but it is *not* finitely supported. To see this, let $\mathsf{Perm}\,\mathbb{A}$ denote the $\mathsf{Perm}\,\mathbb{A}$-set with the multiplication action $\pi \cdot_{\mathsf{Perm}\,\mathbb{A}} \sigma = \pi\sigma$ and consider the identity element $e \in \mathsf{Perm}\,\mathbb{A}$. If $A \subseteq \mathbb{A}$ is a finite support for e, then for any element $\pi \in \mathsf{Perm}\,\mathbb{A}$ such that $\forall a \in A,\ \pi a = a$

$$\pi \cdot_{\mathsf{Perm}\,\mathbb{A}} e = e.$$

Or in other words $\pi = e$, which is not true in general. Since A is finite we can always find two distinct elements a_1 and a_2 in $\mathbb{A} \setminus A$. Then the transposition $(a_1\ a_2)$ satisfies $\forall a \in A,\ (a_1\ a_2)\, a = (a_1\ a_2)$, but is clearly not equal to the identity element e. This discussion does not mean that an initial algebras theorem for nominal sets as model for System F can not be proven, just that it is not a direct generalisation of the work here.

Departing from System F in the next chapter we will look at types that are indexed by dimensions. We will see that G-sets will also play an important role and we will look at parametricity in this setting.

CHAPTER 5
DIMENSIONS

In this chapter we depart from looking at System F and instead focus on a new type system that has types for physical dimensions. We will use the fibrational approach to give a general categorical semantics for this type system and then explore parametricity in this context, where we will discover a deep connection with group actions.

This material in this chapter has been published in a joint paper [3].

5.1 MISMATCHING MISTAKES ON MARS

Humans make errors. In 1999 the Mars Climate Orbiter disintegrated in the Martian atmosphere due to a unit mismatch in the computer systems — one part used Newton-seconds and the other pound-seconds [35]. This mismatch was not be detected by the computer systems, instead it was assumed that the software had been correctly designed. Due to human error the software specification had not been followed exactly. For the Mars Climate Orbiter, this had a catastrophic result.

There has been a lot of work by many different people to try to avoid unit errors by developing programming languages that can manipulate physical quantities [36–40]. Often inspired by tales like the Mars Climate Orbiter, the idea is that if your programming language knows the units that you are using, it can check whether there is ever a mismatch.

In this chapter (and thesis), we're not interested in the specifics of the Mars Climate Orbiter, and we are also not interested in the specifics of these programming languages, but we are interested in their underlying structure. As part of our motivation, we hope that by exploring a general semantics, we can unravel intrinsic properties that will help to guide future development. Moreover, there is a natural fibrational approach in this situation due to the indexing nature of units, and the machinery that we have developed in Chapters 2–4 can provide additional insights.

Our starting point is the work of Andrew Kennedy. In his thesis [41] and subsequent papers [25, 39], he developed a type theory for programs that can manipulate physical quantities using dimension types, as well as a semantics based on complete partial orders. Kennedy also develops a relational semantics, and proves a parametricity result, which he uses to prove theorems about his type system.

We take a different approach by developing a general categorical notion of model for programming languages with dimension types. This allows us to explore different ways of building different models, as well using categorical tools to provide simple, elegant proofs of some relevant theorems.

Specifically, in this chapter we will define a simple type theory called λD, indexed by dimensions. From there we will introduce the semantics by defining the notion of a λD-*model* in Section 5.4.

In Section 5.6 we will explore an important example of a λD-model, which is built from group actions (Example 96), and we show that this model supports a diverse range of parametricity-like theorems, without the need to define a separate relational semantics. This results in simple proofs of theorems that would otherwise require more heavy machinery. Finally, in Section 5.7 we explore the relationship between the parametricity-like theorems of the λD-model built from group actions, and a natural notion of a relational model.

5.2 TYPES WITH PHYSICAL DIMENSIONS

Throughout science different physical quantities are organised into dimensions, such as length or time. For any equation to be meaningful it must have the same dimensions on both the left and the right-hand side of the equation. This provides a plausibility check on derived equations and results, which is a crucial tool in the physical sciences, and is central to the discipline known as *dimensional analysis.* As part of dimensional analysis, a fundamental principle is that it is not meaningful to add or compare quantities of different dimensions, but they can be multiplied.

To measure a physical quantity we use units, such as metres or seconds. We understand these units as chosen constant quantities of given dimensions, which can then be scaled to express different measurements. For example, a cricket pitch has a length that is twenty times as long as one metre or more succinctly, has a length of 20m.

To express dimensions in a programming language we introduce a type $\mathsf{Quantity}(X)$ of quantities with dimension X. We introduce this type more formally below, but let's first have a look at how it works. Here is a polymorphic program that is defined for all dimensions; it takes a quantity x of a given dimension X, and returns its double, which has the same dimension.

$$f := (\Lambda X.\, \lambda x : \mathsf{Quantity}(X).\, x + x) :\ \forall X.\, \mathsf{Quantity}(X) \to \mathsf{Quantity}(X) \tag{5.1}$$

Though the symbols "ΛX" and "$\forall X$" look like they belong to System F, here they mean something completely different. Both occurrences of "X" in "ΛX" and "$\forall X$" correspond to *dimension variables* and not *type variables.* This means that the term in Equation 5.1 cannot be instantiated at an arbitrary type, but only at an arbitrary dimension. This difference means that types no longer define functors, but instead simply objects in the total category of a fibration. We give the full details of this below, but to illustrate consider the following instantiation. Suppose we would like to know how long a cricket pitch would be if we doubled its length, and we would like to use the polymorphic function f to find out.

$$\begin{aligned} f_{Length}\,(20\text{m}) &= 20\text{m} + 20\text{m} : \mathsf{Quantity}(\mathit{Length}) \\ &= 40\text{m} : \mathsf{Quantity}(\mathit{Length}) \end{aligned}$$

Unsurprisingly, the total length is 40m, but there are a couple of key points that are worth emphasising about this example.

- There are two kinds of variable, X and x. The first variable X stands for a dimension whereas x stands for an inhabitant of a type. To emphasise this distinction, when we abstract each variable we will use a different symbol, λ for term abstraction and Λ for dimension abstraction, as seen in equation 5.1 above, and analogous to our use of λ and Λ for System F.
- The type $\mathsf{Quantity}(X)$ depends on a dimension X, and it is inhabited by quantities of that dimension. For example, the standard unit of measure for length, the metre, is a quantity of the length dimension, i.e. a constant m : $\mathsf{Quantity}(\mathit{Length})$, as well as any multiple of the standard unit such as 20m.

To express programs such the one above, we introduce a simple type theory, which we will call λD. Since there are two kinds of variables, we have two kinds of contexts.

Dimension Contexts and Dimension Expressions: A *dimension context* Δ is a finite list of distinct dimension variables $X_1 \ldots X_n$. A *dimension-expression-in-context* $\Delta \vdash D\ \mathsf{Dim}$ is a monomial D in the variables Δ. In other words, a dimension-expression-in-context is given by a collection of integers $k_i \in \mathbb{Z}$, and written as $\Delta \vdash X_1^{k_1} \ldots X_n^{k_n}\ \mathsf{Dim}$. The set of dimension-expressions-in-context $\{D \mid \Delta \vdash D\ \mathsf{Dim}\}$ is an Abelian group under addition of exponents, i.e., $X^k \cdot X^{k'} = X_i^{k+k'}$, and this is the free Abelian group on Δ. By the universal property of the free Abelian group, we have a substitution on dimension expressions, which is given by, for example, $X, Z \vdash (X^2Y^3)[^{(XZ^2)}/_{Y}] = X^5Z^6\ \mathsf{Dim}$.

Types: Well-formed types are given by judgements of the form $\Delta \vdash T\ \mathsf{Type}$, where Δ is a dimension context. The judgements are generated by the following rules.

λD TYPES

$\dfrac{\Delta \vdash D\ \mathsf{Dim}}{\Delta \vdash \mathsf{Quantity}(D)\ \mathsf{Type}}$	**DIMENSION TYPES**
$\dfrac{\Delta, X \vdash T\ \mathsf{Type}}{\Delta \vdash \forall X.T\ \mathsf{Type}}$	**DIMENSION POLYMORPHISM**
$\dfrac{\Delta \vdash T\ \mathsf{Type} \quad \Delta \vdash U\ \mathsf{Type}}{\Delta \vdash T \to U\ \mathsf{Type}}$	**ARROW TYPES**
$\dfrac{}{\Delta \vdash 1\ \mathsf{Type}}$	**UNIT TYPE**
$\dfrac{\Delta \vdash T\ \mathsf{Type} \quad \Delta \vdash U\ \mathsf{Type}}{\Delta \vdash T \times U\ \mathsf{Type}}$	**PRODUCT TYPES**
$\dfrac{}{\Delta \vdash 0\ \mathsf{Type}}$	**EMPTY TYPE**
$\dfrac{\Delta \vdash T\ \mathsf{Type} \quad \Delta \vdash U\ \mathsf{Type}}{\Delta \vdash T + U\ \mathsf{Type}}$	**SUM TYPES**

Notice that we do *not* have System-F-style polymorphism, but instead, we have polymorphism of dimensions. In other words, types can be parameterised by dimensions, but they cannot be parameterised by types, since we do no have type variables. This means that this type system behaves very differently to System F. Looking at this setup via the Curry-Howard correspondence, we have a first-order-logic where the domain of discourse is the theory of Abelian groups and a single atomic predicate, Quantity.

5.3 TERMS WITH PHYSICAL DIMENSIONS

As usual we will focus our attention on the types of this system, but for the interested reader we formally introduce the terms here.

Typing Contexts and Terms: Well-formed typing contexts are given by judgements $\Delta \vdash \Gamma\ \mathsf{Ctx}$, where Δ is a dimension context, Γ is of the form $x_1 : T_1, \ldots, x_n : T_n$, and there is a well-formed typing judgement $\Delta \vdash T_i\ \mathsf{Type}$ for every i. Well-formed terms are given by judgements $\Delta; \Gamma \vdash t : T$, where there is a well-formed typing context $\Delta \vdash \Gamma\ \mathsf{Ctx}$, and a well-formed type $\Delta \vdash T\ \mathsf{Type}$. The rules for the type formers $\mathsf{1}$, $\mathsf{0}$, $\_ \times \_$, $\_ + \_$, and $\_ \to \_$ are the usual ones from the Simply Typed λ-Calculus, as stated below.

λD TERMS

$$\frac{\Delta \vdash \Gamma, \Gamma'\ \mathsf{Ctx} \quad \Delta \vdash T\ \mathsf{Type}}{\Delta; \Gamma, x : T, \Gamma' \vdash x : T} \qquad \frac{\Delta; \Gamma, x : T \vdash t : U}{\Delta; \Gamma \vdash \lambda x.t : T \to U}$$

$$\frac{\Delta; \Gamma \vdash t : T \to U \quad \Delta; \Gamma \vdash u : T}{\Delta; \Gamma \vdash t\, u : U} \qquad \frac{\Delta \vdash \Gamma\ \mathsf{Ctx}}{\Delta; \Gamma \vdash () : \mathsf{1}}$$

$$\frac{\Delta; \Gamma \vdash t_1 : T_1 \qquad \Delta; \Gamma \vdash t_2 : T_2}{\Delta; \Gamma \vdash (t_1, t_2) : T_1 \times T_2} \qquad \frac{\Delta; \Gamma \vdash t : T_1 \times T_2}{\Delta; \Gamma \vdash \mathsf{pr}_i(t) : T_i}$$

$$\frac{\Delta; \Gamma \vdash t : \mathsf{0} \quad \Delta \vdash T\ \mathsf{Type}}{\Delta; \Gamma \vdash \mathsf{case}\, t : T} \qquad \frac{\Delta; \Gamma \vdash t : T_i}{\Delta; \Gamma \vdash \iota_i\, t : T_1 + T_2}$$

$$\frac{\Delta; \Gamma \vdash t : T_1 + T_2 \quad \big(\Delta; \Gamma, x_i : T_i \vdash u_i : U\big)_{i \in \{1,2\}}}{\Delta; \Gamma \vdash \mathsf{case}\, t\, \mathsf{of}\, \{\iota_1\, x_1 \mapsto u_1; \iota_2\, x_2 \mapsto u_2\} : U}$$

In addition, we can abstract over dimension variables, and substitute a dimension expression for an abstracted dimension variable, which gives the introduction and elimination rules for quantification over a dimension variable.

λD TERMS CNTD...

$$\frac{\Delta, X; \Gamma \vdash t : T}{\Delta; \Gamma \vdash \Lambda X.t : \forall X.T} \qquad \frac{\Delta \vdash D\ \mathsf{Dim} \quad \Delta; \Gamma \vdash t : \forall X.T}{\Delta; \Gamma \vdash t_D : T[D/X]}$$

We use bool as an abbreviation for $\mathsf{1} + \mathsf{1}$. Our calculus is parameterised by a collection Ops of primitive operation typings $(\mathrm{op} : T_{\mathrm{op}})$, where for each primitive operation $\mathrm{op} : T_{\mathrm{op}}$, its type T_{op} is closed (i.e., $\vdash T_{\text{op}}\ \mathsf{Type}$). An example set of primitive operations includes dimension-polymorphic arithmetic and test operations on quantities:

$$\begin{aligned}\text{Ops} = (&+ : \forall X.\mathsf{Quantity}(X) \times \mathsf{Quantity}(X) \to \mathsf{Quantity}(X),\\ &\times : \forall X_1.\forall X_2.\mathsf{Quantity}(X_1) \times \mathsf{Quantity}(X_2) \to \mathsf{Quantity}(X_1 \cdot X_2),\\ &1 : \mathsf{Quantity}(1),\\ &\mathsf{inv} : \forall X.\mathsf{Quantity}(X) \to \mathsf{Quantity}(X^{-1}),\\ &< : \forall X.\mathsf{Quantity}(X) \times \mathsf{Quantity}(X) \to \mathsf{bool}).\end{aligned}$$

One could also define a type of signed/zero quantities $\mathsf{real}(X) \coloneqq \mathsf{Quantity}(X) + 1 + \mathsf{Quantity}(X)$, and then extend the language with further arithmetic term constants such as signed addition $+ : \forall X.\mathsf{real}(X) \times \mathsf{real}(X) \to \mathsf{real}(X)$.

To write terms that make use of common sets of dimensions and units, we judge terms in a context $(\Delta_{dim}, \Gamma_{units})$. For example, $\Delta_{dim} = (\mathit{Length}, \mathit{Time})$ and $\Gamma_{units} = (\mathrm{m} : \mathsf{Quantity}(\mathit{Length}), \mathrm{ft} : \mathsf{Quantity}(\mathit{Length}), s : \mathsf{Quantity}(\mathit{Time}))$.

5.4 CATEGORICAL SEMANTICS OF DIMENSION TYPES

Next up we give a general categorical semantics for the λD type theory. Central to this is the notion of a $\lambda\forall$-fibration, as introduced in Chapter 2.

Definition 89 (λD-Model). A λD-model (p, G, Q) is a $\lambda\forall$-fibration $p : \mathcal{E} \to \mathcal{B}$, an Abelian group object G in $\mathcal{B}$, and an object Q in the fibre $\mathcal{E}_G$.

This definition has all of the structure needed to model the λD type theory. As expected, we will use the $\lambda\forall$-fibration to separate the indexing information (the dimensions) from the indexed information (the types and terms).

Dimension contexts will be interpreted as objects of the base category $\mathcal{B}$ and concatenation of dimension contexts will be given by taking the product in $\mathcal{B}$. Recall that an Abelian group object in a category $\mathcal{B}$ with products is given by an object G together with maps $e : 1 \to G$, $m : G \times G \to G$ and $i : G \to G$ satisfying the laws of Abelian groups. We will use this group structure to interpret dimension expressions so that for each vector of n integers we have a morphism $G^n \to G$. Then substituting dimension expressions for dimension variables will be interpreted using the morphisms in $\mathcal{B}$.

An equivalent way to define Abelian group objects if $\mathcal{B}$ has chosen products is as follows. The Lawvere theory for Abelian groups is the category L_{Ab} whose objects are natural numbers, and where a morphism $m \to n$ is an $m \times n$ matrix of integers. Composition of morphisms is given by matrix multiplication, and categorical products are given by arithmetic addition of natural numbers. An Abelian group object in $\mathcal{B}$ is an object G of $\mathcal{B}$ together with a strictly-product-preserving functor $F : \mathsf{L}_{\mathsf{Ab}} \to \mathcal{B}$ such that $F(1) = G$. We will use this alternative view of Abelian group objects in Theorem 95.

Types are interpreted as objects in the fibres above the contexts in which they are defined by using the bicartesian closed structure, and terms are similarly interpreted as morphisms. We will use the reindexing functor to describe substitution for dimension variables in types and terms, and finally universal quantification of dimension variables in types will be interpreted using the right adjoint to reindexing by a projection.

To establish the value of Definition 89 we have to do three things. First, show that a λD-model actually provides categorical models of dimension types. Second, give examples, and finally prove some theorems to show the viability of reasoning at this level of abstraction. We take these in turn.

A λD-model provides a categorical semantics for dimension types as follows.

Dimension Contexts and Dimension Expressions: We interpret dimension contexts $\Delta = X_1, \ldots, X_n$ as the product of the Abelian group object $[\![\Delta]\!] = G^n$ in $\mathcal{B}$. And dimension-expressions-in-context $\Delta \vdash D\ \mathsf{Dim}$ are interpreted as morphisms $G^n \to G$ in the base $\mathcal{B}$, by using the structure of the Abelian group object G. For example, $[\![X, Y \vdash X \cdot Y^{-1}]\!] = G \times G \xrightarrow{\mathrm{id}_G \times i} G \times G \xrightarrow{m} G$, or in other words $[\![X, Y \vdash X \cdot Y^{-1}]\!](g_1, g_2) = g \cdot g^{-1}$. We assume that there is a given interpretation $[\![d]\!] : 1 \to G$ for every primitive dimension constant $d \in \Delta_{dim}$.

Types: The interpretation of well-formed types $\Delta \vdash T\ \mathsf{Type}$ is given by objects $[\![T]\!]$ in the fibre above $[\![\Delta]\!]$, defined by induction on the structure of T. We interpret $\mathsf{1}$, $\mathsf{0}$, $\times$, $+$ and

$\to$ using the bicartesian closed structure of the fibres, and quantification of a dimension variable $[\![\Delta \vdash \forall X.T]\!]$ is defined by right adjoint to reindexing along the projection $\pi : [\![\Delta \vdash \Gamma, X]\!] \to [\![\Delta \vdash \Gamma]\!]$. Quantities $\Delta \vdash \mathsf{Quantity}(D)$ are interpreted by reindexing the object Q along the interpretation of D, i.e. $[\![\Delta \vdash \mathsf{Quantity}(D)]\!] = [\![\Delta \vdash D\ \mathsf{Dim}]\!]^*(Q)$.

Typing Contexts and Terms: Well-formed typing contexts $\Delta \vdash \Gamma\ \mathsf{ctxt}$ are interpreted as products in the fibre above $[\![\Delta]\!]$, i.e., the interpretation of $[\![\Delta \vdash x_1 : T_1, \ldots, x_n : T_n]\!]$ is given by $[\![\Delta \vdash T_1]\!] \times \ldots \times [\![\Delta \vdash T_n]\!]$. We interpret well-formed terms $\Delta, \Gamma \vdash t : T$ as morphisms $[\![t]\!] : [\![\Gamma]\!] \to [\![T]\!]$ in the fibre above $[\![\Delta]\!]$. We assume that there is an interpretation $[\![\mathrm{un}]\!] : 1 \to [\![\mathsf{Quantity}(D)]\!]$ for each unit $(\mathrm{un} : \mathsf{Quantity}(D)) \in \Gamma_{units}$ of dimension D, and an interpretation $[\![\mathrm{op}]\!] : 1 \to [\![T]\!]$ for each primitive operation $(\mathrm{op} : T) \in \mathrm{Ops}$.

For the λD type system we see that the following substitution lemma holds.

Lemma 90. *(Substitution Lemma) Suppose that* $\Delta, X \vdash T\ \mathsf{Type}$ *and that* $\Delta \vdash D\ \mathsf{Dim}$ *denotes a dimension expression. Then* $[\![T[D/X]]\!] \cong (\mathrm{id}_{[\![\Delta]\!]}, [\![D]\!])^*[\![T]\!]$.

Proof. By induction on the structure of T. □

This Lemma says that in the semantics of substituting a dimension expression for a dimension variable is given by reindexing along the identity paired with the dimension expression.

In this Chapter (and in this thesis) we have only considered universal quantification of dimension variables but existential quantification can be given just as easily. Existential quantification is interpreted as the left adjoint to reindexing along a projection, and so we would just require a fibration with this additional structure. Then, properties of existential quantification can be proven by dualising the relevant proofs of properties about universal quantification.

5.5 EXAMPLES

We would expect the syntax to form a λD-fibration and indeed it does. This means that we have a term model and hence a completeness result.

Example 91. *(Syntactical Model) We can construct a category $\mathcal{C}\ell(\lambda D)$ from the syntax in the standard way, and we have a $\lambda\forall$-fibration $p : \mathcal{C}\ell(\lambda D) \to \mathsf{L}_{\mathsf{Ab}}$ with base category given by the Lawvere theory of Abelian groups L_{Ab}, and the fibre $\mathcal{C}\ell(\lambda D)_n$ over n is the category whose objects are types with n dimension variables, and whose morphisms are terms in context. The object 1 in L_{Ab} is an Abelian group object, and hence we have that $(p : \mathcal{C}\ell(\lambda D) \to \mathsf{L}_{\mathsf{Ab}}, 1, (X \vdash \mathsf{Quantity}(X)\ \mathsf{Type}))$ is a λD-model.*

In Kennedy's paper [39], a simpler approach is taken to the semantics of dimensions. Instead of separating the indexing information from the indexed information, Kennedy just focusses on the latter. This means that in his semantics dimensions are simply thrown away in a *dimension-erasure semantics*. From the categorical perspective, this means the calculus is stripped of its fibred structure leaving only a Simply Typed λ-Calculus, which Kennedy models within a bicartesian closed category. In particular, he chooses the category of complete partial orders.[1] Nevertheless, Kennedy's model can be viewed as a λD-model.

Example 92. *(Dimension-Erasure Models) Let $\mathcal{C}$ be a bicartesian closed category. Then the functor $\mathcal{C} \to \mathsf{1}$ is a $\lambda\forall$-fibration, and the unique object of $\mathsf{1}$ is a trivial Abelian group object. By taking $\mathcal{C}$ to be the category of complete partial orders and continuous functions, and by choosing the flat pointed cpo $\mathbb{Q}_\perp$ to interpret $\mathsf{Quantity}$ we obtain a model corresponding to Kennedy's dimension-erasure model. This model supports an array of primitive operations, including all the standard arithmetical ones. However, the model also contains many functions which are not dimensionally invariant, i.e. they do not scale appropriately under change of units. To reduce the model to the dimensionally invariant functions Kennedy uses relational parametricity, which we will discuss further in Section 5.7.*

[1] Kennedy uses the category of complete partial orders because his type system has recursion.

Recall the families fibration $p : \mathsf{Fam}(\mathsf{Set}) \to \mathsf{Set}$ on Set, which has fibres above a set I consisting of pairs (I, f), where I is a set and $f : I \to |\mathsf{Set}|$ is a function. In other words, the fibre above I consists of pairs $(I, \{X_i\}_{i\in I})$, where $\{X_i\}_{i\in I}$ is an I-indexed family of sets. Theorem 23 says that p is a $\lambda\forall$-fibration, and so by choosing an appropriate Abelian group object G, and a G-indexed family of sets, we have a λD-model.

Example 93. *(The Dimension-Indexed Families Model) Suppose that B is a set of fundamental dimensions (e.g. Length, Time, Mass etc.) and let G be the free Abelian group on B. The object Q in the fibre above G must be a G-indexed collection of sets. One (reasonable) choice is given by providing a unit for each generator of G, known here as fundamental dimensions, and obtaining the rest by induction. Formally, for a fundamental dimension D, let $Q_D = \mathbb{R}^+ \times \{\overline{D}\}$, where $\overline{D}$ is a unit for D, i.e. $\overline{Length} = \mathrm{m}$. Then since any element D of G is a monomial $D_1^{k_1} \dots D_n^{k_n}$, we define $Q_D = \mathbb{R}^+ \times \overline{D}$, where $\overline{D} = \overline{D_1^{k_1}} \dots \overline{D_n^{k_n}}$. We then have a λD-model $(p : \mathsf{Fam}(\mathsf{Set}) \to \mathsf{Set},\ G,\ (G, \{Q_D\}_{D\in G}))$.*

In this model, a type with a free dimension variable $X \vdash T$ Type is interpreted as a family of sets, indexed by the dimensions in G. Similarly a term with a free dimension variable is interpreted as a family of functions, one for each dimension in G. This model does support many primitive operations, but it does not support dimension invariant polymorphism. For instance, the model supports adding a term $\mathsf{eq} : \forall X_1.\forall X_2.\mathsf{bool}$ which tests whether two dimensions are the same, which is clearly not invariant under change of representation.

Similarly, we have examples given by the relations fibration $\mathsf{Rel} \to \mathsf{Set}$, and the subobject fibration $\mathsf{Sub}(\mathsf{Set}) \to \mathsf{Set}$, as well as the families fibration $\mathsf{Fam}(\mathcal{C}) \to \mathsf{Set}$ if $\mathcal{C}$ is bicartesian closed.

In Theorem 26, we saw that $\lambda\forall$-fibrations are closed under change-of-base. Next we see that λD-models are too.

Theorem 94. *Let $(p : \mathcal{E} \to \mathcal{B},\ G,\ Q)$ be a λD-model, $F : \mathcal{A} \to \mathcal{B}$ be a product preserving functor, and let A be an Abelian group object in $\mathcal{A}$ such that $FA = G$. Then $(F^*p,\ A,\ (A, Q))$ is a λD-model.*

$$\begin{tikzcd} F^*\mathcal{E} \arrow[r, "p^*F"] \arrow[d, "F^*p"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{E} \arrow[d, "p"] \\ \mathcal{A} \arrow[r, "F"'] & \mathcal{B} \end{tikzcd}$$

Proof. By Theorem 26 F^*p is a $\lambda\forall$-fibration. We have that A is an Abelian group object and Q is in the fibre $(F^*p)_{FA} = (F^*p)_G$ by construction, and hence $(F^*p,\, A,\, (A,Q))$ is a λD-model, as required. □

For a simple illustration of this theorem, notice that the dimension-erasure fibration $\mathcal{C} \to 1$ of Example 92 arises from pulling back the families fibration $\mathsf{Fam}(\mathcal{C}) \to \mathsf{Set}$ along the unique product-preserving functor $F : 1 \to \mathsf{Set}$ given by $F(\star) = 1$.

A simple theorem that shows the usefulness of Theorem 94 says that any λD-model can be expressed as a λD-model with the Lawvere theory of Abelian groups in the base, via change-of-base. Intuitively this means that the model only uses products of the Abelian group object in the base category and so we can essentially throw away all of the other objects.

Theorem 95. *(Models over the Lawvere Theory $\mathsf{L_{Ab}}$) Let $(p : \mathcal{E} \to \mathcal{B},\, G,\, Q)$ be a λD-model. Then there exists a product preserving functor $F : \mathsf{L_{Ab}} \to \mathcal{B}$ such that $\big(F^*p : F^*(\mathcal{E}) \to \mathsf{L_{Ab}},\, 1,\, (1,Q)\big)$ is also a λD-model.*

Proof. Recall that the Abelian group object G in $\mathcal{B}$ gives rise to a unique product-preserving functor $F : \mathsf{L_{Ab}} \to \mathcal{B}$ such that $F(1) = G$. Hence, by Theorem 94, we have a λD-model $(F^*p,\, 1,\, (1,Q))$. □

Next we introduce a model built from group actions that we will examine in detail in Section 5.6.

Theorem 96. *(A Model Built from Group Actions) Let $p : \mathsf{Grp}/\!/\mathsf{Set} \to \mathsf{Grp}$ be the $\mathsf{Grp}/\!/\mathsf{Set}$ fibration, G be an Abelian group, and let Q be a G-set. Then $(p,\, G,\, Q)$ is a λD-model.*

Proof. The $\mathsf{Grp}/\!/\mathsf{Set}$ fibration p is a $\lambda\forall$-fibration by Theorem 33, and since an Abelian group induces an Abelian group object in Grp,[2] we have that $(p,\, G,\, Q)$ is a λD-model. □

[2]There is a functor $\mathsf{Ab}(\mathsf{Set}) \to \mathsf{Ab}(\mathsf{Grp})$ that maps an Abelian group to an Abelian group object in Grp using the group multiplication as the Abelian group object structure. For an Abelian group G, to show that the map $m : G \times G \to G$ is a group homomorphism crucially relies on the fact that G is *Abelian*. i.e., not just a group.

The $\mathsf{Grp}//\mathsf{Set}$ model cannot support dimension constants because there is only one group homomorphism $1 \to G$. It does support several term constants, which we discuss after Theorem 100.

More generally, instead of having sets and group actions, we also have λD-models built from actions of groupoids. Recall the following definitions.

Definition 97 (Groupoid). A *groupoid* is a small category $\mathcal{C}$ where every morphism is an isomorphism, and we denote the category of groupoids and functors by Gpd. A *groupoid action* (or *presheaf*) is a functor $\mathcal{C} \to \mathsf{Set}$.

The category $\mathsf{Gpd}//\mathsf{Set}$ has objects given by pairs $(\mathcal{A}, \phi)$, where $\mathcal{A}$ is a groupoid and $\phi : \mathcal{A} \to \mathsf{Set}$ is a functor, and morphisms given by pairs $(\mathcal{A}, \phi) \to (\mathcal{B}, \psi)$, where $F : \mathcal{A} \to \mathcal{B}$ is a functor and $\eta : \phi \to \psi \circ F$ is a natural transformation between functors $\mathcal{A} \to \mathsf{Set}$.

By looking at the forgetful functor $\mathsf{Gpd}//\mathsf{Set} \to \mathsf{Gpd}$, which we call the $\mathsf{Gpd}//\mathsf{Set}$ *fibration*, we have another example of a λD-fibration.

Example 98. *(A Model Built from Groupoid Actions) The* $\mathsf{Gpd}//\mathsf{Set}$ *fibration* $\mathsf{Gpd}//\mathsf{Set} \to \mathsf{Gpd}$ *is a* λD*-model. The proof of this is very similar to Theorem 96, and so we do not reproduce it here.*

The $\mathsf{Gpd}//\mathsf{Set}$ fibration can be used to construct the fibrations of Example 93 and Example 96 by change-of-base.

- The families fibration $\mathsf{Fam}(\mathsf{Set}) \to \mathsf{Set}$ from Example 93 arises from pulling back the $\mathsf{Gpd}//\mathsf{Set}$ fibration $\mathsf{Gpd}//\mathsf{Set} \to \mathsf{Gpd}$ along the discrete-groupoid-functor $\mathsf{Set} \to \mathsf{Gpd}$.
- The group action fibration $\mathsf{Grp}//\mathsf{Set} \to \mathsf{Grp}$ from Example 96 arises from pulling back the groupoid action fibration $\mathsf{Gpd}//\mathsf{Set} \to \mathsf{Gpd}$ along the functor $\mathsf{Grp} \to \mathsf{Gpd}$, which maps each group as a groupoid with one object in the usual way.

Hence, the $\mathsf{Gpd}//\mathsf{Set}$ fibration subsumes the families and group actions fibrations. In fact it also subsumes them as λD-models.

To see this first recall that a homomorphism of Abelian groups $f : G \to H$ induces a groupoid. The objects are given by the elements of H, and the hom-sets are given by $\hom(h, h') = \{g \in G \mid f(g) \cdot_H h = h'\}$, where composition is given by the group operation in G. This groupoid can be given the structure of an Abelian group object in Gpd, and, moreover, every Abelian group in Gpd arises in this way [42].

To see that the $\mathsf{Gpd}//\mathsf{Set}$ model subsumes the group action model of Example 96, let G be an Abelian group of scale factors. Then the Abelian group object induced by the unique homomorphism $G \to 1$ is a one-object groupoid, and hence we can build the λD-models of group actions. To recover the families model of Example 93, fix a set of dimension constants and let H be the free Abelian group on that set. The unique homomorphism $1 \to H$ induces the discrete groupoid whose objects are H, and hence we build the λD-models of families of sets.

5.6 GROUP ACTIONS AND DIMENSION TYPES

In this section we will look in greater detail at the λD-model given by the $\mathsf{Grp}//\mathsf{Set}$ fibration. Many interesting theorems can be proven in this model, and so we now take the time to spell out the reindexing and simple product structure. Throughout this section we will use semantic brackets $[\![\_]\!]$ to refer only to the $\mathsf{Grp}//\mathsf{Set}$ interpretation.

Reindexing: Let ϕ be a G-set. Then since reindexing is given by precomposition, we have that reindexing along $\pi : \mathcal{G} \times \mathcal{H} \to \mathcal{G}$ gives the $G \times H$-set given by $\phi \circ \pi$. In other words, $\pi^*\phi$ is a $G \times H$-set with the same underlying set $|\pi^*\phi| = |\phi|$ as the G-set ϕ, and action given by $(g, h) \cdot_{\pi^*\phi} x = g \cdot_\phi x$.

Simple Products: Let $\psi : \mathcal{G} \times \mathcal{H} \to \mathsf{Set}$ be a $G \times H$-set. According to Theorem 27, the underlying set of $\forall_\pi \psi$ is given by $|\forall_\pi \psi| = \lim_{y \in \mathcal{H}} \psi(\star, y)$, which we can compute using the universal property of limits as follows.

$$\lim_{y \in \mathcal{H}} \psi(\star, y) \cong \mathsf{Set}\big(1, \lim_{y \in \mathcal{H}} \psi(\star, y)\big) \cong [\mathcal{H}, \mathsf{Set}]\big(K1, \psi(\star, \_)\big)$$

Hence $|\forall_\pi \psi| = \{y \in |\psi| \mid \forall h \in H \ . \ (e_G, h) \cdot_\psi y = y\}$, and the action is given by $g \cdot_{\forall_\pi \psi} x = (g, e_H) \cdot_\psi x$. Notice that to give the group action of $\forall_\pi \psi$, we had to make a particular choice of an element in H, namely the identity element e_H. However, any element of H would have given the same result, since for all $y \in |\forall_\pi \psi|$,

$$(g,h) \cdot_\psi y \;=\; ((g,e_H)(e_G,h)) \cdot_\psi y \;=\; (g,e_H) \cdot_\psi ((e_G,h) \cdot_\psi y) \;=\; (g,e_H) \cdot_\psi y.$$

Note that by Substitution Lemma 90, we have that in the $\mathsf{Grp}/\!/\mathsf{Set}$ fibration

$$(\mathrm{id}_{[\![\Delta]\!]}, [\![D]\!])^* [\![T]\!] \cong [\![T]\!](\mathrm{id}_{[\![\Delta]\!]}, [\![D]\!]),$$

since reindexing is given by precomposition. In other words, substitution of the n^{th} unit variable is given by precomposition at the n^{th} component.

Now that we have explored the structure we can start to prove some theorems. In Kennedy's treatment of dimension types he introduces a relational semantics, and proves a parametricity theorem. However, using our categorical semantics many of the theorems that he proves using parametricity can be proven in the $\mathsf{Grp}/\!/\mathsf{Set}$ fibration, without having to define a separate relational semantics. This forms the content of Theorems 100–105.

First, recall the following theorem about ends (as discussed in [9, Section 5]).

Theorem 99. *For two functors $F : \mathcal{C} \to \mathcal{D}$, $G : \mathcal{C} \to \mathcal{D}$, Nat$(F, G) \cong \int_C \hom(FC, GC)$.*

Using Theorem 99 and the $\mathsf{Grp}/\!/\mathsf{Set}$ model we can characterise the interpretation of arrow types with a universally quantified variable.

Theorem 100. *Suppose that $X_1, \ldots, X_n, X \vdash S, T$ Type, then*

$$|[\![\forall X.S \to T]\!]| \cong [G, \mathsf{Set}]\big([\![S]\!](\underbrace{\star, \ldots, \star}_{n-times}, \_), [\![T]\!](\underbrace{\star, \ldots, \star}_{n-times}, \_)\big).$$

Proof. Let π denote the projection morphism $[\![X_1, \ldots, X_n, X]\!] \to [\![X_1, \ldots, X_n]\!]$, let $\forall_\pi$ denote the right-adjoint to reindexing along π, i.e., $\pi^* \dashv \forall_\pi$, let Δ denote $X_1, \ldots, X_n$, and let Ran_π denote the right Kan extension along π. Then, by definition we have that

$$|[\![\Delta \vdash \forall X.S \to T]\!]| \cong [\![\Delta \vdash \forall X.S \to T]\!]\star_n \cong (\mathrm{Ran}_\pi[\![\Delta, X \vdash T \to S]\!])\star_n\ ,$$

where $\star_n$ denotes the unique object of the category $\mathcal{G}^n$.

By using the end formula for a right Kan extension,[3] we have that

$$\begin{aligned}
&|[\![\Delta \vdash \forall X.S \to T]\!]| \\
&\quad\cong \int_{\star_{n+1}} \mathcal{G}^n(\star_n, \star_n) \Rightarrow [\![\Delta, X \vdash T \to S]\!]\star_{n+1} && \text{by Kan extension formula,}\\
&\quad\cong \int_\star \int_{\star_n} \mathcal{G}^n(\star_n, \star_n) \Rightarrow [\![\Delta, X \vdash T \to S]\!](\star_n, \star) && \text{by separating the end,}\\
&\quad\cong \int_\star [\mathcal{G}^n, \mathsf{Set}]\big(\mathcal{G}^n(\star_n, \_), [\![\Delta, X \vdash T \to S]\!](\_, \star)\big) && \text{by Theorem 99,}\\
&\quad\cong \int_\star [\![\Delta, X \vdash T \to S]\!](\star_n, \star) && \text{by the Yoneda Lemma,}\\
&\quad\cong \int_\star [\![\Delta, X \vdash T]\!](\star_n, \star) \Rightarrow [\![\Delta, X \vdash S]\!](\star_n, \star) && \text{by the definition of } [\![\_]\!],\\
&\quad\cong [G, \mathsf{Set}]([\![\Delta, X \vdash T]\!](\star_n, \_), [\![\Delta, X \vdash S]\!](\star_n, \_) && \text{by Theorem 99,}
\end{aligned}$$

as required. □

This Theorem says that in the $\mathsf{Grp}//\mathsf{Set}$ model a universally quantified variable over an arrow type can be considered as a natural transformation between the domain and codomain of the arrow type, with the first n components fixed. In other words, it is interpreted as the set of functions that are equivariant in the last argument.

As a simple consequence of this theorem, we can see that the group actions model supports several different term constants. Given a choice of G-set Q, then for any element q of Q, we can accommodate a term constant $q : \mathsf{Quantity}(1)$, which is interpreted by $[\![q]\!] = q$. When $Q = G$, we can also accommodate a term constant for multiplication

$$\times : \forall X.\forall Y.\, \mathsf{Quantity}(X) \times \mathsf{Quantity}(Y) \to \mathsf{Quantity}(X \cdot Y)$$

[3] This is a well-known formula, see for example [9].

which is interpreted as the group operation. When $Q = G = (\mathbb{R}^+, \times, 1)$, the positive reals, we also have addition, $+ : \forall X.\, \mathsf{Quantity}(X) \times \mathsf{Quantity}(X) \to \mathsf{Quantity}(X)$, which is equivariant since $q(r + s) = qr + qs$.

Theorem 101. *Suppose that we have a type* $\Delta, X \vdash T$ Type *with one free dimension variable* X. *Then,*

$$|[\![\forall X.\mathsf{Quantity}(X) \to T]\!]| \cong |[\![T[1/X]]\!]|. \tag{5.2}$$

Proof. Firstly, by expanding the left hand side of (5.2) and using Theorem 100, we see that

$$|[\![\Delta \vdash \forall X.\mathsf{Quantity}(X) \to T]\!]| \cong [G, \mathsf{Set}]\big([\![\Delta, X \vdash X]\!]^* \mathcal{Q}(\star_n, \_), [\![\Delta, X \vdash T]\!](\star_n, \_)\big).$$

Then notice that the underlying carrier of the G-Set $[\![\Delta, X \vdash X]\!]^* \mathcal{Q}(\star_n, \_)$ is given by $[\![\Delta, X \vdash X]\!]^* \mathcal{Q}(\star_n, \star) = \mathcal{Q}(\star) = G$, and the action is defined by $[\![\Delta, X \vdash X]\!]^* \mathcal{Q}(\star_n, g) = g \cdot \_$. Hence, the functor $[\![\Delta, X \vdash X]\!]^* \mathcal{Q}(\star_n, \_)$ is equal to the hom-functor $\mathcal{G}(\star, \_)$, and so we see that,

$$\begin{aligned} |[\![\Delta \vdash \forall X.\mathsf{Quantity}(X) \to T]\!]| &\cong [G, \mathsf{Set}](\mathcal{G}(\star, \_), [\![\Delta, X \vdash T]\!](\star_n, \_)) \\ &\cong [\![\Delta, X \vdash T]\!](\star_n, \star) \quad \text{by the Yoneda Lemma.} \end{aligned}$$

Finally, by expanding the right hand side of (5.2), and using Theorem 100, we see that $|[\![\Delta \vdash T[1/X]]\!]| \cong (\mathrm{id}_{[\![\Delta]\!]}, [\![\Delta, X \vdash 1]\!])^* [\![\Delta, X \vdash T]\!] \star_n \cong [\![\Delta, X \vdash T]\!](\star_n, \star)$, as required. □

We now prove some theorems about the $\mathsf{Grp}/\!/\mathsf{Set}$ fibration that are proven using parametricity in Kennedy's original paper. The proofs here involve applications of Lemma 90, Theorem 100 and Theorem 101. First, we take a look at the invariance of polymorphic functions under scaling.

Theorem 102. *Suppose that we have* $\Delta_{dim}; \Gamma_{ops} \vdash t : \forall X.\mathsf{Quantity}(X) \to \mathsf{Quantity}(X^n)$, *where* $n \in \mathbb{N}$. *Then, for any element* g *of* G *and any element* x *of* $|[\![\mathsf{Quantity}(X)]\!]|$, *we have that* $[\![t]\!](g \cdot x) = g^n \cdot [\![t]\!]x$.

Proof. We know from Theorem 100 that $[\![t]\!] \in [G, \mathsf{Set}]([\![\mathsf{Quantity}(X)]\!], [\![\mathsf{Quantity}(X^n)]\!])$. Or in other words, $[\![t]\!](g \cdot x) = g^n \cdot [\![t]\!]x$ for all $x \in |[\![\mathsf{Quantity}(X)]\!]|$, as required. □

This theorem tells us that polymorphic functions are *invariant under scaling*. If we apply Theorem 101 to the type $\forall X.\mathsf{Quantity}(X) \to \mathsf{Quantity}(X^n)$, we see that

$$|[\![\forall X.\mathsf{Quantity}(X) \to \mathsf{Quantity}(X^n)]\!]| \cong |[\![\mathsf{Quantity}(1^n)]\!]| \cong |[\![\mathsf{Quantity}(1)]\!]| \cong Q.$$

Hence, we conclude that all the terms of type $\forall X.\mathsf{Quantity}(X) \to \mathsf{Quantity}(X^n)$ are of the form $\Lambda X.\, \lambda q : \mathsf{Quantity}(X).\ r \times q^n$ for $r \in Q$.

Theorem 103. *There is no ground term* $\vdash t : \forall X.\mathsf{Quantity}(X^2) \to \mathsf{Quantity}(X)$.*, i.e., we cannot write a polymorphic square root function.*

Proof. To see this we exhibit a model where the existence of such a term is impossible. Consider the λD-model $(p : \mathsf{Grp}/\!/\mathsf{Set} \to \mathsf{Grp},\ \mathbb{Z}_2,\ \mathbb{Z}_2)$, where $\mathbb{Z}_2$ denotes the Abelian group $= (\{-1, 1\}, \cdot, 1)$. Theorem 100 says that the interpretation of the type $\forall X.\mathsf{Quantity}(X^2) \to \mathsf{Quantity}(X)$ is given by,

$$|[\![\forall X.\mathsf{Quantity}(X^2) \to \mathsf{Quantity}(X)]\!]| \cong [\mathbb{Z}_2, \mathsf{Set}]([\![\mathsf{Quantity}(X^2)]\!], [\![\mathsf{Quantity}(X)]\!]),$$

i.e. any element f of $|[\![\forall X.\mathsf{Quantity}(X^2) \to \mathsf{Quantity}(X)]\!]|$, satisfies for all $g, x \in \mathbb{Z}_2$,

$$f(g^2 \cdot x) = g \cdot (fx) \tag{5.3}$$

If f exists, then either $f(-1) = -1$ or $f(-1) = 1$, but both lead to contradictions. To this end suppose that $f(-1) = -1$, then by Equation 5.3 we have that $f((-1)^2 \cdot -1) = (-1) \cdot f(-1)$, which is a contradiction since the left-hand side is equal to -1 and the right-hand side is equal to 1. A similar argument shows that $f(-1) = 1$ is also not possible, and hence there exists no such f. □

This result can be extended to also include terms t using primitive operations, i.e. $\Gamma_{ops} \vdash t : \forall X.\mathsf{Quantity}(X^2) \to \mathsf{Quantity}(X)$, as long as these operations can be interpreted in the model. For example, the result holds in the presence of multiplication, since it can be interpreted in the model as mentioned just before Theorem 101. This model does not, however, support a polymorphic zero constant $0 : \forall X.\, \mathsf{Quantity}(X)$, as such a primitive would of course gives rise to a trivial counterexample to the theorem.

Next, we can prove a theorem that relates a dimensionally invariant function to a dimensionless one. This is a simplified version of the *Buckingham Pi Theorem* of dimensional analysis [43] (or for a more modern introduction see Sonin [44]).

Theorem 104. *We have a bijection*

$$|[\![\forall X.\mathsf{Quantity}(X) \times \mathsf{Quantity}(X) \to \mathsf{Quantity}(1)]\!]| \cong |[\![\mathsf{Quantity}(1) \to \mathsf{Quantity}(1)]\!]|.$$

Proof. This is a consequence of Theorem 101, after currying. □

We finish this section with another uninhabitedness result, this time about a higher order type. Higher order types normally show the power of parametricity, since they are mixed variance, but as we do not have type variables, types are not interpreted as functors, and so variance is not an issue.

Theorem 105. *There is no ground term*

$$\vdash t : \forall X_1.\forall X_2.(\mathsf{Quantity}(X_1) \to \mathsf{Quantity}(X_2)) \to \mathsf{Quantity}(X_1 \cdot X_2).$$

Proof. Choose G and Q to be $\mathbb{Z}_2$. Interpreting the type of t, we have that the interpretation of $[\![\forall X_1.\forall X_2.(\mathsf{Quantity}(X_1) \to \mathsf{Quantity}(X_2)) \to \mathsf{Quantity}(X_1 \cdot X_2)]\!]$ has the underlying set given by

$$\{t \in (\mathbb{Z}_2 \to \mathbb{Z}_2) \to \mathbb{Z}_2 \mid \forall g_1, g_2 \in \mathbb{Z}_2,\ f : \mathbb{Z}_2 \to \mathbb{Z}_2.\ (g_1 g_2) \cdot (tf) = t(\lambda q \in \mathbb{Z}_2.\, g_2 \cdot (f(g_1^{-1} \cdot q)))\}.$$

Hence for any $t \in [\![\forall X_1.\forall X_2.(\mathsf{Quantity}(X_1) \to \mathsf{Quantity}(X_2)) \to \mathsf{Quantity}(X_1 \cdot X_2)]\!]$, by choosing f to be equal to id_Q we have that $(g_1 g_2) \cdot (t(\mathrm{id}_Q)) = t(\lambda q \in \mathbb{Z}_2.\, g_2 \cdot (g_1^{-1} \cdot q))$ for all g_1 and g_2 in $\mathbb{Z}_2$, but this is not possible. If $g_1 = 1$ and $g_2 = -1$, then the equation reduces to $-1 \cdot t(\mathrm{id}_Q) = t(\mathrm{id}_Q)$, which is a contradiction since $t(\mathrm{id}_Q) \in \mathbb{Z}_2 = \{-1, 1\}$. □

Again, the result can be extended to terms defined in a context of primitive operations.

5.7 RELATIONAL MODELS

Parametricity is a powerful technique and in Kennedy's work he uses it fully. He is able to prove a series of theorems, all as a direct consequence of providing a relational semantics and proving a parametricity theorem. Curiously, all of those theorems can be proven in the Grp//Set λD-model, and in Section 5.6 we showed how. This did not involve defining a relational semantics, and no parametricity theorem was required. This makes one wonder whether the Grp//Set λD-model is just as good as having full-blown parametricity at ones finger tips?

To look for an answer, we now provide a general method of attaching a (fibrational) logic to a λD-model to give a notion of a *relational* λD-model. This allows us to reconstruct Kennedy's relational parametricity in our setting (Example 107), as well as talking about a relational version of the Grp//Set λD-model (Example 108).

Given a λD-model $q : \mathcal{A} \to \mathcal{L}$ and a logic $p : \mathcal{E} \to \mathcal{B}$, there is a natural way to glue them together to provide a relational semantics.

$$\begin{tikzcd}
F^*(\mathcal{E}) \arrow[r] \arrow[d, "F^*p"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{A} \arrow[r, "F"'] \arrow[d, "q"'] & \mathcal{B} \\
\mathcal{L} &
\end{tikzcd}$$

Theorem 106. *Let $(q : \mathcal{A} \to \mathcal{L}, G, Q_0)$ be a λD-model, $F : \mathcal{A} \to \mathcal{B}$ a product preserving functor and $p : \mathcal{E} \to \mathcal{B}$ a bicartesian closed fibration with products. Consider the pullback of p along F, and let Q_R denote an object in the fibre $\mathcal{E}_{F(Q_0)}$. Then, the triple given by $(q \circ F^*p : F^*\mathcal{E} \to \mathcal{L}, G, (Q_0, Q_R))$ is a λD-model.*

Proof. Clearly G is an Abelian group object in $\mathcal{L}$, and (Q_0, Q_R) is in the fibre $(F^*\mathcal{E})_G$. To check that $F^*p \circ u$ is a bicartesian closed fibration is a simple exercise. Finally, since p has all products, so does F^*p. Hence, $F^*p \circ u$ has simple products by Theorem 25. □

Next, we look at an example that uses Theorem 106 to generate Kennedy's original relationally parametric model of dimension types [39] from essentially the dimension-erasure model back in Example 92.

Example 107. *Let G be an Abelian group. Then using the notation from Theorem 106, let $\mathcal{L}$ be the Lawvere theory of Abelian groups L_{Ab}, $\mathcal{A}$ be the category given by the product $\mathsf{L}_{\mathsf{Ab}} \times \mathsf{Set}$, $q : \mathsf{L}_{\mathsf{Ab}} \times \mathsf{Set} \to \mathsf{L}_{\mathsf{Ab}}$ be the fibration given by the first projection, and $p : \mathsf{Sub}(\mathsf{Set}) \to \mathsf{Set}$ be the subset fibration. Define $F : \mathsf{L}_{\mathsf{Ab}} \times \mathsf{Set} \to \mathsf{Set}$ to be the product preserving functor defined on objects $(n, X) \in \mathsf{L}_{\mathsf{Ab}} \times \mathsf{Set}$ by $F(n, X) = G^n \times X \times X$, and on morphisms $(f, g) : (n, X) \to (m, Y)$ by $F(f, g) = (G^f, g, g)$. Finally, we let $Q_0 = G$, and $Q_R = \{(g, g_1, g_2) \mid gg_1 = g_2\} \subseteq G \times G \times G$.*

In this model, each type $\Delta \vdash T$ is interpreted as a triple $(|\Delta|, [\![T]\!]_o, [\![T]\!]_r) \in \mathsf{L}_{\mathsf{Ab}} \times \mathsf{Set} \times \mathsf{Sub}(\mathsf{Set})$, where $[\![T]\!]_r \subseteq G^{|\Delta|} \times [\![T]\!]_o \times [\![T]\!]_o$. Spelling this out explicitly, we have the following interpretations, which are equivalent to Kennedy's original relationally parametric model for dimension types:

$$
\begin{aligned}
[\![\Delta \vdash \mathsf{Quantity}(D)]\!] &= (|\Delta|,\, G,\, \{(g, g_1, g_2) \mid ([\![D]\!]g)g_1 = g_2\}) \\
[\![\Delta \vdash T \times U]\!] &= (|\Delta|,\, [\![T]\!]_o \times [\![U]\!]_o, \\
&\quad \{(g, (t_1, u_1), (t_2, u_2)) \mid (g, t_1, t_2) \in [\![T]\!]_r, (g, u_1, u_2) \in [\![U]\!]_r\} \\
[\![\Delta \vdash T + U]\!] &= (|\Delta|,\, [\![T]\!]_o + [\![U]\!]_o, \\
&\quad \{(g, \iota_1 t, \iota_1 t') \mid (g, t, t') \in [\![T]\!]_r\} \cup \{(g, \iota_2 u, \iota_2 u') \mid (g, u, u') \in [\![U]\!]_r\} \\
[\![\Delta \vdash T \to U]\!] &= (|\Delta|,\, [\![T]\!]_o \to [\![U]\!]_o, \\
&\quad \{(g, f_1, f_2) \mid \forall t_1, t_2.\ (g, t_1, t_2) \in [\![T]\!]_r \implies (g, f_1 t_1, f_2 t_2) \in [\![U]\!]_r\} \\
[\![\Delta \vdash \forall X.\ T]\!] &= (|\Delta|,\, [\![T]\!]_o,\, \{(g, t_1, t_2) \mid \forall g' \in G.\ ((g, g'), t_1, t_2) \in [\![T]\!]_r\})
\end{aligned}
$$

Note that in the interpretation of types $\forall X.\ T$, the carrier *(i.e., the second component) is exactly the carrier of the interpretation of T.*

We can also apply Theorem 106 to obtain a natural relational model for the $\mathsf{Grp}/\!/\mathsf{Set}$ λD-model (Example 96).

Example 108. *As before, let G be an Abelian group and $\mathcal{L}$ be the Lawvere theory of Abelian groups L_{Ab}. Let $q : \mathcal{A} \to \mathsf{L}_{\mathsf{Ab}}$ be the pullback of the fibration $\mathsf{Grp}/\!/\mathsf{Set} \to \mathsf{Grp}$ along the unique product-preserving functor $M : \mathsf{L}_{\mathsf{Ab}} \to \mathsf{Grp}$ with $M(1) = G$, as in Example 95, so that the objects of $\mathcal{A}$ are triples (n, X, ϕ) with (X, ϕ) a G^n-set. Let $p : \mathsf{Sub}(\mathsf{Set}) \to \mathsf{Set}$ be the subset fibration. Define $F : \mathcal{A} \to \mathsf{Set}$ to be the product preserving functor defined* *on objects by $F(n, X, \phi) = G^n \times X \times X$ and on morphisms $(f, \alpha) : (n, X, \phi) \to (m, Y, \psi)$ by $F(f, \alpha) = (\alpha, f, f)$. Finally, we let $Q_0 = (G, \phi)$, where ϕ denotes group multiplication, and $Q_R = \{(g, g_1, g_2) \mid gg_1 = g_2\} \subseteq G \times G \times G$.*

Then each type $\Delta \vdash T$ is again interpreted as a triple $(|\Delta|, [\![T]\!]_o, [\![T]\!]_r) \in \mathsf{L_{Ab}} \times \mathsf{Sub(Set)}$, with $[\![T]\!]_r \subseteq G^{|\Delta|} \times [\![T]\!]_o \times [\![T]\!]_o$. The only difference between the interpretation of types in this example and Example 107 is the second component of the interpretation of dimension quantification:

$$
\begin{aligned}
[\![\Delta \vdash \forall X.\ T]\!]_r = (|\Delta|,\ &\{t \in |[\![T]\!]_o| \mid \forall g \in G.\ ((e_{G^{|\Delta|}}, g), t, t) \in [\![T]\!]_r\}, \\
&\{(g, t_1, t_2) \mid \forall g' \in G.\ ((g, g'), t_1, t_2) \in [\![T]\!]_r\}).
\end{aligned}
$$

This interpretation, in contrast to the interpretation in Example 107, has "cut-down" the carrier of the interpretation of $\forall$-types to only include the parametric elements. As a consequence, this interpretation satisfies an analogue of the Identity Extension Lemma (see Theorem 37) from relationally parametric models of System F .

Theorem 109. *For any type $\Delta \vdash T$ Type with semantics given by $(|\Delta|, [\![T]\!]_o, [\![T]\!]_r)$ as outlined in Example 108, we have that for any $x_1, x_2 \in [\![T]\!]_o$,*

$$
(e, x_1, x_2) \in [\![T]\!]_r \Leftrightarrow x_1 = x_2.
$$

Comparing this to the Identity Extension Lemma, we see that equality relations for the free type variables are replaced by the unit element of the group $G^{|\Delta|}$.

We end this discussion of relational models by showing the relationships between the models in Examples 107 and 108 and the $\mathsf{Grp}//\mathsf{Set}$ model we considered in detail in Section 5.6. By construction, the carriers of the interpretations of each type in the model in Example 108 and the $\mathsf{Grp}//\mathsf{Set}$ model are identical. Moreover, the relational interpretation in Example 108 and the group action in the $\mathsf{Grp}//\mathsf{Set}$ model are related as follows.

Theorem 110. *For any type $\Delta \vdash T$ Type, if the interpretation of T in the model of Example 108 is $(|\Delta|, A, P \subseteq G^{|\Delta|} \times A \times A)$ and the $\mathsf{Grp}//\mathsf{Set}$ model interpretation is $(G^{|}\Delta|, A, \psi)$, then $(g, a_1, a_2) \in P \Leftrightarrow g \cdot_{\psi} a_1 = a_2$.*

Proof. By induction on the derivation of $\Delta \vdash T$ Type. □

Using Theorem 110, we can see that we could have used the relationally parametric model to derive the results in Section 5.6. There is literally no difference between the two models for the purposes of interpreting the types of our calculus.

It remains to discuss the relationship between Kennedy's original relational model (Example 107), and the relational model in Example 108 that satisfies the identity extension property. As noted above, the difference between these interpretations lies in the semantics of the $\forall$-type. Kennedy's model does not restrict the carrier of the interpretation to just the "parametric" elements, *i.e.*, the elements that preserve all relations. Therefore, the interpretations of types that contain nested $\forall$s are not directly comparable. We might expect that we could observe a difference between the two models when proving statements about terms whose types contain negatively nested forall types. However, Kennedy's original work does not present any results involving terms with such types, and we have not found any natural examples. This is in contrast with the situation with relationally parametric models of System F, where the proof that final coalgebras can be represented crucially relies on the restriction of the interpretation of quantified types to the parametric elements [29].

Therefore, our Grp//Set model and the equivalent relational model in Example 108 practically coincides with Kennedy's original model, but offer the advantage of not requiring a separate relational semantics to prove important theorems. This in many cases makes proofs of these theorems clearer. Additionally, the Grp//Set model offers an interpretation that directly links the semantics to symmetry.

5.8 CONCLUDING REMARKS

To conclude, in this chapter we have studied a typed λ-calculus with polymorphism over physical dimensions, which we called λD (Section 5.2) and we have developed a model theory for the calculus. Under the Curry-Howard correspondence, the λD-calculus is a fragment of first-order logic where the domain of discourse is an unspecified Abelian group, and so our notion of model (Definition 89) is based on the standard fibrational techniques in categorical logic.

One particular model turned out to be particularly straightforward and yet informative — the model based on group actions (Example 96). Of course, automorphisms and group actions play a key role in the classical model theory of first order logic, but in this paper we have shown that these techniques are also useful on the other side of the Curry-Howard correspondence. Many arguments about the λD-calculus, including type isomorphisms and definability arguments, can be made in this model (Section 5.6).

Parametricity is most often studied using relational techniques, and in this Chapter 5 we have developed a method for building relational λD-models (Theorem 106). Using this method we were able to reconstruct two particular relational models: a relational model due to Kennedy (Example 107, [39]) and a restriction of a relational model due to Atkey (Example 108, [32]). Although the group-actions model is different in style, we showed (formally) that it is actually closely related to the two relational models (Theorem 110).

CHAPTER 6

CONCLUSION

In this thesis we have used fibrations to understand parametricity. In Chapter 3 we demonstrated how the original description of relational parametricity can be reinterpreted using fibrations. This revealed some previously hidden categorical structure in the original definitions. In this setting we were able to prove the existence of initial algebras for functorial types, using fibrational structure to prove one of the key results required — the Graph Lemma.

In Chapter 4 we kept System F as our type system but looked to change the backdrop. Departing from sets and relations, we were able to give a parametric semantics for System F using G-sets and equivariant relations. This was described in terms of the relations fibration on G-sets. Whilst looking at initial algebras in this setting we came to the conclusion that a generalised version of the "normal" initial algebra theorem holds for G-sets. For example the characterisation of the terminal object 1 became the G-set given by the group with the conjugate action G_C, and the characterisation of the natural numbers $\mathbb{N}$ became the collection of non-empty lists $\mathsf{List}^+(G_C)$ containing elements of G_C. We were able to generalise this discovery to prove an initial algebra theorem for G-sets.

In Chapter 5 we moved away from System F to explore a type system with dimension types. We were able to give a general categorical model to add to the semantics based on complete partial orders that has previously been developed by Andrew Kennedy. This allowed us to view a large array of different models for the λD-type system, and in particular for us to look at a model given by G-sets. Using the G-sets model we were able to prove a big variety of theorems that previously had required parametricity. This meant that without defining a separate relational semantics we could prove results directly, which lead to simple and slick proofs. The key to this approach was that our categorical semantics kept track of the dimensions that indexed types, instead of the dimension-erasure approach taken by Kennedy. Additionally, we were able to define a notion of a general relational model for the λD-type system, and use this to compare the G-set model to one in the presence of parametricity. When it came to interpreting the syntax and proving theorems, the G-set model was just as powerful as the relational model.

There was big difference between the parametricity seen in Chapters 3 and 4 compared to Chapter 5. In Chapters 3 and 4 parametricity clearly gave us extra strength to prove theorems, even when working with G-sets. But in Chapter 5 we found that the G-set model was as powerful a parametricity. The key difference between these chapters was the presence of type variables. By not having type variables types did not define functors, meaning that in Chapter 5 there was no initial algebras theorem, but also that parametricity was in some way restricted.

Much of the strength of parametricity comes from the fact that it applies to mixed-variance functors and is more powerful than dinaturality, and hence the λD-type system is not able to express this power. This suggests a possible direction for future work. One could develop a type system where the types were indexed by some kind of indexing structure (such as dimensions) but there was also the presence of type variables. This would allow a discussion of parametricity that would cover Chapters 3 – 5.

Another direction for possible future work would be to look more closely at the relationship between symmetry and parametricity. A paper by Atkey [32] has shown how relational parametricity can be used to derive free theorems from continuous smooth functions used in classical mechanics. These free theorems turn out to give the exact conditions in which Noether's Theorem can be applied, which means that parametricity can lead directly to conservation properties.

It would be interesting to see what insights the fibrational perspective could bring to Atkey's work on Noether's theorem. Extending this further, there are many examples of invariance properties being used throughout physics and it would be interesting to see whether parametricity (and the fibrational perspective) can play a role in these other areas.

I hope that this thesis will be useful to those (categorically inclined people) looking to understand parametricity in different settings. There has yet to be developed a fully general characterisation of parametricity, and so for many it still seems to have some mysterious properties. This thesis has built on and been directly inspired by the pioneering work of Ma and Reynolds [45], Hermida [11, 46], Dunphy and Reddy [47], and Birkedal and Møgelberg [29], who have all developed categorical formulations for parametricity in different settings.

It is my hope that the material presented here will help to demystify parametricity in the situations discussed in this thesis, and contribute to the overall understanding of the subject.

Thank you for reading.

BIBLIOGRAPHY

[1] John C Reynolds. Types, abstraction and parametric polymorphism. 1983.

[2] Neil Ghani, Patricia Johann, Fredrik Nordvall Forsberg, Federico Orsanigo, and Timothy Revell. Bifibrational functorial semantics of parametric polymorphism. *MFPS*, 2015.

[3] Robert Atkey, Neil Ghani, Fredrik Nordvall Forsberg, Timothy Revell, and Sam Staton. Models for polymorphism over physical dimensions. In *13th International Conference on Typed Lambda Calculi and Applications (TLCA'15)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 45–59, 2015. doi: 10.4230/LIPIcs.TLCA.2015.45.

[4] Bart Jacobs. *Categorical logic and type theory*, volume 141. Elsevier, 1999.

[5] Thomas Streicher. Fibred categories à la Jean Bénabou. 1999.

[6] Wesley Phoa. *An introduction to fibrations, topos theory, the effective topos and modest sets.* LFCS Report Series, 1992.

[7] Michael Barr and Charles Wells. *Category theory for computing science*, volume 49. Prentice Hall New York, 1990.

[8] Francis Borceux. *Handbook of Categorical Algebra 1, Basic Category Theory, vol. 50 of Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1994.

[9] Saunders Mac Lane. *Categories for the working mathematician.* Springer, 1998.

[10] Joachim Lambek and Philip J Scott. *Introduction to higher-order categorical logic*, volume 7. Cambridge University Press, 1988.

[11] Claudio Hermida. *Fibrations, logical predicates and indeterminates.* PhD thesis, University of Edinburgh, 1993.

[12] Claudio Hermida. Some properties of Fib as a fibred 2-category. *Journal of Pure and Applied Algebra*, 134(1):83–109, 1999.

[13] F. William Lawvere. Adjointness in foundations. *Dialectica*, 23(3-4):281–296, 1969.

[14] Paul-André Melliès and Noam Zeilberger. Functors are type refinement systems. In *Proceedings of the 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2015)*, pages 3–16, 2015.

[15] Benjamin C Pierce. *Types and programming languages.* MIT press, 2002.

[16] Jean-Yves Girard. Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur. 1972.

[17] John C Reynolds. Towards a theory of type structure. In *Programming Symposium*, pages 408–425. Springer, 1974.

[18] Christopher Strachey. Fundamental concepts in programming languages. *Higher Order Symbolic Computation*, 13(1-2):11–49, 2000.

[19] John Reynolds. Polymorphism is not set-theoretic. In *Semantics of Data Types*, pages 145 – 156. 1984.

[20] A.M. Pitts. Polymorphism is set theoretic, constructively. In *CTCS*, pages 12 – 39. 1987.

[21] Thierry Coquand and Gérard Huet. The Calculus of Constructions. *Information and Computation*, 76:95 – 120, 1988.

[22] Nick Benton and Chung-Kil Hur. Biorthogonality, step-indexing and compiler correctness. *ACM Sigplan Notices*, 44(9):97–108, 2009.

[23] Amal Ahmed and Matthias Blume. Typed closure conversion preserves observational equivalence. In *ACM Sigplan Notices*, volume 43, pages 157–168. ACM, 2008.

[24] Jason Reed and Benjamin C Pierce. Distance makes the types grow stronger: a calculus for differential privacy. In *ACM Sigplan Notices*, volume 45, pages 157–168. ACM, 2010.

[25] Robert Atkey, Patricia Johann, and Andrew Kennedy. Abstraction and invariance for algebraically indexed types. In *ACM SIGPLAN Notices*, volume 48, pages 87–100. ACM, 2013.

[26] Ryu Hasegawa. Categorical data types in parametric polymorphism. *Mathematical Structures in Computer Science*, 4(01):71–109, 1994.

[27] Philip Wadler. Theorems for free! In *Proceedings of the fourth international conference on Functional programming languages and computer architecture*, pages 347–359. ACM, 1989.

[28] Gordon Plotkin and Martín Abadi. A logic for parametric polymorphism. In *Typed Lambda Calculi and Applications*, pages 361–375. Springer, 1993.

[29] Lars Birkedal and Rasmus E Møgelberg. Categorical models for Abadi and Plotkin's logic for parametricity. *Mathematical Structures in Computer Science*, 15(04):709–772, 2005.

[30] Michael Abbott, Thorsten Altenkirch, Neil Ghani, and Conor McBride. Constructing polymorphic programs with quotient types. In *Mathematics of Program Construction*, pages 2–15. Springer, 2004.

[31] Murdoch J Gabbay and Andrew M Pitts. A new approach to abstract syntax with variable binding. *Formal aspects of computing*, 13(3-5):341–363, 2002.

[32] Robert Atkey. From parametricity to conservation laws, via Noether's theorem. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2014)*, 2014. doi: 10.1145/2535838.2535867.

[33] Joseph Rotman. *An introduction to the theory of groups*, volume 148. Springer Science & Business Media, 2012.

[34] Andrew M Pitts. *Nominal Sets: Names and Symmetry in Computer Science*, volume 57. Cambridge University Press, 2013.

[35] Arthur G Stephenson, Daniel R Mulville, Frank H Bauer, Greg A Dukeman, Peter Norvig, LS LaPiana, PJ Rutledge, D Folta, and R Sackheim. Mars climate orbiter mishap investigation board phase i report. *NASA, Washington, DC*, 1999.

[36] Ronald T. House. A proposal for an extended form of type checking of expressions. *The Computer Journal*, 26(4):366–374, 1983.

[37] R Männer. Strong typing and physical units. *ACM Sigplan Notices*, 21(3):11–20, 1986.

[38] Mitchell Wand and Patrick O'Keefe. Automatic dimensional inference. In *Computational Logic – Essays in Honor of Alan Robinson*, pages 479–483, 1991.

[39] Andrew J. Kennedy. Relational parametricity and units of measure. In *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '97, pages 442–455, New York, NY, USA, 1997. ACM. ISBN 0-89791-853-3. doi: 10.1145/263699.263761. URL `http://doi.acm.org/10.1145/263699.263761`.

[40] Martin Erwig and Margaret Burnett. Adding apples and oranges. In *Practical Aspects of Declarative Languages*, pages 173–191. Springer, 2002.

[41] Andrew John Kennedy. *Programming languages and dimensions.* Number 391. University of Cambridge, Computer Laboratory, 1996.

[42] Ronald Brown and Christopher B Spencer. G-groupoids, crossed modules and the fundamental groupoid of a topological group. *Proc. Indag. Math.*, 79(4):296–302, 1976.

[43] Edgar Buckingham. On physically similar systems; illustrations of the use of dimensional equations. *Physical Review*, 4(4):345–376, 1914.

[44] Ain A Sonin. The physical basis of dimensional analysis. *Department of Mechanical Engineering, MIT, Cambridge, MA*, 2001.

[45] QingMing Ma and John C. Reynolds. Types, abstractions, and parametric polymorphism, part 2. In *MFPS*, pages 1–40, 1992.

[46] Claudio Hermida. Fibrational relational polymorphism. Draft. Available at `http://maggie.cs.queensu.ca/chermida/papers/FibRelPoly.pdf`, 2006.

[47] Brian Dunphy and Uday S Reddy. Parametric limits. In *Logic in Computer Science, 2004. Proceedings of the 19th Annual IEEE Symposium on*, pages 242–251. IEEE, 2004.

"You're still here? It's over! Go home."

Ferris Bueller