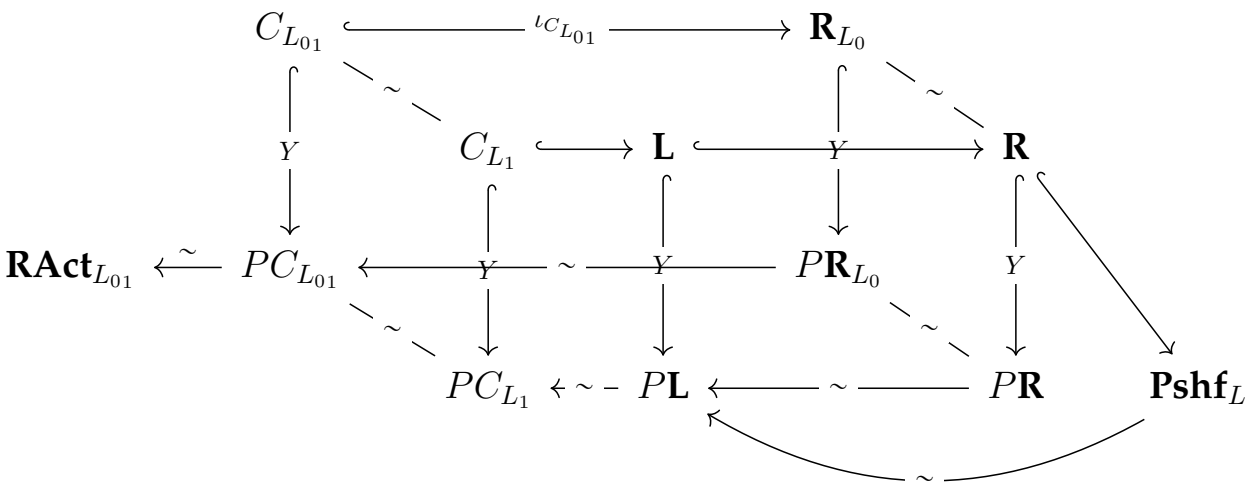
Universal algebra and the λ-calculus

Version of June 25, 2024



Arnoud van der Leer

Universal algebra and the λ-calculus

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Arnoud van der Leer
born in Leiderdorp, the Netherlands

Programming Languages Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Cover picture: A 2-commutative diagram featuring categories that appear in this thesis.

Universal algebra and the λ-calculus

Author: Arnoud van der Leer
Student id: 4485890
Email: a.a.vanderleer@student.tudelft.nl

Abstract

Abstract here.

Thesis Committee:

Chair:	Prof. dr. C. Hair, Faculty EEMCS, TU Delft
Committee Member:	Dr. A. Bee, Faculty EEMCS, TU Delft
Committee Member:	Dr. C. Dee, Faculty EEMCS, TU Delft
University Supervisor:	Ir. E. Ef, Faculty EEMCS, TU Delft

Contents

Chapter 1

Category Theoretic Preliminaries

I will assume a familiarity with the category-theoretical concepts presented in [AW23]. These include categories, functors, isomorphisms, natural transformations, adjunctions, equivalences and limits.

1.1 Notation

For an object c in a category C, I will write $c : C$.

For a morphism f between objects c and c' in a category C, I will write $f : C(c, c')$ or $f : c \to c'$.

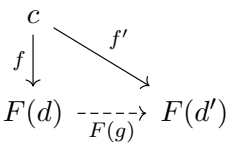
For composition of morphisms $f : C(c, d)$ and $g : C(d, e)$, I will write $f \cdot g$.

For composition of functors $F : A \to B$ and $G : B \to C$, I will write $F \bullet G$.

1.2 Universal Arrows

One concept in category theory that can be used to describe a lot of limits and adjunctions is that of a universal arrow (see for example [Mac98], Part III)

Definition 1. A *universal arrow* from an object $c : C$ to a functor $F : D \to C$ consists of an object $d : D$ and a morphism $f : D(c, F(d))$ such that for every similar pair (d', f'), f' factors uniquely as $f \cdot F(g)$ for some $g : C(d, d')$:



Alternatively, we can characterize universal arrows by their action on hom-sets:

Lemma 1. *Let $F : C \to D$ be a functor and $d : D$ an object. An object $c : C$ and an arrow $f : D(d, F(c))$ form a universal arrow from d to F if and only if the function*

$$(g \mapsto f \cdot F(g)) : C(c, x) \cong D(d, F(x))$$

is a bijection.

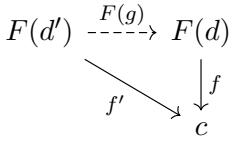
Conversely, for all $c : C$ and $d : D$, every bijection

$$C(c, x) \cong D(d, F(x))$$

that is natural in x arises in this way from some universal arrow $f : D(d, F(c))$.

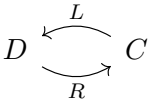
Proof. See [Mac98], Chapter III.2, Proposition 1. □

There is also the dual concept of a universal arrow (d, f) from a functor F to an object $c : C$. Its universal property can be summarized in the following diagram:



1.3 Adjunctions and equivalences

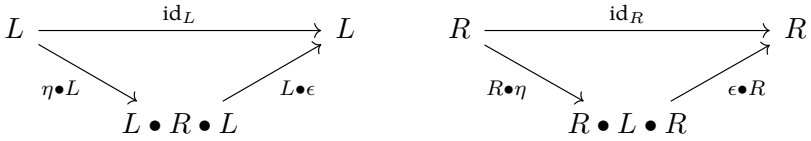
Recall that an *adjunction* $L \dashv R$ is a pair of functors



with natural transformations (the unit and counit)

$$\eta : \mathrm{id}_C \Rightarrow L \bullet R \quad \text{and} \quad \epsilon : R \bullet L \Rightarrow \mathrm{id}_D$$

such that the diagrams



commute (these are called the *triangle identities* or *zigzag identities*). Here the natural transformation $\eta \bullet L : L \bullet R \bullet L$ is the natural transformation η whiskered on the right by L, and the other whiskered transformations are similar.

An alternative characterization (see [Mac98], chapter IV.1, Theorem 2) of an adjunction $L \dashv R$ is as a natural bijection

$$\varphi : D(L(c), d) \xrightarrow{\sim} C(c, R(d)).$$

Naturality means that for all $f : C(c', c)$, $g : D(d, d')$ and $h : D(L(c), d)$,

$$\varphi(L(f) \cdot h \cdot g) = f \cdot \varphi(h) \cdot R(g).$$

Lastly, one can construct an adjunction using universal arrows. This lends itself particularly well for a formalization, where it is often preferable to have as little 'demonstranda' as possible:

Lemma 2. *One can construct an adjunction (L, R, η, ϵ) as above from only the functor $L : C \to D$ and, for each $c : C$, a universal arrow $(R(c), \epsilon_c)$ from L to c.*

Proof. See [Mac98], Chapter IV.1, Theorem 2 (iv). □

1.3.1 Adjoint equivalences

An (adjoint) equivalence of categories has multiple definitions. The one we will use here is the following:

Definition 2. An *adjoint equivalence* between categories C and D is a pair of adjoint functors $L \dashv R$ like above such that the unit $\eta : \mathrm{id}_C \Rightarrow L \bullet R$ and counit $\epsilon : R \bullet L \Rightarrow \mathrm{id}_D$ are isomorphisms of functors.

1.3.2 Weak equivalences

There is also the notion of 'weak equivalence'. In some cases, this is equivalent to an adjoint equivalence (for example, when its domain is univalent).

Definition 3. A functor $F : C \to D$ is called a *weak equivalence* if it is essentially surjective and fully faithful.

1.3.3 Exponential objects

Note that in the category of sets, for all $X, Y : \mathbf{Set}$, we have a set $(X \to Y)$. Also, for all X, Y, Z, there is a (natural) bijection

$$(X \times Y \to Z) \cong (X \to (Y \to Z))$$

which we can also write as

$$\mathbf{Set}(X \times Y, Z) \cong \mathbf{Set}(X, (Y \to Z)).$$

In other words, we have functors $X \mapsto X \times Y$ and $Z \mapsto (Y \to Z)$, and these two form an adjunction. The following generalizes this

Definition 4. A category C has *exponential objects* (or *exponentials*) if for all $c : C$, the functor $c' \mapsto c' \times c$ has a right adjoint, which we denote $d \mapsto d^c$.

Remark 1. It is actually very well possible that a category does not have all exponentials, but it has some objects $c, d, d^c : C$ with a natural bijection

$$C(d' \times c, d) \cong C(d', d^c).$$

Then d^c is still called an exponential object.

1.3.4 Forgetful functors and free objects

In mathematics, we often deal with objects that are 'based on' other objects. For example, a ring is a set with some additional structure. Often, this is a relation between the respective categories (for example, in the case of a displayed category, see Section 6.3), and such a relation gives rise to a *forgetful functor*, that 'forgets' about the additional structure. In the examples of rings and sets, the forgetful functor sends a ring to its underlying set, and a ring morphism to the function between the sets. However, note that there is no formal definition of forgetful functors. The name is more of a way to talk about the perceived relation between the categories.

Definition 5. Given a forgetful functor $F : C \to D$, we define the *free functor* associated to F to be the left adjoint to F, if it exists.

Example 1. Consider the forgetful functor from the category of commutative rings to the category of sets, sending a ring to its underlying set. This has a left adjoint, sending the set $\{1, 2, \ldots, n\}$ to the polynomial ring $\mathbb{Z}[X_1, \ldots, X_n]$, and more generally, sending S to the polynomial ring $\mathbb{Z}[X_s]_{s:S}$. This ring is then called 'the free commutative ring on S'. If S is finite of size n, the ring is also called 'the free commutative ring on n generators'.

The free functor sends a function $f : S \to T$ to the ring morphism $\mathbb{Z}[X_s]_{s:S} \to \mathbb{Z}[X_t]_{t:T}$ that sends X_s to $X_{f(s)}$.

The natural bijection

$$\mathbf{Rng}(\mathbb{Z}[X_s]_{s:S}, R) \cong \mathbf{Set}(S, R)$$

then sends $f : \mathbf{Rng}(\mathbb{Z}[X_s]_{s:S}, R)$ to $s \mapsto f(X_s)$ and $g : \mathbf{Set}(S, R)$ to the morphism that sends X_s to $g(s)$.

However, sometimes, we have a forgetful functor $F : C \to D$, but we cannot give a free functor on the entire category D. In such a case, we might still talk about free 'objects':

Definition 6. Let $F : C \to D$ be a forgetful functor. Given $d : D$, the *free object* on d is a universal arrow (c, f) from d to F.

Remark 2. By [Mac98], Chapter IV.1, Theorem 2 (ii), if we have a free object on every $d : D$, we can piece these together to get a free functor associated to F.

1.4 Yoneda

We can embed a category C fully faithfully into the functor category $PC = [C^{\mathrm{op}}, \mathbf{Set}]$ as follows (see [KS06], Section 1.4):

Definition 7. The *Yoneda embedding* $Y : C \hookrightarrow PC$ is given on objects by $Y(c) = C(-, c)$:

$$Y(c)(d) = C(d, c) \quad \text{and} \quad Y(c)(f)(g) = f \cdot g$$

for $d : C$, $f : C(d, d')$ and $g : C(d', c)$. It sends a morphism $f : C(c, c')$ to the natural transformation $Y(f) : C(-, c) \Rightarrow C(-, c')$ given by

$$Y(f)(d)(g) = g \cdot f$$

for $d : C$ and $g : C(d, c)$.

Now, this embedding has a couple of properties:

Lemma 3. *For any $c : C$ and $F : PC$, we have an equivalence $PC(Y(c), F) \simeq F(c)$, and this equivalence is natural in c and F.*

Proof. See [KS06], Proposition 1.4.3. □

Lemma 4. *The Yoneda embedding functor preserves limits.*

Proof. See [Bor94], Volume 1, Proposition 2.15.5. □

Lemma 5. *The Yoneda embedding functor is cartesian. That is, not only does it preserve binary products, but it also preserves exponential objects* ([*htt*]).

Proof. (**TODO**) □

For a functor between categories $f : C \to D$, given the Yoneda embeddings $Y_C : C \to PC$ and $Y_D : D \to PD$ (we will often omit the subscript C and D), we can create a diagram

$$\begin{array}{ccc} C & \xrightarrow{f} & D \\ \downarrow{\scriptstyle Y} & & \downarrow{\scriptstyle Y} \\ PC & \xleftarrow[f_*^{\mathrm{op}}]{} & PD \end{array}$$

Note that the arrows in this diagram are functors, so objects in a category, instead of elements of a set. Therefore, it does not make sense to talk about 'equality' of the morphisms along the different paths, but we rather talk about isomorphism in the functor category $[C, PC]$. If we have such an isomorphism, we say the diagram '2-commutes':

Lemma 6. *If $f : C \to D$ is a fully faithful functor, the diagram above 2-commutes.*

Proof. For $c, d : C$, since f is fully faithful, we have isomorphisms of sets, given by

$$Y(c)(d) = C(d,c) \xrightarrow[f_{d,c}]{\sim} D(f(d), f(c)) = f_*^{\mathrm{op}}(Y(f(c)))(d).$$

Also, for $g : C(d', d)$, the following diagram commutes

$$\begin{array}{ccc} Y(c)(d) & \xrightarrow{f_{d,c}} & f_*^{\mathrm{op}}(Y(f(c)))(d) \\ \downarrow{\scriptstyle g\cdot -} & & \downarrow{\scriptstyle f_{d',d}(g)\cdot -} \\ Y(c)(d') & \xrightarrow{f_{d',c}} & f_*^{\mathrm{op}}(Y(f(c)))(d') \end{array}$$

so the isomorphism is natural in d and we have $Y(c) \cong f_*^{\mathrm{op}}(Y(f(c)))$ in PC. Lastly, for $g : C(c, c')$ and $d : C$, the following diagram commutes

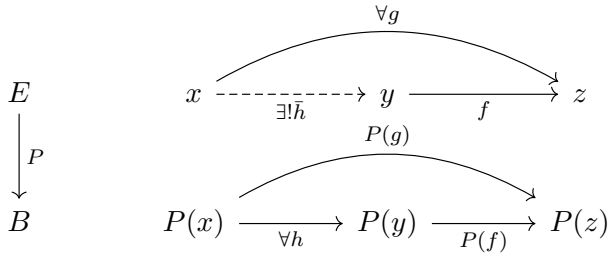
$$\begin{array}{ccc} Y(c)(d) & \xrightarrow{f_{d,c}} & f_*^{\mathrm{op}}(Y(f(c)))(d) \\ \downarrow{\scriptstyle -\cdot g} & & \downarrow{\scriptstyle -\cdot f_{c,c'}(g)} \\ Y(c')(d) & \xrightarrow{f_{d,c'}} & f_*^{\mathrm{op}}(Y(f(c')))(d) \end{array}$$

so the isomorphism is natural in c and we have $Y \cong f \bullet Y \bullet f_*^{\mathrm{op}}$ in $[C, PC]$. □

1.5 Fibrations

Let $P : E \to B$ be a functor. In this case, we will view this as the category E 'lying over' the category B, with for every point $b : B$, a slice $E_b = P^{-1}(B)$ lying 'above' b.

Definition 8. A morphism $f : E(y, z)$ is called *cartesian* if for all $g : E(x, z)$ and $h : B(P(x), P(y))$ with $h \cdot P(f) = P(g)$, there exists $\bar{h} : E(x, y)$ such that $P(\bar{h}) = h$ and $\bar{h} \cdot f = g$, like illustrated in the following diagram from [nLa24b]



Definition 9. P is a *fibration* if for all $y : E$ and morphisms $f : B(x, P(y))$, there exist an object $\bar{x} : E$ and a cartesian morphism $\bar{f} : E(\bar{x}, y)$ such that $P(\bar{x}) = x$ and $P(\bar{f}) = f$:

$$\begin{array}{ccc} E & & \bar{x} \xrightarrow{\exists \bar{f}} y \\ \downarrow{\scriptstyle P} & & \\ B & & x \xrightarrow{\forall f} P(y) \end{array}$$

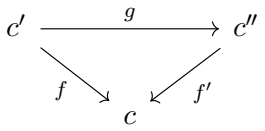
1.6 (Co)slice categories

Given an object in a category $c : C$, the morphisms to and from c constitute the slice and coslice categories

Definition 10. The *slice category* $C \downarrow c$ is the category with as objects the morphisms to c:

$$(C \downarrow c)_0 = \sum_{c':C} C(c', c).$$

The morphisms from (c', f) to (c'', f') are the morphisms $g : c' \to c''$ making the following diagram commute.



The *coslice category* $c \downarrow C$ is similar, but with the morphisms *from* c instead of *to* c:

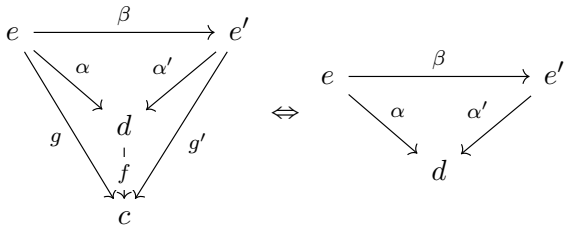
$$(c \downarrow C)_0 = \sum_{c':C} C(c, c').$$

Now, if we have an object in a slice category, we can again look at the slice of the slice category over that object. However, this gives us nothing new:

Lemma 7. *Let (d, f) be an object in the slice category $(C \downarrow c)$. The slice category $((C \downarrow c) \downarrow (d, f))$ is equivalent to $(C \downarrow d)$.*

Proof. An object of $((e, g), \alpha) : ((C \downarrow c) \downarrow (d, f))$ is an object $(e, g) : (C \downarrow c)$, together with a morphism $\alpha : (C \downarrow c)((e, g), (d, f))$. That is, a morphism $\alpha : C(e, d)$ such that the obvious triangle commutes (shown in the diagram below on the far left).

Then a morphism between $((e, g), \alpha)$ and $((e', g'), \alpha')$ is a morphism β between (e, g) and (e', g') that commutes with α and α'. Note that a morphism between (e, g) and (e', g') is a morphism between e and e' that commutes with g and g'.



Now, note that g and g' are completely determined by $g = \alpha \cdot f$ and $g' = \alpha' \cdot f$, so we can leave them out. Also, if β commutes with α and α', it automatically also commutes with g and g'. Therefore, as shown above, we have a correspondence between objects and morphisms

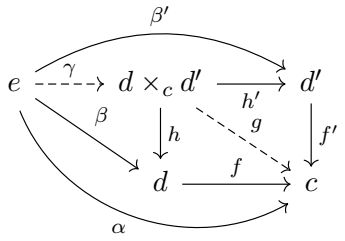
$$\beta : ((e, g), \alpha) \to ((e', g'), \alpha') \Leftrightarrow \beta : (e, \alpha) \to (e', \alpha').$$

□

Also, it turns out that we can derive some structure of the slice categories from the original category:

Lemma 8. *For $(d, f), (d', f') : (C \downarrow c)$, their product in the slice category is given by their pullback / fibered product $d \times_c d'$, together with the induced morphism $g : d \times_c d' \dashrightarrow c$.*

Proof. Consider the diagram below. The fibered product gives 'projections' h and h'. Also, if we have some $(e, \alpha) : (C \downarrow c)$, together with morphisms $\beta : (e, \alpha) \to (d, f)$ and $\beta' : (e, \alpha) \to (d', f')$. Then β and β' commute with f and f', so by the universal property of the fibred product, there exists a unique morphism $\gamma : e \to d \times_c d'$ that makes the triangles with β and h, and with β' and h' commute. Then γ commutes with α and g as well, so it is a morphism in $(C \downarrow c)$. This shows that $(d \times_c d', g)$ has the universal property of the product in $(C \downarrow c)$.



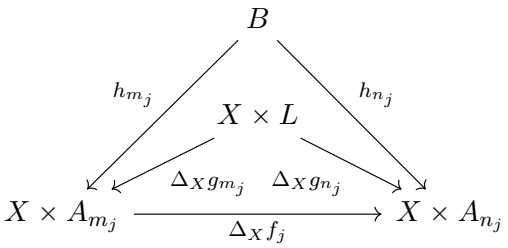
□

For a category C with products, Hyland introduces the notation $\Delta_X Y$ for the element $(X \times Y, p_1) : (C \downarrow X)$, and we will follow his example in this.

In fact, $\Delta_X : C \to C/X$ is a functor, with $\Delta_X(f) = \mathrm{id}_X \times f$ for $f : C(Y, Y')$. This functor preserves the terminal object, products and pullbacks:

Lemma 9. *Δ_X preserves all limits.*

Proof. Take a diagram $((A_i)_i, (f_j)_j)$ in C. Suppose that this has a limit $(L, (g_i)_i) : C$. Now, consider an object $(B, q) : (C \downarrow X)$, together with morphisms $h_i : (C \downarrow X)((B, q), \Delta_X A_i)$, that commute with the $\Delta_X f_j$:



Then the morphisms in $(C \downarrow X)((B, q), \Delta_X L)$, commuting with the $\Delta_X g_j$ and h_j are the morphisms in $C(B, X \times L) \cong C(B, X) \times C(B, L)$ that commute with the projections to X and the $\mathrm{id}_X \times g_j$ and h_j. Since the morphisms in $C(B, X)$ commuting with q and id_X are exactly q, we can forget about this component, and the morphisms we are looking for correspond to the morphisms in $C(B, L)$ that commute with the g_j and $h_j \cdot p_1$. Since $(L, (g_j)_j)$ is a limit, this is a unique morphism. □

Lemma 10. *Δ_X preserves exponential objects:*

Proof. See [Bor94], Volume 3, Lemma 5.8.2. This lemma shows this via the following equivalences:

$$\begin{aligned}(C \downarrow X)((A, f), \Delta_X Y^Z) &\cong C(A, Y^Z) \\ &\cong C(A \times Z, Y) \\ &\cong (C \downarrow X)((A \times X \times Z, \langle f, p_1 \rangle), \Delta_X Y) \\ &\cong (C \downarrow X)((A, f) \times \Delta_X Z, \Delta_X Y)\end{aligned}$$

□

1.7 Dependent products and sums

The following is based loosely on Section 4.1 of [Tay86].

Take a category C. To talk about dependent sums $\sum_{a:A} X_a$ and products $\prod_{a:A} X_a$ in C, we first need some way to construct the family of objects $(X_a)_a$. Of course, we can do this

externally using a set A, and picking an object $X_a : C$ for every element $a : A$. We then have a category of such families C^A, with objects $(X_a)_a$ and morphisms $(f_a)_a : C^A((X_a)_a, (Y_a)_a)$, with $f_a : X_a \to Y_a$. We write C^A because we can view this as just the A-fold power of C. Now, this assignment of categories $A \mapsto C^A$ can be turned into a contravariant (pseudo)functor of 2-categories $\mathbf{Set}^{\mathrm{op}} \to \mathbf{Cat}$. It sends a morphism $f : A \to B$ to the 'relabeling' or 'substitution' functor $C^B \to C^A$, $(X_b)_b \mapsto (X_{f(a)})_a$.

However, there is also an *internal* representation, as a morphism $X \to A$. We can turn the collection of these morphisms over all the $A : C$ simultaneously into a category C^2 (abusing notation a bit, writing 2 for the two-point category $\bullet \to \bullet$). Then taking codomains $(X \to A) \mapsto A$ gives a functor $C^2 \to C$. The fiber of this functor above A is the slice category $(C \downarrow A)$.

In **Set**, the external and internal ways of indexing are actually equivalent, because given a family $(X_a)_a$ we can construct a morphism $f : \sum_{a:A} X_a \to A$ and conversely, we can recover the family $(X_a)_a$ as $(f^{-1}(a))_a$.

Also note that for **Set**, if we consider an indexed family $(X_a)_a$ as some function $X : A \to \mathbf{Set}$, then substitution over $\alpha : B \to A$ is just given by postcomposition $X \circ \alpha : B \to \mathbf{Set}$. It turns out that in the internal representation this arises as the pullback

$$\begin{array}{ccc} \sum_{b:B} X_{\alpha(b)} & \longrightarrow & \sum_{a:A} X_a \\ \downarrow{\scriptstyle \alpha^* f} & \lrcorner & \downarrow{\scriptstyle f} \\ B & \xrightarrow{\quad\alpha\quad} & A \end{array}$$

This can be turned into a pullback or 'substitution' functor α^*, which Taylor calls $\mathsf{P}\alpha$. This turns the functor $\mathbf{Set}^2 \to \mathbf{Set}$ into a fibration. We can construct $\alpha^* : (C \downarrow A) \to (C \downarrow B)$ for any category C with pullbacks and any morphism $\alpha : C(B, A)$. The existence of this pullback makes the functor $C^2 \to C$ into a fibration.

Now, in **Set**, for a family $(X_a)_a$, consider the dependent product $\prod_{a:A} X_a$. Its elements $(x_a)_a$ can be identified with morphisms from the terminal set: $\{\star\} \to \prod_{a:A} X_a$, sending $\star$ to $(x_a)_a$. However, they can also be identified with the morphisms $\varphi : A \to \sum_{a:A} X_a$ that make the following diagram commute, sending a to x_a:

$$\begin{array}{ccc} A & \xrightarrow{\quad\varphi\quad} & \sum_{a:A} X_a \\ & {\scriptstyle \mathrm{id}_A} \searrow \quad \swarrow {\scriptstyle f} & \\ & A & \end{array}$$

These are morphisms in $(\mathbf{Set} \downarrow A)$ from (A, id_A) to $(\sum_{a:A} X_a, f)$. Note that for the terminal morphism $\alpha : A \to \{\star\}$, we have $\mathrm{id}_A = \alpha^*(\mathrm{id}_{\{\star\}})$. To summarize, we have an equivalence

$$(\mathbf{Set} \downarrow A)(\alpha^*(\mathrm{id}_{\{\star\}}), f) \simeq (\mathbf{Set} \downarrow \{\star\})(\mathrm{id}_{\{\star\}}, \prod_{a:A} X_a).$$

Now, given a family of families $((X_b)_{b:B_a})_{a:A}$, we can wonder whether we can construct the family of dependent products $(\prod_{b:B_a} X_b)_a$. In **Set**, this is definitely possible, and from this, we get an equivalence again

$$(\mathbf{Set} \downarrow (\sum_{a:A} B_a))(\alpha^*(\mathrm{id}_A), f) \simeq (\mathbf{Set} \downarrow A)(\mathrm{id}_A, (\prod_{b:B_a} X_b)_a).$$

These equivalences suggest an adjunction $\alpha^* \dashv \prod_{\dots}$. We can use this to define in general

Definition 11. For a category C and a morphism $\alpha : B \to A$, the *dependent product* along α is, if it exists, the right adjoint to the pullback functor:

$$(C \downarrow A) \underset{\prod_\alpha}{\overset{\alpha^*}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} (C \downarrow B)$$

Remark 3. As argued above, we can recover the familiar dependent product $\prod_{a:A} X_a$ of a family $(X_a)_a$ as the dependent product $\prod_\alpha f$ along the terminal morphism $\alpha : A \to I$, with $f : \sum_a X_a \to A$ the internal representation of the family. Here we use the equivalence between $(C \downarrow I)$ and C.

Now we turn our attention to dependent sums. In **Set**, let $(X_a)_a$ and $(B_a)_a$ be two families over A and let $((Y_b)_{b:B_a})_{a:A}$ be a family of families. Let $\alpha : \sum_{a:A} B_a \to A$ be the internal representation of $(B_a)_a$. A family of maps $f_a : (\sum_{b:B_a} Y_b)_a \to X_a$ consists of maps $Y_b \to X_a$ for all $b : B_a$, so these are maps $f_b : Y_b \to X_{\alpha(b)}$. This gives an equivalence

$$(\mathbf{Set} \downarrow A)((\sum_{b:B_a} Y_b)_a, (X_a)_a) \simeq (\mathbf{Set} \downarrow (\sum_{a:A} B_a))((Y_b)_b, \alpha^*((X_a)_a)).$$

This, again, suggests an adjunction which we will use as a definition.

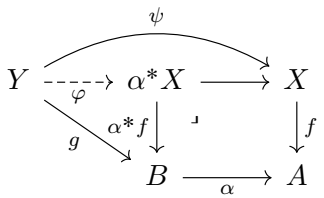
Definition 12. For a category C and a morphism $\alpha : B \to A$, the *dependent sum* along α is, if it exists, the left adjoint to the pullback functor:

$$(C \downarrow A) \underset{\alpha^*}{\overset{\sum_\alpha}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} (C \downarrow B)$$

However, note that the conversion from an external to an internal representation in **Set** already contained a dependent sum, which is no coincidence. It turns out that in practice, we will never have a hard time obtaining dependent sums:

Lemma 11. *Let $\alpha : C(B, A)$ be a morphism in a category. If the pullback functor $\alpha^* : (C \downarrow A) \to (C \downarrow B)$ exists, it has a left adjoint given by postcomposition with α.*

Proof. For morphisms $f : X \to A$, $g : Y \to B$, the universal property of the pullback, with the following diagram



gives an equivalence between morphisms $\varphi : Y \to \alpha^* X$ that commute with g and $\alpha^* f$, and morphisms $\psi : Y \to X$ that commute with g, f and α. In other words:

$$(C \downarrow A)(g \cdot \alpha, f) \simeq (C \downarrow B)(g, \alpha^*(f)),$$

which shows the adjunction. □

Now, let $f : X \to A$ be the internal representation of an indexed family $(X_a)_a$ and let $\alpha : A \to I$ be the terminal projection. We have $\sum_{a:A} X_a = f \circ \alpha : X \to I$. By the equivalence between $(C \downarrow I)$ and C, we see that the dependent sum of the family $(X_a)_a$ is exactly X. Therefore, our attention is mainly focused on the dependent product.

We will close this section with a name for a category that has all dependent products:

Definition 13. A *locally cartesian closed* category is a category C with pullbacks such that each pullback functor α^* has a right adjoint.

Apart from having dependent sums and products, there also is the following theorem that shows the significance of locally cartesian closedness:

Lemma 12. *A category C is locally cartesian closed iff $(C \downarrow A)$ is cartesian closed for each $A : C$.*

Proof. See the end of Section 1.3 of [Fre72]. □

Remark 4. Note that for $X, Y, Z : C$ and $f : C(Y, X)$, the following diagram shows that $f^*\Delta_X Z \cong \Delta_Y Z$:

$$\begin{array}{ccccc} Y \times Z & \xrightarrow{f\times \mathrm{id}_Z} & X \times Z & \xrightarrow{p_2} & Z \\ \downarrow{\scriptstyle p_1} & & \downarrow{\scriptstyle p_1} & & \downarrow{\scriptstyle !} \\ Y & \xrightarrow{f} & X & \xrightarrow{!} & T \end{array}$$

Lemma 13. *For $W, X, Z : C$ and $p_1 : X \times Z \to X$,*

$$\prod_{p_1} \Delta_{X\times Z} W \cong \Delta_X W^Z$$

Proof. First of all, note that $(C \downarrow X \times Z) \cong ((C \downarrow X) \downarrow \Delta_X Z)$. Also note that the composite morphism $(X \times Z) \times W \xrightarrow{p_1} X \times Z \xrightarrow{p_1} X$ is the element $\Delta_X(W \times Z) : (C \downarrow X)$.

By Proposition 1.34 in [Fre72], $\prod_{p_1} \Delta_{X\times Z} W$ is given as the following pullback:

$$\begin{array}{ccc} \prod_{p_1} \Delta_{X\times Z} W & \longrightarrow & (\Delta_X Z \times W)^{\Delta_X Z} \\ \downarrow & & \downarrow{\scriptstyle (\Delta_X p_1)^{\Delta_X Z}} \\ X & \longrightarrow & (\Delta_X Z)^{\Delta_X Z} \end{array}$$

By Lemma 10, Δ_X preserves exponential objects, so the morphism on the right is $\Delta_X p_1^Z : (C \downarrow X)(\Delta_X(Z \times W)^Z, \Delta_X Z^Z)$. However, we have an isomorphism $(Z \times W)^Z \cong Z^Z \times Z^W$, and then the morphism on the right becomes

$$\Delta_X p_1 : (C \downarrow X)(\Delta_X(Z^Z \times Z^W), \Delta_X Z^Z)$$

We also have an isomorphism $X \cong \Delta_X T$. Then by Lemma 9 and Remark 4, the pullback of this diagram is $\Delta_X W^Z$:

$$\begin{array}{ccc} \Delta_X W^Z & \longrightarrow & \Delta_X(Z^Z \times W^Z) \\ \downarrow & & \downarrow{\scriptstyle \Delta_X p_1} \\ \Delta_X T & \longrightarrow & \Delta_X Z^Z \end{array}$$

□

1.8 (Weakly) terminal objects

Definition 14. If a category has an object t, such that there is a (not necessarily unique) morphism to it from every other object in the category, t is said to be a *weakly terminal object*.

Definition 15. Let C be a category with terminal object t. For an object $c : C$, a *global element* of c is a morphism $f : C(t, c)$.

1.9 Kan Extensions

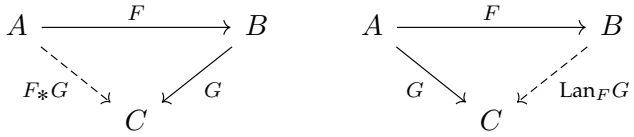
One of the most general and abstract concepts in category theory is the concept of *Kan extensions*. In [Mac98], Section X.7, MacLane notes that

"The notion of Kan extensions subsumes all the other fundamental concepts of category theory."

In this thesis, we will use left Kan extension a handful of times. It comes in handy when we want to extend a functor along another functor in the following way:

Let A, B and C be categories and let $F : A \to B$ be a functor.

Definition 16. Precomposition gives a functor between functor categories $F_* : [B, C] \to [A, C]$. If F_* has a left adjoint, we will denote call this adjoint functor the *left Kan extension* along F and denote it $\mathrm{Lan}_F : [A, C] \to [B, C]$.



Analogously, when F_* has a right adjoint, one calls this the *right Kan extension* along F and denote it $\mathrm{Ran}_F : [A, C] \to [B, C]$.

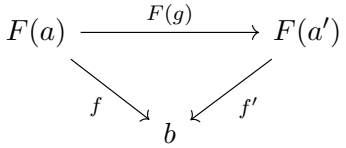
If a category has limits (resp. colimits), we can construct the right (resp. left) Kan extension in a 'pointwise' fashion (see Theorem X.3.1 in [Mac98] or Theorem 2.3.3 in [KS06]). Below, I will outline the parts of the construction that we will need explicitly in this thesis.

Lemma 14. *If C has colimits, Lan_F exists.*

Proof. First of all, for objects $b : B$, we take

$$\mathrm{Lan}_F G(b) := \mathrm{colim}\left((F \downarrow b) \to A \xrightarrow{G} C\right).$$

Here, $(F \downarrow b)$ denotes the comma category with as objects the morphisms $B(F(a), b)$ for all $a : A$, and as morphisms from $f : B(F(a), b)$ to $f' : B(F(a'), b)$ the morphisms $g : A(a, a')$ that make the diagram commute:



and $(F \downarrow b) \to A$ denotes the projection functor that sends $f : B(F(a_1), b)$ to a_1.

Now, a morphism $h : B(b, b')$ gives a morphism of diagrams, sending the $F(a)$ corresponding to $f : B(G(a), b)$ to the $F(a)$ corresponding to $f \cdot h : B(G(a), b')$. From this, we get a morphism $\mathrm{Lan}_F G(h) : C(\mathrm{Lan}_F G(b), \mathrm{Lan}_F G(b'))$.

The unit of the adjunction is a natural transformation $\eta : \mathrm{id}_{[A,C]} \Rightarrow \mathrm{Lan}_F \bullet F_*$. We will define this pointwise, for $G : [A, C]$ and $a : A$. Our diagram contains the $G(a)$ corresponding to $\mathrm{id}_{F(a)} : (F \downarrow F(a))$ and the colimit cocone gives a morphism

$$\eta_G(a) : C(G(a), \mathrm{Lan}_F G(F(a))),$$

the latter being equal to $(\mathrm{Lan}_F \bullet F_*)(G)(a)$.

The counit of the adjunction is a natural transformation $\epsilon : F_* \bullet \mathrm{Lan}_F \Rightarrow \mathrm{id}_{[B,C]}$. We will also define this pointwise, for $G : [B, C]$ and $b : B$. The diagram for $\mathrm{Lan}_F(F_* G)(b)$ consists of

$G(F(a))$ for all $f : B(F(a), b)$. Then, by the universal property of the colimit, the morphisms $G(f) : C(G(F(a)), G(b))$ induce a morphism

$$\epsilon_G(b) : C(\text{Lan}_F(F_*G)(b), G(b)).$$

□

Lemma 15. *If $F : A \to B$ is a fully faithful functor, and C is a category with colimits, $\eta : id_{[A,C]} \Rightarrow Lan_F \bullet F_*$ is a natural isomorphism.*

Proof. To show that η is a natural isomorphism, we have to show that $\eta_G(a') : G(a') \Rightarrow \text{Lan}_F G(F(a'))$ is an isomorphism for all $G : [A, C]$ and $a' : A$. Since a left adjoint is unique up to natural isomorphism ([AW23], Exercise 153), we can assume that $\text{Lan}_F G(F(a'))$ is given by

$$\text{colim}((F \downarrow F(a')) \to A \xrightarrow{G} C).$$

Now, the diagram for this colimit consists of $G(a)$ for each arrow $f : B(F(a), F(a'))$. Since F is fully faithful, we have $f = F(\overline{f})$ for some $\overline{f} : A(a, a')$. If we now take the arrows $G(\overline{f}) : C(G(a), G(a'))$, the universal property of the colimit gives an arrow

$$\varphi : C(\text{Lan}_F G(F(a')), G(a'))$$

which constitutes an inverse to $\eta_G(a')$. The proof of this revolves around properties of the colimit and its (induced) morphisms. □

Remark 5. In the same way, if C has limits, ϵ is a natural isomorphism.

Corollary 1. *If C has limits or colimits, precomposition of functors $[B, C]$ along a fully faithful functor is (split) essentially surjective.*

Proof. For each $G : [A, C]$ we take $\text{Lan}_F G : [B, C]$, and we have $F_*(\text{Lan}_F G) \cong G$. □

Corollary 2. *If C has colimits (resp. limits), left (resp. right) Kan extension of functors $[A, C]$ along a fully faithful functor is fully faithful.*

Proof. Since left Kan extension along F is the left adjoint to precomposition, we have

$$[A, C](\text{Lan}_F G, \text{Lan}_F G') \cong [B, C](G, F_*(\text{Lan}_F G')) \cong [B, C](G, G').$$

□

1.10 Coends

This section is based on Section 1.2 of [Rie14].

In this thesis, we will encounter co-ends a couple of times, so we will introduce them here.

Definition 17. Let C, D be categories and $F : C^{\text{op}} \times C \to D$ a functor. We define the *coend* $\int^{c:C} F(c, c)$ to be the colimit

$$\coprod_{f:C(a,b)} F(b, a) \underset{F(\text{id}_a, f)}{\overset{F(f, \text{id}_b)}{\rightrightarrows}} \coprod_{c:C} F(c, c) \dashrightarrow \int^C F$$

Remark 6. An alternative way to phrase this, is that $\int^C F : D$ is an object, equipped with arrows $F(c, c) \to \int^C F$ such that for all $f : C(a, b)$, the following diagram must commute

$$\begin{array}{ccc} F(b,a) & \xrightarrow{F(f,\mathrm{id}_a)} & F(a,a) \\ \downarrow{\scriptstyle F(\mathrm{id}_b,f)} & & \downarrow \\ F(b,b) & \longrightarrow & \int^C F \end{array}$$

and such that for any other $G : D$ with the same properties, we have a unique morphism $\int_C F \to G$, making the triangles commute

$$\begin{array}{ccc} F(c,c) & \longrightarrow & \int_C F \\ & \searrow & \downarrow \\ & & G \end{array}$$

Remark 7. Of course, a co-end is actually the dual notion of an end, which can be defined as the equalizer of the diagram above, but with the arrows reversed.

Remark 8. Left Kan extension can be expressed as a coend:

$$\mathrm{Lan}_F G(b) = \int^{a:A} D(Fa, b) \cdot Ga$$

where $S \cdot X$ for S a set and $X : C$ denotes the 'copower'. In most cases, we will have

$$S \cdot X = \coprod_{s:S} X.$$

1.11 The Karoubi envelope

Let C be a category. If we have a retraction-section pair $c \underset{s}{\overset{r}{\rightleftarrows}} d$ we have (by definition) $s \cdot r = \mathrm{id}_d$. On the other hand, $r \cdot s : c \to c$ is an idempotent morphism, since $r \cdot s \cdot r \cdot s = r \cdot s$. Conversely, we can wonder whether for some idempotent morphism $a : c \to c$, we can find a retraction-section pair (r, s) such that $a = r \cdot s$. If this is the case, we say that the idempotent a *splits*. If a does not split, we can wonder whether we can find an embedding $\iota_C : C \hookrightarrow \overline{C}$ such that the idempotent $\iota_C(a) : \iota_C(c) \to \iota_C(c)$ does split. This is one way to arrive at the *Karoubi envelope*.

Definition 18. We define the category $\overline{C}$. The objects of $\overline{C}$ are tuples (c, a) with $c : C$, $a : C(c, c)$ such that $a \cdot a = a$. The morphisms between (c, a) and (d, b) are morphisms $f : C(a, b)$ such that $a \cdot f \cdot b = f$. The identity morphism on (c, a) is given by a and $\overline{C}$ inherits morphism composition from C.

This category is called the *Karoubi envelope*, the *idempotent completion*, the *category of retracts*, or the *Cauchy completion* of C.

Remark 9. Note that for a morphism $f : \overline{C}((c, a), (d, b))$,

$$a \cdot f = a \cdot a \cdot f \cdot b = a \cdot f \cdot b = f$$

and in the same way, $f \cdot b = f$.

Definition 19. We have an embedding $\iota_C : C \to \overline{C}$, sending $c : C$ to (c, id_c) and $f : C(c, d)$ to f.

Lemma 16. *Every object $c : \overline{C}$ is a retract of $\iota_C(c_0)$ for some $c_0 : C$.*

Proof. Note that $c = (c_0, a)$ for some $c_0 : C$ and an idempotent $a : c \to c$. We have morphisms $\iota_C(c) \underset{a_\leftarrow}{\overset{a_\rightarrow}{\rightleftarrows}} (c, a)$, both given by a. We have $a_\leftarrow \cdot a_\rightarrow = a = \mathrm{id}_{(c,a)}$, so (c, a) is a retract of $\iota_C(c)$. □

Lemma 17. *In $\overline{C}$, every idempotent splits.*

Proof. Take an idempotent $e : \overline{C}(c, c)$. Note that c is given by an object $c_0 : C$ and an idempotent $a : C(c_0, c_0)$. Also, e is given by some idempotent $e : C(c_0, c_0)$ with $a \cdot e \cdot a = e$.

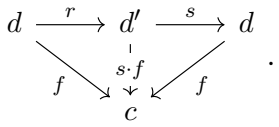
Now, we have $(c_0, e) : \overline{C}$ and morphisms $(c_0, a) \underset{e_\leftarrow}{\overset{e_\rightarrow}{\rightleftarrows}} (c_0, e)$, both given by e. We have $e_\leftarrow \cdot e_\rightarrow = e = \mathrm{id}_{(c_0,e)}$, so (c_0, e) is a retract of (c_0, a). Also, $e = e_\rightarrow \cdot e_\leftarrow$, so e is split. □

Remark 10. Note that the embedding is fully faithful, since

$$\overline{C}((c, \mathrm{id}_c), (d, \mathrm{id}_d)) = \{f : C(c, d) \mid \mathrm{id}_c \cdot f \cdot \mathrm{id}_d = f\} = C(c, d).$$

Remark 11. Let D be a category. Suppose that we have a retraction-section pair in D, given by $d \underset{s}{\overset{r}{\rightleftarrows}} d'$. Now, suppose that we have an object $c : D$ and a morphism f with $(r \cdot s) \cdot f = f$. Then we get a morphism $s \cdot f : d' \to c$ such that f factors as $r \cdot (s \cdot f)$. Also, for any g with $r \cdot g = f$, we have

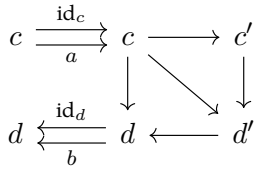
$$g = s \cdot r \cdot g = s \cdot f.$$



Therefore, d' is the equalizer of $d \underset{r \cdot s}{\overset{\mathrm{id}_d}{\rightrightarrows}} d$. In the same way, it is also the coequalizer of this diagram.

Now, note that if we have a coequalizer c' of id_c and a, and an equalizer d' of id_d and b, the universal properties of these give an equivalence

$$D(c', d') \cong \{f : D(c, d') \mid a \cdot f = f\} \cong \{f : D(c, d) \mid a \cdot f = f = f \cdot b\}.$$



Since a functor preserves retracts, and since every object of $\overline{C}$ is a retract of an object in C, one can lift a functor from C (to a category with (co)equalizers) to a functor on $\overline{C}$.

For convenience, the lemma below works with pointwise left Kan extension using colimits, but one could also prove this using just (co)equalizers (or right Kan extension using limits).

Lemma 18. *Let D be a category with colimits. We have an adjoint equivalence between $[C, D]$ and $[\overline{C}, D]$.*

Proof. We already have an adjunction $\mathrm{Lan}_{\iota_C} \dashv \iota_{C*}$. Also, since ι_C is fully faithful, we know that η is a natural isomorphism. Therefore, we only have to show that ϵ is a natural isomorphism. That is, we need to show that $\epsilon_G(c, a) : D(\mathrm{Lan}_{\iota_C}(\iota_{C*}G)(c, a), G(c, a))$ is an isomorphism for all $G : [\overline{C}, D]$ and $(c, a) : \overline{C}$.

One of the components in the diagram of $\mathrm{Lan}_{\iota_C}(\iota_{C*}G)(c,a)$ is the $\iota_{C*}G(c) = G(c, \mathrm{id}_c)$ corresponding to $a : \iota_C(c) \to (c,a)$. This component has a morphism into our colimit

$$\varphi : C(G(\iota_C(c)), \mathrm{Lan}_{\iota_C}(\iota_{C*}G)(c,a)).$$

Note that we can view a as a morphism $a : \overline{C}((c,a), \iota_C(c))$. This gives us our inverse morphism

$$G(a) \cdot \varphi : C(G(c,a), \mathrm{Lan}_{\iota_C}(\iota_{C*}G)(c,a)).$$

□

Lemma 19. *The formation of the opposite category commutes with the formation of the Karoubi envelope.*

Proof. An object in $\overline{C^{\mathrm{op}}}$ is an object $c : C^{\mathrm{op}}$ (which is just an object $c : C$), together with an idempotent morphism $a : C^{\mathrm{op}}(c,c) = C(c,c)$. This is the same as an object in $\overline{C}^{\mathrm{op}}$.

A morphism in $\overline{C^{\mathrm{op}}}((c,a),(d,b))$ is a morphism $f : C^{\mathrm{op}}(c,d) = C(d,c)$ such that

$$b \cdot_C f \cdot_C a = a \cdot_{C^{\mathrm{op}}} f \cdot_{C^{\mathrm{op}}} b = f.$$

A morphism in $\overline{C}^{\mathrm{op}}((c,a),(d,b)) = \overline{C}((d,b),(c,a))$ is a morphism $f : C(d,c)$ such that $b \cdot f \cdot a = f$.

Now, in both categories, the identity morphism on (c,a) is given by a.

Lastly, $\overline{C^{\mathrm{op}}}$ inherits morphism composition from C^{op}, which is the opposite of composition in C. On the other hand, composition in $\overline{C}^{\mathrm{op}}$ is the opposite of composition in $\overline{C}$, which inherits composition from C. □

Corollary 3. *As the category* **Set** *is cocomplete, we have an equivalence between the category of presheaves on C and the category of presheaves on $\overline{C}$:*

$$[C^{op}, \mathbf{Set}] \cong [\overline{C^{op}}, \mathbf{Set}] \cong [\overline{C}^{op}, \mathbf{Set}].$$

Now, there is also another (classically equivalent) definition of the Karoubi envelope:

Definition 20. Let $Y : C \to PC$ be the Yoneda embedding of C into its category of presheaves. We define the category $\hat{C}$ as the full subcategory of PC consisting of objects $F : PC$ such that there exists an object $c : C$ and a retraction of $Y(c)$ onto F. Note that the existence of c and the retraction is *mere existence* (see the chapter on Univalent Foundations) because we need this to be a subcategory of PC.

Lemma 20. *$\hat{C}$ is the Rezk completion of $\overline{C}$.*

Proof. First of all, note that $\hat{C}$ is univalent, since it is a presheaf category (see Theorem 4.5 in [AKS15]). So we must show that we have a weak equivalence from $\overline{C}$ to $\hat{C}$.

Secondly, by Corollary 3 we have an adjoint equivalence ${\iota^{\mathrm{op}}_C}_* : P\overline{C} \xrightarrow{\sim} PC$ and by Lemma 6, the following diagram 2-commutes:

$$\begin{array}{ccc} C & \xrightarrow{\iota_C} & \overline{C} \\ \downarrow{\scriptstyle Y} & & \downarrow{\scriptstyle Y} \\ PC & \xleftarrow[{\iota^{\mathrm{op}}_C}_*]{\sim} & P\overline{C} \end{array}$$

By Lemma 16, every object of $\overline{C}$ is a retract of some object in c. Therefore, every object in the image of $\overline{C}$ inside PC is a retract of some object in the image of C, so we can turn $Y \bullet {\iota^{\mathrm{op}}_C}_* : \overline{C} \to PC$ into a fully faithful embedding into $\hat{C}$.

Now, to show that it is essentially surjective, take an object $F : \hat{C}$. That is, take a retract $F \underset{r}{\overset{s}{\leftrightarrows}} Y(c)$. Note that $r \cdot s$ is an idempotent on $Y(c)$. Since Y is a fully faithful embedding, it gives an equivalence on the hom-sets, so we have an element $e : C(c, c)$ that maps to $r \cdot s : PC(Y(c), Y(c))$. We will show that (c, e) is our preimage in $\overline{C}$ of F.

Also, (c, e) is a retract of $\iota_C(c)$ with section and retraction e. Therefore, $\iota_C^{\text{op}} \bullet Y(c, e)$ is a retract of $\iota_C^{\text{op}} \bullet Y(\iota_C(c))$ with section and retraction $\iota_C^{\text{op}} \bullet Y(e)$. Composing these gives an idempotent

$$\iota_C^{\text{op}} \bullet Y(e) : d \mapsto f \mapsto f \cdot e$$

on $\iota_C^{\text{op}} \bullet Y(\iota_C(c))$. Transporting this along the isomorphism with $Y(c)$, we get an idempotent

$$d \mapsto f \mapsto \iota_C^{-1}(\iota_C(f) \cdot e)$$

on $Y(c)$. Note that we can view e both as an idempotent on c and on $\iota_C(e)$, and that $\iota_C(e) = e$, so $\iota_C^{-1}(\iota_C(f) \cdot e) = f \cdot e$.

Therefore, the idempotent on $Y(c)$ is exactly $Y(e) = r \cdot s$. By Remark 11, $\iota_C^{\text{op}} \bullet Y(c, e)$ is the equalizer of $\text{id}_{Y(c)}$ and $r \cdot s$. Note that by the same remark, F is the equalizer of $\text{id}_{Y(c)}$ and $r \cdot s$ as well. Since the equalizer is unique up to unique isomorphism, we have an isomorphism $F \cong \iota_C^{\text{op}} \bullet Y(c, e)$ and we see that $Y \bullet \iota_C^{\text{op}}{}_*$ is essentially surjective. □

Remark 12. As shown above, $\overline{C}$ and $\hat{C}$ have a very strong relation. In classical mathematics, they are even equivalent. However, since $\overline{C}$ is not univalent, we do not have a full equivalence in univalent foundations and our choice of definition will have consequences. One one hand, $\hat{C}$ is univalent. On the other hand, every object of $\overline{C}$ is uniquely associated with one idempotent morphism in C.

In this thesis, I will choose to interpret the 'category of retracts' or 'Karoubi envelope' to mean $\overline{C}$. This is because Dana Scott and Paul Taylor perform most of their constructions explicitly using the (idempotent) morphisms of C, which is closest to working in $\overline{C}$.

It is however very well possible that most of the proofs can be made to work in $\hat{C}$ as well. For future research, it would be interesting to see what the differences are between working in $\overline{C}$ and $\hat{C}$ for these proofs.

1.12 Monoids as categories

Take a monoid M.

Definition 21. We can construct a category C_M with $C_{M0} = \{\star\}$, $C_M(\star, \star) = M$. The identity morphism on $\star$ is the identity $1 : M$. The composition is given by multiplication $g \cdot_{C_M} f = f \cdot_M g$.

Remark 13. Actually, we have a functor from the category of monoids to the category of set-categories (categories whose object type is a set).

A monoid morphism $f : M \to M'$ is equivalent to a functor $F_f : C_M \to C_{M'}$. Any functor between C_M and $C_{M'}$ sends $\star_M$ to $\star_{M'}$. The monoid morphism manifests as $F_f(m) = f(m)$ for $m : C_M(\star, \star) = M$.

Lemma 21. *An isomorphism of monoids gives an (adjoint) equivalence of categories.*

Proof. Given an isomorphism $f : M \to M'$. Then we have functors $F_f : C_M \to C_{M'}$ and $F_{f^{-1}} : C_{M'} \to C_M$. Take the identity natural transformations $\eta : \text{id}_{C_M} \Rightarrow F_f \bullet F_{f^{-1}}$ and $\epsilon : F_{f^{-1}} \bullet F_f \Rightarrow \text{id}_{C_{M'}}$. Of course these are natural isomorphisms. □

Definition 22. A *right monoid action* of M on a set X is a function $X \times M \to X$ such that for all $x : X$, $m, m' : M$,

$$x1 = x \qquad \text{and} \qquad (xm)m' = x(m \cdot m').$$

Definition 23. A *morphism* between sets X and Y with a right M-action is an M-equivariant function $f : X \to Y$: a function such that $f(xm) = f(x)m$ for all $x : X$ and $m : M$.

These, together with the identity and composition from **Set**, constitute a category $\mathbf{RAct}_M$ of right M-actions.

Lemma 22. *Presheaves on C_M are equivalent to sets with a right M-action.*

Proof. This correspondence sends a presheaf F to the set $F(\star)$, and conversely, the set X to the presheaf F given by $F(\star) := X$. The M-action corresponds to the presheaf acting on morphisms as $xm = F(m)(x)$. A morphism (natural transformation) between presheaves $F \Rightarrow G$ corresponds to a function $F(\star) \to G(\star)$ that is M-equivariant, which is exactly a monoid action morphism. □

Remark 14. Since the category of sets with an M-action is equivalent to a presheaf category, it has all limits. However, we can make this concrete. The set of the product $\prod_i X_i$ is the product of the underlying sets. The action is given pointwise by $(x_i)_i m = (x_i m)_i$.

Note that the initial set with M-action is $\{\star\}$, with action $\star m = \star$.

Lemma 23. *The global elements of a set with right M-action correspond to the elements that are invariant under the M-action.*

Proof. A global element of X is a morphism $\varphi : \{\star\} \to X$ such that for all $m : M$, $\varphi(\star)m = \varphi(\star m) = \varphi(\star)$. Therefore, it is given precisely by the element $\varphi(\star) : X$, which must be invariant under the M-action. □

Lemma 24. *The category C of sets with an M-action has exponentials.*

Proof. Given sets with M-action X and Y. Consider the set $C(M \times X, Y)$ with an M-action given by $\phi m'(m, x) = \phi(m'm, x)$. This is the exponential object X^Y, with the (universal) evaluation morphism $X \times X^Y \to Y$ given by $(x, \phi) \mapsto \phi(1, x)$. Explicitly, we get a natural isomorphism $\psi : \mathbf{RAct}_M(Z \times Y, X) \xrightarrow{\sim} \mathbf{RAct}_M(Z, X^Y)$ given by

$$\psi(f)(z)(m, y) = f(zm, y) \quad \text{and} \quad \psi^{-1}(g)(z, y) = g(z)(1, y).$$

□

Definition 24. We can view M as a set U_M with right M-action $mn = m \cdot_M n$ for $m : U_M$ and $n : M$.

1.12.1 Extension and restriction of scalars

Let $\varphi : M \to M'$ be a morphism of monoids.

Remember that sets with a right monoid action are equivalent to presheaves on the monoid category. Also, φ is equivalent to a functor between the monoid categories. The following is a specific case of the concepts in the section about Kan extension:

Lemma 25. *We get a restriction of scalars functor φ^* from sets with a right M'-action to sets with a right M-action.*

Proof. Given a set X with right M'-action, take the set X again, and give it a right M-action, sending (x, m) to $x\varphi(m)$.

On morphisms, send an M'-equivariant morphism $f : X \to X'$ to the M-equivariant morphism $f : X \to X'$. □

Since **Set** has colimits, and restriction of scalars corresponds to precomposition of presheaves (on $C_{M'}$), we can give it a left adjoint. This is the (pointwise) left Kan extension, which boils down to a very concrete definition, reminiscent of a tensor product:

Lemma 26. *We get an extension of scalars functor φ_* from sets with a right M-action to sets with a right M'-action.*

Proof. Given a set X with right M-action. Take $Y = X \times M'/\sim$ with the relation $(xm, m') \sim (x, f(m) \cdot m')$ for $m : M$. This has a right M'-action given by $(x, m')n' = (x, m'n')$.

On morphisms, it sends the m-equivariant $f : X \to X'$ to the morphism $(x, m') \mapsto (f(x), m')$. □

Lemma 27. *For U_M the set M with right M-action, we have $\varphi_*(U_M) \cong U_{M'}$.*

Proof. The proof relies on the fact that for all $m : U_M$ and $m' : M'$, we have

$$(m, m') \sim (1, \varphi(m)m').$$

□

Consider the category D with $D_0 = M'$ and

$$D(m', \overline{m}') = \{m : M \mid \varphi(m) \cdot m' = \overline{m}'\}.$$

Lemma 28. *Suppose that D has a weakly terminal element. Then for I_M the terminal object in the category of sets with a right M-action, we have $\varphi_*(I_M) \cong I_{M'}$.*

Proof. If D has a weakly terminal object, there exists $m_0 : M'$ such that for all $m' : M'$, there exists $m : M$ such that $\varphi(m) \cdot m' = m_0$.

The proof relies on the fact that every element of $\varphi_*(I_M)$ is given by some $(\star, m')$, but then

$$(\star, m') = (\star \cdot m, m') \sim (\star, \varphi(m) \cdot m') = (\star, m_0),$$

so $\varphi_*(I_M)$ has exactly 1 element. □

Remark 15. For φ_* to preserve terminal objects, we actually only need D to be connected. The fact that $\varphi_*(I_M)$ is a quotient by a symmetric and transitive relation then allows us to 'walk' from any $(\star, m_1')$ to any other $(\star, m_2')$ in small steps.

For any $m_1', m_2' : M'$, consider the category $D_{m_1', m_2'}$, given by

$$D_{m_1', m_2', 0} = \{(m', m_1, m_2) : M' \times M \times M \mid m_i' = \varphi(m_i) \cdot m'\}$$

and

$$D_{m_1', m_2'}((m', m_1, m_2), (\overline{m}', \overline{m}_1, \overline{m}_2)) = \{m : M \mid \varphi(m) \cdot m' = \overline{m}', m_i = \overline{m}_i \cdot m\}.$$

Lemma 29. *Suppose that $D_{m_1', m_2'}$ has a weakly terminal object for all $m_1', m_2' : M'$. Then for sets A and B with right M-action, we have $\varphi_*(A \times B) \cong \varphi_*(A) \times \varphi_*(B)$.*

Proof. Now, any element in $\varphi_*(A) \times \varphi_*(B) = (A \times M'/\sim) \times (B \times M'/\sim)$ is given by some (a, m_1', b, m_2').

The fact that $D_{m_1', m_2'}$ has a weakly terminal object means that we have some $\overline{m}' : M'$ and $\overline{m}_1, \overline{m}_2 : M$ with $m_i' = \varphi(\overline{m}_i) \cdot \overline{m}'$. Therefore,

$$(a, m_1', b, m_2') = (a, \varphi(\overline{m}_1) \cdot \overline{m}', b, \varphi(\overline{m}_2) \cdot \overline{m}') \sim (a\overline{m}_1, \overline{m}', b\overline{m}_2, \overline{m}'),$$

so this is equivalent to some element in $\varphi_*(A \times B) = (A \times B \times M'/\sim)$. Note that this trivially respects the right M'-action.

The fact that $(\overline{m}', \overline{m}_1, \overline{m}_2)$ is weakly terminal also means that for all $m' : M'$ and $m_1, m_2 : M$ with $m'_i = \varphi(m_i) \cdot m'$, there exists $m : M$ such that $\varphi(m) \cdot m' = \overline{m}'$ and $m_i = \overline{m}_i \cdot m$. This means that the equivalence that we established is actually well-defined: equivalent elements in $\varphi_*(A) \times \varphi_*(B)$ are sent to equivalent elements in $\varphi_*(A \times B)$.

Therefore, we have an isomorphism $\psi : \varphi^*(A) \times \varphi^*(B) \xrightarrow{\sim} \varphi^*(A \times B)$. Now we only need to show that the projections are preserved by this isomorphism. To that end, take $x = (a, m'_1, b, m'_2) \sim (a\overline{m}_1, \overline{m}', b\overline{m}_2, \overline{m}') : \varphi^*(A) \times \varphi^*(B)$. We have

$$\varphi^*(\pi_1)(\psi(x)) = (a\overline{m}_1, \overline{m}') = \pi'_1(x).$$

In the same way, $\varphi^*(\pi_2) \circ \psi = \pi'_2$ and this concludes the proof. □

Chapter 2

Univalent Foundations

2.1 Dependent type theory

Univalent foundations takes place in a framework of type theory. In this section, we will quickly introduce some topics that we will need in subsequent sections.

First of all, *type theory* is the study of 'type systems'. A *type system* is a collection of *terms* or *elements*, each of which has a corresponding type. A type is much like a set in set theory in that it has elements (for example, true : Bool or $1 : \mathbb{N}$), but there are important differences. First of all, in type theory, terms are declared together with their type, so every term has exactly one type, whereas in set theory, something can be an element of multiple sets, like $5 \in \mathbb{N}$ and $5 \in \mathbb{R}$. Secondly, equality in type theory can work a bit differently than in set-based mathematics, but I will cover this in the next section.

Of course, in computer science, we are very familiar with type systems. In most programming languages, values explicitly (e.g. in Java) or implicitly (e.g. in Python) have a type associated to them. For example, `"Hello, World"` is of type *string*, `true` and `false` are of type *boolean*, `1` is of type *integer* and `-5.8` is of type *floating point number*. And here, already, we see some subtleties: due to their different internal representations, many programming languages distinguish between integers and floating point numbers, even though every integer can be considered to be a floating point number. Usually, programming languages resolve this by offering methods (sometimes in the form of *coercions*) to convert between the two types (discarding what comes after the decimal point when converting a floating point number to an integer). Another coercion that occurs sometimes is the conversion of a *character* like `a`, `1` or `&` to an integer and back.

So if we have types and elements, are types also elements of some type? This is a tricky question, because if we say that there exists a type **Type** which contains all types (and therefore, itself), we introduce inconsistency in our type system (see [Hur95]). This is just like having a 'set of all sets', which results in problematic questions like "does its subset, containing only the sets which do not contain itself, contain itself?" This is usually solved by either stating that types do not themselves have a type, or by assuming the existence of *type universes* $U_1, U_2, \ldots$, such that every type is an element of some U_i. Note that universes are allowed to be inclusive or not: we may have $U_n \subseteq U_{n+1}$. In this thesis, we will not explicitly mention particular universes. We will just write **Type** to denote some type universe (or its category of types and functions). This is called *typical ambiguity*, meaning that every theorem holds for all (suitable) assignments of universes to these instances of **Type**.

One notable class of types is given by the *function types*. For types A and B, our type system might have the type $A \to B$. An element f of this type can be combined with an element a of A to give an element $f(b)$ of B. Of course, the elements of this type are thought of (and usually are) functions from A to B.

Now, a type system may or may not have all kinds of constructs. One of these constructs

is *dependent types*. A dependent type is a type which depends on values of other types. For example, $\texttt{array}(T, n)$, the type of arrays of length n, with elements of type T. The study of type systems that have such dependent types is called *dependent type theory*.

Suppose that we have a type A, and a dependent type which we will write $B : A \to \mathbf{Type}$.

One of the possible constructs in a type system is a type $\sum_{a:A} B(a)$ called the *dependent sum*, consisting of all pairs (a, b) with $a : A$ and $b : B(a)$. So every element of $\sum_{a:A} B(a)$ gives an element of one $B(a)$ (for some $a : A$). Note that for the constant dependent type $B(a) = B$, $\sum_{a:A} B$ is the product type $A \times B$.

Another construct which may exist, is a type $\prod_{a:A} B(a)$, consisting of all 'functions' f which map elements $a : A$ to elements $f(a) : B(a)$. Every element of $\prod_{a:A} B(a)$ gives therefore elements of all the $B(a)$ simultaneously. Note that $\prod_{a:A} B$ is the function type $A \to B$.

Lastly, there is a very strong connection between logic and type theory. This is called the *Curry-Howard correspondence* or sometimes referred to as *products as types*. We can view a type T as the proposition "T has an element". An element of T is then a 'proof' or 'witness' of this proposition. If the type system is strong enough, it allows us to do mathematics in it, where:

- A function $f : A \to B$ that takes an argument of type A and yields a result of type B corresponds to the proof of B under the hypothesis that A holds: "suppose that A, then B". Note that the notation $A \to B$ also makes sense if we think of A and B propositions.

- The empty type $\mathbf{0}$ corresponds to 'false'. Note that for all types A, we can construct a function $\mathbf{0} \to A$. In other words, we can prove everything from 'false'.

- The unit type $\mathbf{1}$ corresponds to 'true'.

- The negation of T is the function type $T \to \mathbf{0}$.

- The conjuction "A and B" is given as the product type $A \times B$.

- The disjunction "A or B" is given as the propositional truncation of the coproduct, union or sum type $A \sqcup B$.

- The statement "For all $a : A$, $B(a)$ holds", for some dependent type B (i.e. predicate) over A, is given as the dependent product $\prod_{a:A} B(a)$.

If we think of types as propositions, the question whether some axiom is assumed or not becomes the question whether some (family of) type(s) is inhabited. Note that most of the axioms that we list here require the notion of mere propositions, propositional truncation and mere existence, which we will cover from Section 2.4 onwards.

- If for all mere propositions A, the type $\|A \vee (A \to \mathbf{0})\|$ is inhabited, we say that the type system assumes the *axiom of excluded middle*: "Either A is true, or A is not true". This is axiom allows us to prove A from 'not not A'.

- If for all dependent types C over A and B, the type

$$\left(\prod_{a:A} \exists(b : B), C(a, b)\right) \to \exists(f : A \to B), \prod_{a:A} C(a, f(a))$$

  is inhabited, we say that the type system assumes the *propositional axiom of choice*: "The product of a family of nonempty sets is nonempty".

- In any type theory which has the required constructs, the following holds: For all dependent types C over A and B, the type

$$\left(\prod_{a:A}\sum_{b:B} C(a,b)\right) \to \sum_{f:A\to B}\prod_{a:A} C(a,f(a))$$

is inhabited. This axiom (or actually more of a theorem) is called the *type theoretic axiom of choice*.

2.2 The univalence principle

"*Isomorphic objects are equivalent.*"

This principle is visible in most of mathematics: Sets with a bijection have the same number of elements, isomorphic groups have the same properties, and since the universal property of limits makes them "unique up to unique isomorphism", we can talk about 'the' limit of some diagram in a category.

Now, the *univalence principle* takes this a step further. It states that

"*Isomorphic objects are equal.*"

Univalent foundations seeks to be a foundation for mathematics that is in line with this principle. This is often done within the framework of 'Martin-Löf dependent type theory', a type theory developed by Per Martin-Löf [Mar71]. It is a family of dependent type systems with dependent products, dependent sums, an empty type, a unit type and union types. It is a constructive type theory, so it does not automatically assume the axiom of excluded middle or the propositional axiom of choice. It is however compatible with these axioms, so one can still assume these alongside its usual axioms.

It is important to note that Martin-Löf type theory has *identity types*: given a type T and elements $x, y : T$, we have a type $\mathrm{Id}_T(x,y)$ (note that this is a dependent type), which we will usually denote with $x = y$. An element $p : x = y$ is a proof that x is 'equal' to y. This type comes with an interesting induction principle called *path induction*: we can show any statement about paths $p : x = y$ for generic x and y, if we can show it about the 'trivial' path $\mathrm{refl} : x = x$ for generic x. For example, symmetry of the equality boils down to a function $\prod_{xy:T} x = y \to y = x$. We construct this using path induction with the function that sends $\mathrm{refl} : x = x$ to itself. For more information, see [Uni13], Section 1.12.1.

In set-theoretic mathematics, there is the concept of a 'bijection' $S \simeq T$ of sets (or an isomorphism in the category **Set**), which is often treated as an equivalence. It consists of functions $f : S \to T$ and $g : T \to S$ with $f \cdot g = \mathrm{id}_S$ and $g \cdot f = \mathrm{id}_T$. In type theory, we have a similar concept, which is called 'equivalence' (of types) $S \simeq T$. Since bijections are not well-behaved for types that are not sets (see 2.4), because in those cases, a function $f : S \to T$ can have multiple distinct inverses. Therefore, we define $S \simeq T := \sum_{f:S\to T} \mathrm{isequiv}(f)$, for some predicate $\mathrm{isequiv} : (S \to T) \to \mathbf{Type}$ (see [Uni13], Equation 2.4.10). However, intuitively we can still think of these as bijections.

Using the identity type, we can make our statement of the univalence principle more precise (and a bit stronger). For types, we can construct a function

$$\mathrm{idtoequiv} : \prod_{S,T:\mathbf{Type}} (S = T) \to (S \simeq T).$$

We construct this function using path induction with the identity bijection $\prod_S \mathrm{id}_S : (S \simeq S)$. In fact, this is a specific case of the following: for a category C, if we denote the type of isomorphisms between objects c and d with $c \cong d$, we can construct a function

$$\mathrm{idtoiso} : \prod_{c,d:C} (c = d) \to (c \cong d),$$

using path induction with the identity isomorphism $\prod_{c:C} \mathrm{id}_c : (c \cong c)$. We can then formulate the univalence principle for categories as

"For all $c, d : C$, $\mathrm{idtoiso}_{c,d} : (c = d) \to (c \cong d)$ is an equivalence."

A category that adheres to the univalence principle is called a *univalent category*.

Note that if B is a univalent category and A is any cateory, the functor category $[A, B] = B^A$ is univalent as well. In particular, the presheaf category $PA = [A^{\mathrm{op}}, \mathbf{Set}]$ is univalent.

Lemma 30. *For every category C, there exists a univalent category $\hat{C}$ with a weak equivalence*

$$\iota : C \hookrightarrow \hat{C}.$$

Proof. See [AKS15], Theorem 8.5. □

We call this univalent category the *Rezk completion*. **(TODO)** : Uniqueness?

2.3 The univalence axiom

Now, even for a basic category, like the category of types, it seems impossible to prove that the univalence principle holds. However, this is no surprise: it turns out that it is independent of the axioms of Martin-Löf type theory (**(TODO) Is this true? I need a reference for this, but could not find one**). That is: it cannot be proven, but assuming it as an axiom does not yield contradictions.

As mentioned before, univalent foundations attempts to develop as much of mathematics as possible along the univalence principle. Therefore, we assume as our first axiom the *univalence axiom*:

Axiom. *For all S, T : **Type**, the function* $\mathrm{idtoequiv}_{S,T} : (S = T) \to (S \simeq T)$ *is an equivalence.*

In other words:

Axiom. Type *is univalent.*

Remark 16. One consequence of the independence of the univalence axiom is that equivalent objects are 'indiscernible'. That is: even if we do not yet assume the univalence axiom, we cannot formulate a property that is satisfied by some type, but not by another, equivalent type. This is because such a property would yield a contradiction when we would assume the univalence axiom.

Now, the question arises: how about the univalence axiom for categories other than **Type**? Do we need to keep assuming an additional axiom for every category that we want to be univalent? It turns out that this is not necessary. In practice, most categories consist of 'sets (or types) with additional structure'. For example: topological spaces, groups, λ-theories and algebraic theory algebras. In such categories, we can leverage the univalence of **Type** to show that for isomorphic objects, their underlying types are equal. Also morphisms are usually defined in such a way that they 'preserve' the 'additional structure', which is what we need to show that the category is univalent.

Also, Theorem 4.5 in [AKS15] shows that if a category B is univalent (in the paper, categories are called 'precategories' and univalent categories are just called 'categories'), then the functor category $A \to B$ is also univalent. In particular, the category of (pre)sheaves $A \to \mathbf{Set}$ is univalent.

Therefore, the univalence axiom is a very powerful axiom, and we usually do not need to assume additional axioms to show that more categories satisfy the univalence principle.

The last structure in this section for which we want to consider the univalence axiom, is the 2-category **Cat** of categories. In general, we cannot show that this satisfies the univalence

principle. However, we will restrict our attention to the sub-2-category of univalent categories, which are the categories that we want to study. Then Theorem 6.8 in [AKS15] shows that for univalent categories C and D, there is an equivalence between $C = D$ and $C \simeq D$, where $C \simeq D$ denotes the type of (adjoint) equivalences of categories (see Definition 2).

Lastly, a result about univalent categories that we will use a couple of times in this thesis:

Lemma 31. *For a functor between univalent categories $F : A \to B$, the types 'F is an adjoint equivalence' and 'F is a weak equivalence' are equivalent propositions (see 2.4).*

Proof. See [AKS15], Lemma 6.8. □

2.4 hProps and hSets

If we have a type T and objects x and y, we can wonder how many elements $x = y$ has. In set-based mathematics, this would be a nonsensical question: two elements of a set are either equal or not equal. Therefore, we can expect the answer to be that $x = y$ has at most one element. And indeed, if we do not assume the univalence axiom, we can assume another axiom, called 'uniqueness of identity proofs', which states that for $p, q : x = y$, we have a proof of equality $h : p = q$.

On the other hand, suppose that we do assume the univalence axiom. Consider the two-element type $T = \{-1, 1\}$. We can construct two different equivalences $\mathrm{id}_T, \sigma : T \simeq T$:

$$\mathrm{id}_T(x) = x \quad \text{and} \quad \sigma(x) = -x.$$

By the univalence axiom, we must have that the identity type $(T = T)$ has (at least) two distinct elements, corresponding to id_T and σ. Therefore, the univalence axiom is not compatible with uniqueness of identity proofs, and we see that in a univalent setting, some identity types have more than one element.

A consequence of this is that types in general have too little structure to serve as a foundation for mathematics that was originally set-based. For example, suppose that we want to formalize the theory of groups. A group homomorphism $f : \mathbf{Grp}(H, G)$ is defined as a function on the underlying types $f_1 : H \to G$, together with a proof that it commutes with the group operations: $f_2 : \prod_{x,y:H} f_1(x \circ y) = f_1(x) \circ f_1(y)$. Now, normally in group theory, to show that two homomorphisms $f, g : \mathbf{Grp}(H, G)$ are equal, we show that $\prod_{x:H} f_1(x) = g_1(x)$, the 'data' is equal. However, if we are working with types instead of sets, we also need to show that the proofs of the 'properties' are equal: $f_2 = g_2$ (note that we actually would need to transport f_2 here, for this equality to typecheck). This makes showing equality of morphisms much more complicated for concrete groups, and sometimes outright impossible for generic groups.

To deal with this, we need the concepts of mere propositions and (homotopy) sets:

Definition 25. A *mere proposition* is a type T such that for all $x, y : T$, $x = y$.

Definition 26. A *homotopy set* is a type T such that for all $x, y : T$, $x = y$ is a mere proposition.

Since the identity types for a homotopy set are mere propositions, a homotopy set mimicks a set in set theory, where equality between elements 'is' or 'is not'. If we restrict the underlying type of a group to be a homotopy set, it can be shown that $\prod_{x,y:H} f_1(x \circ y) = f_1(x) \circ f_1(y)$ is a mere proposition, so $f_2 = g_2$ trivially. This is often true when translating definitions from set theory to univalent foundations: if we base our objects on homotopy sets instead of types, we only have to worry about equality of 'data', the equality of 'properties' follows automatically.

For similar reasons, we restrict the hom-types $C(c, d)$ of categories to be sets. Note that **Type** is not a category under this definition (it is a 'precategory', for some definition of precategory), because in general, the type of functions between sets $S \to T$ is not a set.

2.5 Truncation and (mere) existence

As mentioned before, if we want to do mathematics in a dependent type theory, we can 'encode' propositions as types. The elements of the type correspond to the proofs that the proposition is true, see for example the identity types. However, we need to be careful here about the distinction between types in general and mere propositions:

A proof that a type T is nonempty usually consists of giving an element $t : T$. If we encode the statement "T is nonempty" as T, and if T is not a mere proposition, then "T is nonempty" is not a mere proposition, so it has multiple distinct proofs. In some cases, this is exactly what we want, because we want to retrieve the chosen element of T. However, in some cases, we want the fact that a set is nonempty (or any statement in general) to be a mere proposition, to avoid having to carry around a specific element and having to prove equality of two specific elements. For such cases, we have the 'propositional truncation':

Definition 27. For a type T, a type $\|T\|$ exists ([Uni13], Section 3.7) with the properties that for all $t : T$, we have an element $|t| : \|T\|$ and that $\|T\|$ is a mere proposition. It has a recursion principle stating that for a mere proposition B, a function $f : A \to B$ induces a function $|f| : \|T\| \to B$ that commutes with $|\cdot|$. This object is called the *propositional truncation*.

The propositional truncation forgets the details about a type, and only keeps the information about whether it is inhabited or not. The recursion principle means that if we are trying to prove a mere proposition based on some element $|t| : \|T\|$, we can pretend that we do have a concrete element $t : T$.

There also is the concept of higher order truncations. For example, the *set truncation* $\|A\|_0$, which is a homotopy set and has an equivalence $(\|A\|_0 \to B) \simeq (A \to B)$ for any set B. However, these higher truncation types become increasingly harder to construct, and in this thesis, we will only need to consider the propositional truncation.

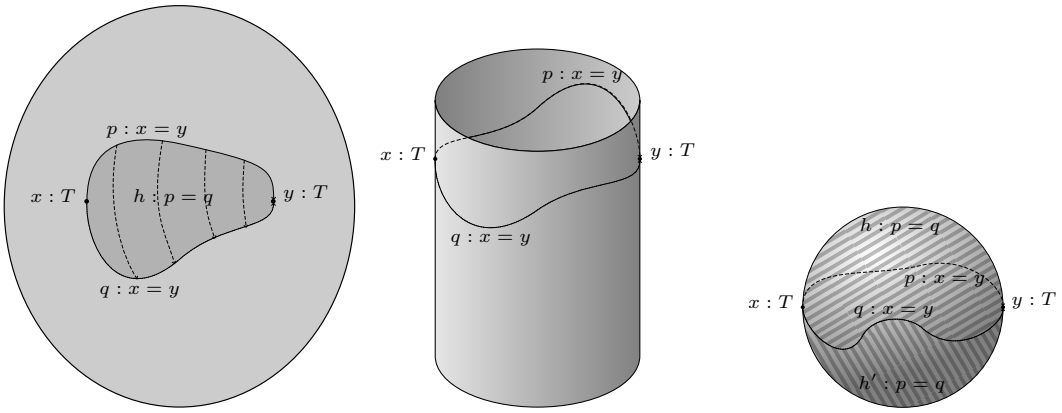
Often, when we prove a theorem or lemma that "there exists some $x : X$ such that $Y(x)$", what we actually mean is that we can construct an element $x : X$ and then an element $y : Y(x)$ of the dependent type Y over X. This is equivalent to having an element of $\sum_{x:X} Y(x)$. However, this is in general of course not a mere proposition. If we want to express that the set of such x is nonempty as a mere proposition, we talk about *mere existence*:

Definition 28. Given a dependent type Y over X, if we say that there *merely exists* an element $x : X$ such that $Y(x)$, we mean that we have an element of the propositional truncation

$$h : (\exists x : Y(x)) := \left\| \sum_{x:X} Y(x) \right\|.$$

For example, if we have objects in a category $c, d : C$ and we want to talk about a retraction f of c onto d, we commonly define this as "a morphism $f_1 : C(c, d)$, such that there exists a 'section': a morphism $f_2 : C(d, c)$ with $f_2 \cdot f_1 = \mathrm{id}_d$". Now, we commonly consider f_1 to be the 'data' of the retraction; we consider retractions f and f' to be the same retraction if $f_1 = f'_1$. This means that being a retraction is about the 'mere existence' of a section. Note, however, that in this case, we cannot use the section in constructions, except when we are trying to prove mere propositions.

Note that for the Curry-Howard correspondence, the product or conjunction $A \wedge B = A \times B$ of two mere propositions is again a mere proposition. However, the union $A \sqcup B$ is not necessarily a mere proposition. To make it into a mere proposition, we need to take the propositional truncation $A \vee B = \|A \sqcup B\|$.



(a) (b) (c)

Figure 2.1: Different possible homotopy structures

2.6 Equality and homotopy

Univalent Foundations is often mentioned together with Homotopy Type Theory, because they are related but distinct concepts. Therefore, we will mention it here.

As we saw before, if we do not assume uniqueness of identity proofs, given two elements $x, y : T$, we can have multiple distinct proofs that $x = y$, which is counterintuitive. One way to think about this, is the perspective of homotopy type theory. In homotopy type theory, one considers a type T to be a 'space', intuitively similar to a topological space, but without an explicit topology. The elements $t : T$ are then the points of the space. Elements of the identity type $(s = t)$ are interpreted as 'paths' from s to t. That is why the induction principle of $(s = t)$ is called 'path induction'. Of course, we can go higher: for $p, q : x = y$, the elements $(p = q)$ are 'paths between paths', (path) 'homotopies', 'sheets' or '1-cells', for homotopies $h, h' : p = q$, the elements of $h = h'$ are paths between paths between paths, 'volumes' or '2-cells' etc.

If we have a 'geometric' interpretation of our type theory, we can investigate the 'shape' of a (nonempty) type T, given by the structure of the (higher) identity types.

First of all, if we have $x, y : T$ for which $x = y$ does not have an inhabitant, x and y lie in different 'connected components'. Now, we focus on a connected type:

For example, are all elements $x, y : T$ of the type equal to each other, and are all elements $p, q : x = y$ of all the (higher) identity types also equal in a unique way? Then we have a *contractible type*, which intuitively looks somewhat like a plane (Figure 2.1a).

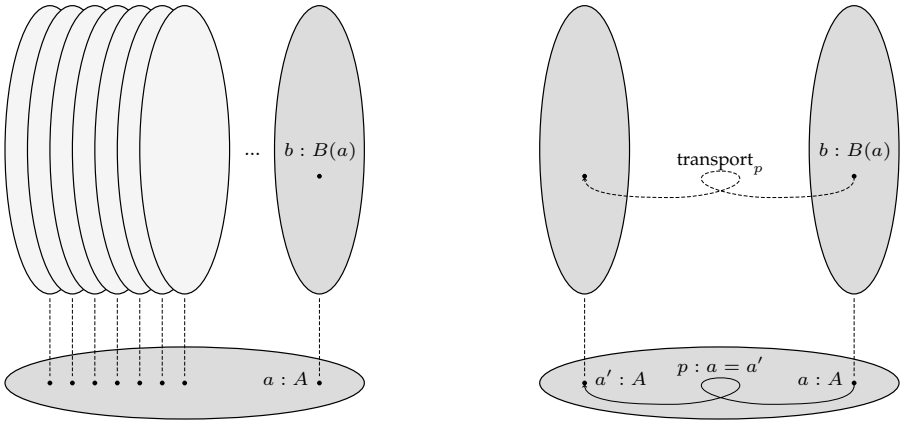
It is also possible that any two elements $x, y : T$ are equal, but that there are distinct paths $p, q : x = y$, with no homotopy between them. Then intuitively T looks like a circle or a tube or a projective space, or something more complicated (Figure 2.1b).

Remark 17. Note that we can give the type of paths $x = x$ a group structure and $x = y$ is a 'torsor' for this group. If $x = y$ has exactly two distinct elements, the group $x = x$ looks like $\mathbb{Z}/2\mathbb{Z}$, which suggests some projective plane-like structure.

For something to look like a circle or tube, we need this group $x = x$ to be isomorphic to $\mathbb{Z}$. For an example, see [Bez+20].

As a third example, consider a type T in which any two elements $x, y : T$ are equal, and any two paths $p, q : x = y$ have two distinct homotopies $h, h' : p = q$ between them. Then we can imagine the type to look somewhat like a sphere (Figure 2.1c) or something more complicated.

This is the lens through which homotopy type theory studies types.



(a) (b)

Figure 2.2: A fibration with a path in the base space, giving rise to transport in the fibration

2.7 Transport

Suppose that we have a dependent type $B : A \to UU$. Intuitively, B is a collection of spaces, lying over the points of A (Figure 2.2a). Now, if we have a path $p : a = a'$ in A, we can use path induction to get a function $\mathrm{transport}_p : B(a) \to B(a')$ (Figure 2.2b).

For example, this allows us to transfer an element from $\mathtt{array}(T, n + n)$ to $\mathtt{array}(T, 2 \cdot n)$ with transport over the path $n + n = 2 \cdot n$. Also, if we assume the univalence axiom, and we have, for example, an isomorphism of groups $f : G \cong G'$ and a proof h that G is semisimple, we can transport h along the equality given by f to a proof that G' is semisimple.

Example 2. In addition, consider the following example. We take A to be a type universe, and let $B : T \mapsto \mathtt{array}(T, 3)$. Consider the types $a = \{\top, \bot\}$ and $a' = \{0, 1\}$. We have equivalences $\overline{p}$ and $\overline{q}$ between a and a':

$$\overline{p}(\top) = 1, \overline{p}(\bot) = 0 \quad \text{and} \quad \overline{q}(\top) = 0, \overline{q}(\bot) = 1.$$

That means that we have distinct equalities $p, q : a = a'$. Now, suppose that we have an array $x = [\top, \top, \bot] : B(a)$. Since $a = a'$, we would want to treat x as an element of $B(a')$, but then we need to make a choice whether we treat it as $[1, 1, 0]$ or $[0, 0, 1]$ (or something else altogether). This is exactly the question whether we transport along p or q, and therefore, when our base type is not a set, it is important to be aware that we are transporting a property along an equality, and we need to be careful which equality it is that we transport along.

Another place where transports occur frequently is when proving equality $(x, y) = (x', y')$ of elements of a dependent sum $\sum_{a:A} B(a)$. We can start componentwise by proving $p : x = x'$, but after this, we cannot directly prove $q : y = y'$, because these two live in different types: $B(x)$ and $B(x')$ respectively. Therefore, we need to transport, and then prove

$$\mathrm{transport}_p(y) = y' : B(x').$$

2.7.1 Caution: 'transport hell'

Now, it seems that we can transport all properties and data along equalities. And of course, that is true, but some caution needs to be taken with this when working with a proof assistant.

For example, consider the situation in Example 2. As mentioned, we can transport x to get an element $y := \mathrm{transport}_p(x)$, which is an element of $B(a')$, but now suppose that we want to compute something using its first coefficient y_1. Of course, on paper it quickly becomes clear that $y_1 = 1$, but in practice, it takes quite some work to convince a proof assistant of that

fact. Of course, we could write a lemma which states that for any array x, any equivalence $\overline{p}$ with corresponding equality p, and any index i,

$$\mathsf{transport}_p(x)_i = \overline{p}(x_i).$$

However, at that point, we have more or less constructed our own function between $B(a)$ and $B(a')$, which is much easier to work with than $\mathsf{transport}_p$.

Experience teaches that in general, it is fine to transport properties $b : B(a)$ of which we will never need the value, only the fact that it (merely) exists. On the other hand, for properties and constructions of which we might later want to use the actual value, it is much better to transfer them 'by hand'. Often, but not always, this criterion coincides with $B(a)$ being a mere proposition.

Another case where unwanted transports often occur is when showing the equality of two complicated elements of a type. For example, $((x_1, x_2), (x_3, x_4)) = ((x_1', x_2'), (x_3', x_4'))$, both elements of

$$\sum_{(x_1,x_2):\sum_{a:A} B(a)} \sum_{x_3:C(x_1)} D(x_1, x_2, x_3).$$

This example involves proofs:

- $p : x_1 = x_1'$;
- $q : \mathsf{transport}_p(x_2) = x_2'$;
- $r : \mathsf{transport}_{(p,q)}(x_3) = x_3'$, which can be simplified to $\mathsf{transportf}_p(x_3) = x_3'$, since x_3 does not depend on x_2
- and finally $s : \mathsf{transport}_{((p,q),r)}(x_4) = x_4'$.

In general, these latter proofs r and s are very challenging and best avoided whenever possible. The situation in which one has to work with such complicated transports, which make a proof (seemingly unnecesarily) complicated is called *transport hell*. Some mathematical tools have been developed (see for example 6.3 about displayed categories) that help avoid transport hell in some cases. Additionally, the structure or 'strategy' of a proof can sometimes be changed in order to avoid the worst of the transports. However, it is often not possible to avoid transports altogether, and this is one of the reasons why proving something in a proof assistant takes a lot more work than proving it on paper.

Chapter 3

Algebraic Structures

3.1 Algebraic Theories

Definition 29. We define an *algebraic theory* T to be a sequence of sets T_n indexed over $\mathbb{N}$ with for all $1 \leqslant i \leqslant n$ elements ("variables" or "projections") $x_{n,i} : T_n$ (we usually leave n implicit), together with a substitution operation

$$\_ \bullet \_ : T_m \times T_n^m \to T_n$$

for all m, n, such that

$$\begin{aligned} x_j \bullet g &= g_j \\ f \bullet (x_{l,i})_i &= f \\ (f \bullet g) \bullet h &= f \bullet (g_i \bullet h)_i \end{aligned}$$

for all $1 \leqslant j \leqslant l$, $f : T_l$, $g : T_m^l$ and $h : T_n^m$.

Remark 18. (**TODO**) Say something about universal algebra For equivalent definitions of different kinds, see Chapter A.

Definition 30. A *morphism* f between algebraic theories T and T' is a sequence of functions $f_n : T_n \to T'_n$ (we usually leave the n implicit) such that

$$\begin{aligned} f_n(x_j) &= x_j \\ f_n(f \bullet g) &= f_m(f) \bullet (f_n(g_i))_i \end{aligned}$$

for all $1 \leqslant j \leqslant n$, $f : T_m$ and $g : T_n^m$.

Together, these form the category of algebraic theories **AlgTh**.

Remark 19. We can construct binary products of algebraic theories, with sets $(T \times T')_n = T_n \times T'_n$, variables (x_i, x_i) and substitution

$$(f, f') \bullet (g, g') = (f \bullet g, f' \bullet g').$$

In the same way, the category of algebraic theories has all limits.

Later on, we will see an example of a trivial algebraic theory (the terminal theory) T, in which every T_n only contains one element. However, we can say a bit about nontrivial algebraic theories.

Lemma 32. *Let T be an algebraic theory, such that T_n has at least two distinct elements for some n. Then for all $1 \leqslant i, j \leqslant m$ with $i \neq j$, we have $x_{m,i} \neq x_{m,j}$.*

Proof. We can also formulate the statement as: If there exist $1 \leqslant i, j \leqslant m$ with $i \neq j$ such that $x_{m,i} = x_{m,j}$, then all L_n contain at most one distinct element.

So, suppose that $x_{m,i} = x_{m,j}$ for some $i \neq j$. For $a, b : L_n$, we define $v : L_n^m$ as

$$v_k = \begin{cases} a & k = i \\ b & k \neq i \end{cases},$$

so in particular, $v_j = b$. Then we have

$$a = v_i = x_{m,i} \bullet v = x_{m,j} \bullet v = v_j = b,$$

so L_n contains at most one distinct element. □

3.2 Algebras

Definition 31. An *algebra* A for an algebraic theory T is a set A, together with an action

$$\bullet : T_n \times A^n \to A$$

for all n, such that

$$x_j \bullet a = a_j$$
$$(f \bullet g) \bullet a = f \bullet (g_i \bullet a)_i$$

for all j, $f : T_m$, $g : T_n^m$ and $a : A^n$.

Definition 32. For an algebraic theory T, a *morphism* f between T-algebras A and A' is a function $f : A \to A$ such that

$$f(t \bullet a) = f \bullet (f(a_i))_i$$

for all $t : T_n$ and $a : A^n$.

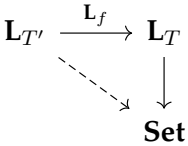
Together, these form the category of T-algebras $\mathbf{Alg}_T$.

Remark 20. The category of algebras has all limits. The set of a limit of algebras is the limit of the underlying sets.

Remark 21. Note that for an algebraic theory T, the T_n are algebras for T, with the action given by $\bullet$.

Definition 33. If we have a morphism of algebraic theories $f : T' \to T$, we have a *pullback functor of algebras* $f^* : \mathbf{Alg}_T \to \mathbf{Alg}_{T'}$. It endows T-algebras with an action from T' given by $g \bullet_{T'} a = f(g) \bullet_T a$. Then T-algebra morphisms commute with this T'-action, so we indeed have a functor.

Remark 22. Note that by Lemma 71, algebras for T are equivalent to finite-product-preserving functors from its Lawvere theory to **Set**. Then $f : T' \to T$ corresponds to a functor on the Lawvere theories $\mathbf{L}_f : \mathbf{L}_{T'} \to \mathbf{L}_T$, and $f^* : \mathbf{Alg}_T \to \mathbf{Alg}_{T'}$ corresponds to precomposition with $\mathbf{L}_f$:



Actually, we can recover some information about the algebraic theory morphism from this pullback functor. For example,

Lemma 33. *If f^* is an equivalence of categories, f is an isomorphism.*

Proof. (**TODO**) □

We can also consider the category of 'all' algebraic theory algebras together. That is, the category C with $C_0 = \sum_{T:\mathbf{AlgTh}}, \mathbf{Alg}_T$ and $C((T, A), (T', A'))$ consisting of pairs $(f, f') : \mathbf{AlgTh}(T, T') \times \mathbf{Set}(A, A')$ such that for all $t : T_n$ and $a : A^n$,

$$f'(t \bullet a) = f(t) \bullet (f'(a_i))_i.$$

We then have a functor $P : C \to \mathbf{AlgTh}$, projecting onto the first coordinate.

Lemma 34. *P is a fibration.*

Proof. Given an algebraic theory morphism $f : \mathbf{AlgTh}(S, T)$ and a T-algebra A_T, Definition 33 gives an S-algebra A_S with underlying set A_T. The cartesian morphism is $(f, \mathrm{id}_{A_T}) : C((S, A_S), (T, A_T))$.

It is cartesian because for $(R, A_R) : C$ and $(g, g') : C((R, A_R), (T, A_T))$ and $h : \mathbf{AlgTh}(R, S)$ with $h \cdot f = g$, the required morphism $C((R, A_R), (S, A_S))$ is given by $g' : \mathbf{Set}(A_R, A_S)$. □

3.3 Presheaves

Definition 34. A *presheaf* P for an algebraic theory T is a sequence of sets P_n indexed over $\mathbb{N}$, together with an action

$$\bullet : P_m \times T_n^m \to P_n$$

for all m, n, such that

$$t \bullet (x_{l,i})_i = t$$
$$(t \bullet f) \bullet g = t \bullet (f_i \bullet g)_i$$

for all $t : P_l$, $f : T_m^l$ and $g : T_n^m$.

Definition 35. For an algebraic theory T, a *morphism* f between T-presheaves P and P' is a sequence of functions $f_n : P_n \to P'_n$ such that

$$f_n(t \bullet f) = f_m(t) \bullet f$$

for all $t : P_m$ and $f : T_n^m$.

Together, these form the category of T-presheaves $\mathbf{Pshf}_T$.

Remark 23. The category of presheaves has all limits. The nth set $\overline{P}_n$ of a limit $\overline{P}$ of presheaves P_i is the limit of the nth sets $P_{i,n}$ of the presheaves in the limit diagram.

An analogue to Lemma 34 shows that, like with algebras, the total category of presheaves is fibered over the category of algebraic theories.

3.4 λ-theories

Let $\iota_{m,n} : T_m \to T_{m+n}$ be the function that sends f to $f \bullet (x_{m+n,1}, \ldots, x_{m+n,m})$. Note that

$$\iota_{m,n}(f) \bullet g = f \bullet (g_i)_{i \leqslant m} \quad \text{and} \quad \iota_{m,n}(f \bullet g) = f \bullet (\iota_{m,n}(g_i))_i.$$

For tuples $x : X^m$ and $y : X^n$, let $x + y$ denote the tuple $(x_1, \ldots, x_m, y_1, \ldots, y_n) : X^{m+n}$.

Definition 36 (λ-theory)**.** A *λ-theory* is an algebraic theory L, together with sequences of functions $\lambda_n : L_{n+1} \to L_n$ and $\rho_n : L_n \to L_{n+1}$, such that

$$\lambda_m(f) \bullet h = \lambda_n(f \bullet ((\iota_{n,1}(h_i))_i + (x_{n+1})))$$
$$\rho_n(g \bullet h) = \rho_m(g) \bullet ((\iota_{n,1}(h_i))_i + (x_{n+1}))$$

for all $f : L_{m+1}$, $g : L_m$ and $h : L_n^m$.

Together, these form the category of λ-theories **LamTh**.

Definition 37 (β- and η-equality)**.** We say that a λ-theory L satisfies β-equality (or that it is a λ-theory with β) if $\rho_n \circ \lambda_n = \mathrm{id}_{L_n}$ for all n. We say that is satisfies η-equality if $\lambda_n \circ \rho_n = \mathrm{id}_{L_{n+1}}$ for all n.

Definition 38 (λ-theory morphism)**.** A *morphism* f between λ-theories L and L' is an algebraic theory morphism f such that

$$f_n(\lambda_n(s)) = \lambda_n(f_{n+1}(s))$$
$$\rho_n(f_n(t)) = f_{n+1}(\rho_n(t))$$

for all $s : L_{n+1}$ and $t : L_n$.

Remark 24. The category of lambda theories has all limits, with the underlying algebraic theory of a limit being the limit of the underlying algebraic theories.

Definition 39. A λ-theory algebra or presheaf is an algebra or presheaf for the underlying algebraic theory.

3.4.1 The λ-calculus operations

Now, for a λ-theory L, we have variables $x_{n,i} : L_n$ and λ-abstraction $f \mapsto \lambda(f)$. We will sometimes denote $\lambda(f)$ as $\lambda x_{n+1}, f$ for $f : L_{n+1}$. Now, consider the element $\rho(x_{1,1}) : L_2$.

Definition 40. Using the substitution, we have binary operations on the L_n, sending $(f, g) : L_n \times L_n$ to $\rho(x_{1,1}) \bullet (f, g) : L_n$.

We will denote $\rho(x_{1,1}) \bullet (f, g)$ as fg, and this gives us our application operation, so we can interpret all three operations of the λ-calculus in L.

Remark 25. Note that if L has β-equality, we have

$$\lambda(f)g = \rho(\lambda(f)) \bullet (x_1, \ldots, x_n, g) = f \bullet (x_1, \ldots, x_n, g),$$

so we have the usual β equality. In the same way η-equality in L gives

$$\lambda(\iota_{n,1}(f)x_{n+1}) = \lambda(\rho(f)) = f.$$

Remark 26. Also that for $f : L_n$,

$$\rho(f) = \rho(x_1 \bullet f) = \iota_{n,1}(f)x_{n+1}.$$

Therefore, we can also think of the application as being the primary operation, from which we derive ρ. In the same way, we have

$$\rho^m(f) = \iota_{n,m}(f)x_{n+1} \ldots x_{n+m}.$$

Remark 27. Note that for $f, g : L_n$,

$$\rho(f) \bullet (x_1, \ldots, x_n, g) = \rho(x_1) \bullet (\iota_{n,1}(f), x_{n+1}) \bullet (x_1, \ldots, x_n, g) = \rho(x_1) \bullet (\iota_{n,1}(f), g),$$

so we could also define the application as $fg = \rho(f) \bullet (x_1, \ldots, x_n, g)$, although that is more complicated.

3.5 Examples

There is a lot of different examples of algebraic theories and their algebras. Some of these even turn out to be λ-theories. In this section, we will discuss a couple of these.

3.5.1 The free algebraic theory on a set

Example 3. Let S be a set. We can construct an algebraic theory $F(S)$ by taking $F(S)_n = S \sqcup \{1, \dots, n\}$ with projections $x_i = i$ and substitution

$$i \bullet g = g_i \qquad\qquad s \bullet g = s$$

for $i : \{1, \dots, n\}$ and $s : S$.

If we have a function $f : S \to S'$, we get a morphism $F(f) : F(S) \to F(S')$ given by

$$F(f)_n(i) = i \qquad\qquad F(f)_n(s) = f(s)$$

for $i : \{1, \dots, n\}$ and $s : S$.

Also, F obviously respects the identity and substitution morphisms, so it is a functor.

Note that we have a forgetful functor $(\cdot)_0$ that sends a morphism of algebraic theories $g : T \to T'$ to the function $f_0 : T_0 \to T'_0$.

Lemma 35. *The algebraic theory $F(S)$ defined above, is the free algebraic theory on the set S.*

Proof. Let T be an algebraic theory. We have an equivalence

$$\mathbf{AlgTh}(F(S), T) \cong \mathbf{Set}(S, T_0),$$

sending $f : \mathbf{AlgTh}(F(S), T)$ to $f_0 : S = S \sqcup \varnothing \to T_0$ (this is trivially natural in S and T) and $f : \mathbf{Set}(S, T_0)$ to the functions $g_n : F(S)_n \to T_n$ given by

$$g_n(i) = x_i \qquad\qquad g_n(s) = f(s) \bullet ().$$

□

The proofs that $F(S)$ is an algebraic theory and that $F(f)$ and g are algebraic theory morphisms is an easy exercise in case distinction.

Corollary 4. *$F(\varnothing)$ is the initial algebraic theory.*

Proof. For $S = \varnothing$, the equivalence of hom-sets becomes

$$\mathbf{AlgTh}(F(\varnothing), T) \cong \mathbf{Set}(\varnothing, T_0)$$

and the latter has exactly one element. □

Lemma 36. *There is an adjoint equivalence between the category $\mathbf{Alg}_{F(S)}$ and the coslice category $S \downarrow \mathbf{Set}$.*

Proof. For the equivalence, we send a $F(S)$-algebra A to the set A with morphism $s \mapsto s \bullet ()$. An algebra morphism $f : A \to B$ is sent to the coslice morphism $f : (S \to A) \to (S \to B)$. This constitutes a functor.

Note that the category of $F(S)$-algebras is univalent.

Also, the functor is fully faithful, since one can show that for $F(S)$-algebras, the coslice morphism $\varphi : (f : S \to A) \to (f' : S \to B)$ also has the structure of an algebra morphism $\varphi : A \to B$.

Lastly, the functor is essentially surjective, since we can lift an object $f : S \to X$ to a $F(S)$-algebra X, with action

$$i \bullet x = x_i \quad \text{and} \quad s \bullet x = f(s).$$

Therefore, the functor $\mathbf{Alg}_{F(S)} \to S \downarrow \mathbf{Set}$ is an adjoint equivalence.

The proofs of these facts work by simple case distinction, and by using the properties of the coslice and algebra morphisms. □

Remark 28. $F(\varnothing)$ is, in some sense, the smallest nontrivial algebraic theory. Then $F(S)$ is the smallest nontrivial algebraic theory that has the elements of S as constants.

Remark 29. Note that the category of $F(\varnothing)$-algebras is equivalent to the coslice-category $(\varnothing \downarrow \mathbf{Set})$, which, since $\varnothing$ is the initial set, is just equivalent to **Set**.

3.5.2 The free λ-theory on a set

In this subsection, we will use the λ-calculus operations defined in Subsection 3.4.1.

Like with the free algebraic theory, we will construct the free λ-theory as the smallest nontrivial λ-theory (which is the λ-calculus) with some additional constants.

Let S be a set. Consider the sequence of inductive types $(\Lambda(S)_n)_n$ with the following constructors:

$$\begin{aligned}
&\mathtt{Var}_n : \{1, \dots, n\} \to \Lambda(S)_n;\\
&\mathtt{App}_n : \Lambda(S)_n \to \Lambda(S)_n \to \Lambda(S)_n;\\
&\mathtt{Abs}_n : \Lambda(S)_{n+1} \to \Lambda(S)_n;\\
&\mathtt{Con}_n : S \to \Lambda(S)_n.
\end{aligned}$$

Define a substitution operator $\bullet : \Lambda(S)_m \times \Lambda(S)_n^m \to \Lambda(S)_n$ by induction on the first argument:

$$\begin{aligned}
\mathtt{Var}_m(i) \bullet g &= g_i;\\
\mathtt{App}_m(a, b) \bullet g &= \mathtt{App}_n(a \bullet g, b \bullet g);\\
\mathtt{Abs}_m(a) \bullet g &= \mathtt{Abs}_n(a \bullet ((g_i \bullet (x_{n+1,j})_j)_i + (x_{n+1})));\\
\mathtt{Con}_m(s) \bullet g &= \mathtt{Con}_n(s).
\end{aligned}$$

And then quotient $\Lambda(S)$ by the relation generated by

$$\mathtt{App}_m(\mathtt{Abs}_m(f), g) \sim f \bullet ((x_{n,i})_i + (g))$$

for all $f : \Lambda(S)_{n+1}$ and $g : \Lambda(S)_n$.

Example 4. We can give the sequence of sets $\Lambda(S)$ an algebraic theory structure with variables $x_{m,i} = \mathtt{Var}_m(i)$ and the substitution operator $\bullet$ defined above. We can give $\Lambda(S)$ a λ-theory structure with β-equality by taking

$$\lambda_n(f) = \mathtt{Abs}_n(f) \quad \text{and} \quad \rho_n(f) = \mathtt{App}_{n+1}(f \bullet (\mathtt{Var}_{n+1}(i))_i, \mathtt{Var}_{n+1}(n+1)).$$

Now, given a function $S \to S'$, we define a morphism $\mathbf{LamTh}(\Lambda(S), \Lambda(S'))$ by induction, sending $\mathtt{Var}(i)$, $\mathtt{App}(a, b)$ and $\mathtt{Abs}(a)$ in $\Lambda(S)$ to their corresponding elements in $\Lambda(S')$ and sending $\mathtt{Con}(s)$ to $\mathtt{Con}(f(s))$.

Note that, like with the previous example, we have a forgetful functor $\dot{()}_0 : \mathbf{LamTh} \to \mathbf{Set}$.

Lemma 37. *$\Lambda(S)$ is the free λ-theory on S.*

Proof. Let L be a λ-theory. We have an equivalence

$$\mathbf{LamTh}(\Lambda(S), L) \cong \mathbf{Set}(S, L_0),$$

sending $f : \mathbf{LamTh}(\Lambda(S), L)$ to $f_0|_S : S \to L_0$ (again, trivially natural in S and L) and conversely, $g : \mathbf{Set}(S, L_0)$ to the inductively defined $f : \mathbf{LamTh}(\Lambda(S), L)$ given by

$$\begin{aligned} f(\mathtt{Var}(i)) &= x_i; \\ f(\mathtt{App}(a, b)) &= f(a)f(b); \\ f(\mathtt{Abs}(a)) &= \lambda x_{n+1}, f(a); \\ f(\mathtt{Con}(s)) &= g(s) \bullet (). \end{aligned}$$

□

Remark 30. One can also consider this lemma a proof that we can 'interpret' the λ-calculus with some constants inside any λ-theory L if we give an interpretation of the constants as terms in L_0.

The proofs that $\Lambda(S)$ is indeed a λ-theory and that $\Lambda(f)$ and g are λ-theory morphisms, mainly work by definition of $\bullet$, λ and ρ, by induction on the terms of $\Lambda(S)$ and by invoking the properties of the λ-theory L.

Corollary 5. *The 'pure' lambda calculus is the initial λ-theory.*

Proof. If we take $S = \varnothing$, $\Lambda(\varnothing)$ is the lambda calculus, which we will call Λ. We have, like with the free algebraic theory, that $\Lambda(\varnothing)$ is the initial λ-theory. □

About Λ-algebra morphisms

Lemma 38. *Let A and B be Λ-algebras and let $f : \mathbf{Set}(A, B)$ be a function that preserves the application and the Λ-definable constants:*

$$f((x_1x_2) \bullet (a, b)) = (x_1x_2) \bullet (f(a), f(b)) \quad \textit{and} \quad f(s \bullet ()) = s \bullet ()$$

for all $a, b : A$ and $s : \Lambda_0$. Then f is a Λ-algebra morphism.

Proof. Note that for $s : \Lambda_{n+1}$ and $a : A^{n+1}$,

$$(x_1x_2) \bullet (\lambda(s) \bullet (a_i)_i, a_{n+1}) = s \bullet a.$$

By induction, we can express $s \bullet a$ using a combination of $(x_1x_2) \bullet (\cdot, \cdot)$ and $\lambda^n(s)$:

$$\begin{aligned} f(s \bullet a) &= f((x_1x_2) \bullet (\dots((x_1x_2) \bullet (\lambda^n(s) \bullet (), a_1), \dots), a_n)) \\ &= (x_1x_2) \bullet (\dots((x_1x_2) \bullet (f(\lambda^n(s) \bullet ()), f(a_1)), \dots), f(a_n)) \\ &= s \bullet (f(a_i))_i, \end{aligned}$$

so f is a Λ-algebra morphism. □

3.5.3 The free object algebraic theory

Example 5. Take a category C, with a forgetful functor $G : C \to \mathbf{Set}$ and a free functor $F : \mathbf{Set} \to C$. Let $\eta : \mathrm{id}_{\mathbf{Set}} \Rightarrow F \bullet G$ be the unit of the adjunction and let $\varphi : C(F(c), d) \cong \mathbf{Set}(c, G(d))$ be the natural equivalence of homsets.

We define an algebraic theory T with $T_n = G(F(\{1, \dots, n\}))$, projections $x_{n,i} = \eta_{\{1,\dots,n\}}(i)$. For the substitution, note that we take $t_1, \dots, t_m : T_n$, so we have $t : \{1, \dots, m\} \to G(F(\{1, \dots, n\}))$. We then take

$$s \bullet t = G(\varphi^{-1}(t))(s).$$

Now, given an object $c : C$, we can create a T-algebra $\alpha(c)$, with set $G(c)$ and action

$$s \bullet t = G(\varphi^{-1}(t))(s).$$

Also, given a morphism $f : C(c, d)$. This gives a morphism $G(f) : \alpha(c) \to \alpha(d)$. Therefore, $\alpha : C \to \mathbf{Alg}_T$ is a functor.

The proofs that T is an algebraic theory, that $G(c)$ is an algebra and that $G(f)$ is an algebra morphism mainly rely on the fact that φ is natural.

So we have a functor from C to the category of T-algebras. One can wonder whether there also is a functor the other way, or whether α is even an equivalence. This is hard to characterize precisely, but in algebra, there is a broad class of examples where the functor is an equivalence, so where C is equivalent to $\mathbf{Alg}_T$. That is probably why T is called an *algebraic theory*.

The idea is that if an object of C is a set, together with some operations between its elements, one can carefully choose some elements of T_0, T_1, T_2 etc., which act on an algebra like the specified operations.

Example 6. For C the category of monoids, $\alpha : C \to \mathbf{Alg}_T$ is an adjoint equivalence.

Note that T_n is the free monoid on n elements. Its elements can be viewed as strings $(x_1x_5x_3x_{18}\dots x_7)$ with the characters $x_1, \dots, x_n$, with the x_i the generators of the monoid, acting as the projections of the algebraic theory.

Let A be a T-algebra. We can give A a monoid structure by taking, for $a, b : A$,

$$ab = (x_1x_2) \bullet (a, b)$$

and unit element

$$1 = () \bullet ().$$

Then the laws like associativity follow from those laws on the monoid and from the fact that the action on the algebra commutes with the substitution:

$$a(bc) = (x_1(x_2x_3)) \bullet (a, b, c) = ((x_1x_2)x_3) \bullet (a, b, c) = (ab)c.$$

Note that if we take a monoid, turn it into a T-algebra and then into a monoid again, we still have the same underlying set, and it turns out that the monoid operation and unit element are equal to the original monoid operation and unit element. Therefore, α is essentially surjective. It is also fully faithful, since any T-algebra morphism respects the action of T, which makes it into a monoid morphism. Therefore, α is an adjoint equivalence.

Remark 31. In the same way, one can characterize groups, rings and R-algebras (for R a ring) as algebras of some algebraic theory. On the other hand, one can not use this method to describe fields as algebras for some theory T, because one would need to describe the inverse $z \mapsto z^{-1}$ operation as $t \bullet (z)$ for some $t : T_1$, with $zz^{-1} = 1$, but since the elements of the algebraic theory act on all (combinations of) elements of the algebra, one would be able to take the inverse $0^{-1} = t \bullet (0)$ with $00^{-1} = 1$, which would make no sense.

Remark 32. Another counterexample is the category **Top** of topological spaces. We have a forgetful functor $G : \mathbf{Top} \to \mathbf{Set}$ that just forgets the topology. On the other hand, we have a free functor $F : \mathbf{Set} \to \mathbf{Top}$ which endows a set with the discrete topology. The construction above yields the inital algebraic theory $T_n = \{1, \dots, n\}$, with an algebra action on every topological space $i \bullet (a_1, \dots, a_n) = a_i$. Now, note that we can endow the set $\{\top, \bot\}$ with four different, nonisomorphic topologies, which all yield the same T-algebra. In other words: the T-algebra structure does not preserve the topological information. Therefore, the functor $\alpha : \mathbf{Top} \to \mathbf{Alg}_T$ is not an equivalence.

3.5.4 The terminal theory

Example 7. We can create a (somewhat trivial) algebraic theory T by taking $T_n = \{\star\}$, with projections $x_i = \star$ and substitution $\star \bullet \star = \star$. Taking $\lambda(\star) = \star$ and $\rho(\star) = \star$, we give it a λ-theory structure (with β and η-equality). Checking that this is indeed an algebraic theory and even a λ-theory is trivial.

Now, given any other algebraic theory T', there exists a unique function $T'_n \to T_n$ for every n, sending everything to $\star$. These functions actually constitute an algebraic theory morphism $T' \to T$. If T' is a λ-theory, the algebraic theory morphism is actually a λ-theory morphism. Again, checking this is trivial.

Therefore, T is the terminal algebraic theory and λ-theory.

Lemma 39. *$\{\star\}$ is the only algebra of the terminal theory.*

Proof. Let A be a T-algebra. First of all, we have an element $\star_A = \star_T \bullet_0 ()$. Secondly, for all elements $\star, \star' : A$, we have

$$\star = x_1 \bullet (\star, \star') = \star \bullet (\star, \star') = x_2 \bullet (\star, \star') = \star'.$$

Therefore, $A = \{\star\}$, which allows exactly one possible T-action:

$$\star \bullet (\star, \dots, \star) = \star.$$

□

3.5.5 The endomorphism theory

Definition 41. Suppose that we have a category C and an object $X : C$, such that all powers X^n of X are also in C. The *endomorphism theory* $E(X)$ of X is the algebraic theory given by $E(X)_n = C(X^n, X)$ with projections as variables $x_{n,i} : X^n \to X$ and a substitution that sends $f : X^m \to X$ and $g_1, \dots, g_m : X^n \to X$ to $f \circ \langle g_i \rangle_i : X^n \to X^m \to X$.

Definition 42. Now, suppose that the exponential object X^X exists, and that we have morphisms back and forth $abs : X^X \to X$ and $app : X \to X^X$. Let, for $Y : C$, φ_Y be the isomorphism $C(X \times Y, X) \xrightarrow{\sim} C(Y, X^X)$. We can give $E(X)$ a λ-theory structure by setting, for $f : E(X)_{n+1}$ and $g : E(X)_n$,

$$\lambda(f) = abs \circ \varphi_{X^n}(f) \qquad \rho(g) = \varphi^{-1}_{X^n}(app \circ g).$$

The proofs that $E(X)$ is an algebraic theory and a λ-theory, use properties of the product, and naturality of the isomorphism φ_Y.

3.5.6 The theory algebra

Example 8. Let T be an algebraic theory and n a natural number. We can endow the T_n with a T-algebra structure, by taking the substitution operator of T as the T-action. Since this commutes with the substitution operator and the projections, T_n is a T-algebra.

Remark 33. Recall from 71 that T-algebras are equivalent to finite-product-preserving **Set**-valued functors on the Lawvere theory **L** associated to T. Now, recall from ((**TODO**)) that we can embed any category inside its category of **Set**-valued functors. This embeds $n : \mathbf{L}$ as the theory algebra T_n. Then the Yoneda Lemma gives a bijection

$$\mathbf{Alg}_T(T_n, A) \cong A^n = \mathbf{Set}(\{1, \dots, n\}, A)$$

natural in n and A. Explicitly, it sends $f : \mathbf{Alg}_T(T_n, A)$ to $i \mapsto f(x_i)$ and $(a_i)_i : A^n$ to $f \mapsto f \bullet a$. The natural equivalence immediately shows that T_n is the free T-algebra on n elements.

Using some additional machinery, we can combine this with the algebra pullback functor to get another functor:

Definition 43. Take a nonnegative integer n. Take an object S : **AlgTh**. For every object T : **AlgTh**, we can take the T-algebra T_n. Then every morphism f : **AlgTh**(S, T) gives an S-algebra f^*T_n. In fact, this is a functor from $(S \downarrow \textbf{AlgTh})$ to $\textbf{Alg}_S$: We send a morphism $g : (S \downarrow \textbf{AlgTh})((T, f), (T', f'))$ to

$$g_n : f^*T_n \to (f')^*T'_n.$$

This is an algebra morphism because it commutes with f, f' and the substitution of T and T': For $s : S_m$ and $t : (f^*T_n)^m$, we have

$$g_n(s \bullet_{f^*T_n} t) = g_m(f_m(s)) \bullet_{T'} (g_n(t_i))_i = s \bullet_{(f')^*T'_n} (g_n(t_i))_i.$$

The functor obviously preserves the identity morphisms and composition of morphisms, so it is indeed a functor.

3.5.7 The initial presheaf

Example 9. Let T be an algebraic theory. We can construct a T-presheaf P, with $P_n = \varnothing$. Then $\bullet : P_m \times T_n^m \to P_m$ is trivial, and the presheaf laws hold trivially.

Lemma 40. *This is indeed the initial presheaf.*

Proof. Let Q be a T-presheaf. For all n, since P_n is empty, there is only one possible function $P_n \to Q_n$. These functions trivially satisfy the presheaf morphism laws, so they constitute the unique presheaf morphism $P \to Q$. □

3.5.8 The theory presheaf

Example 10. Let T be an algebraic theory. We can endow T with a T-presheaf structure, by taking the substitution operator of T as the action on T. Since this commutes with the substitution operator and the projections, T is a T-presheaf.

Remark 34. Recall from 72 that T-presheaves are equivalent to presheaves on the Lawvere theory **L** associated to T. Now, recall from Definition 7 that we can embed any category inside its own category of presheaves. This embeds n : **L** as the power $T^n = (T_m^n)_m$ of the theory presheaf. Then the Yoneda Lemma gives a bijection

$$\textbf{Pshf}_T(T^n, Q) \cong Q_n$$

natural in n and Q. Explicitly, it sends f : $\textbf{Pshf}_T(T^n, Q)$ to $f_n(x_1, \ldots, x_n)$ and q : Q_n to $(t_i)_i \mapsto q \bullet t$.

3.5.9 The "+l" presheaf

Example 11 (The '+l' presheaf). Given a T-presheaf Q, we can construct a presheaf $A(Q, l)$ with $A(Q, l)_n = Q_{n+l}$ and, for $q : A(Q, l)_m$ and $f : T_n^m$, action

$$q \bullet_{A(Q,l)} f = q \bullet_Q ((\iota_{n,l}(f_i))_i + (x_{n+i})_i).$$

Lemma 41. *For all l and T-presheaves Q, $A(Q, l)$ is the exponential object Q^{T^l}.*

Proof. We will show that $A(-, l)$ constitutes a right adjoint to the functor $- \times T^l$. We will do this using universal arrows.

For Q a T-presheaf, take the arrow $\varphi : A(Q, l) \times T^l \to Q$ given by $\varphi(q, t) = q \bullet_Q ((x_{n,i})_i + t)$ for $q : A(Q, l)_n = Q_{n+l}$ and $t : T^l_n$.

Now, given a T-presheaf Q' and a morphism $\psi : Q' \times T^l \to Q$. Define $\tilde{\psi} : Q'_n \to A(Q, l)_n$ by $\tilde{\psi}(q) = \psi(\iota_{n,l}(q), (x_{n+i})_i)$.

Then ψ factors as $\varphi \circ (\tilde{\psi} \times \mathrm{id}_{T^l})$. Also, some equational reasoning shows that $\tilde{\psi}$ is unique, which proves that φ indeed is a universal arrow. □

Chapter 4

Previous work in categorical semantics

4.1 The correspondence between categories and typed λ-calculi

In [SH80], page 413, Scott and Lambek argue that there is a correspondence between simply typed λ-calculi and cartesian closed categories (categories with products and 'function objects').

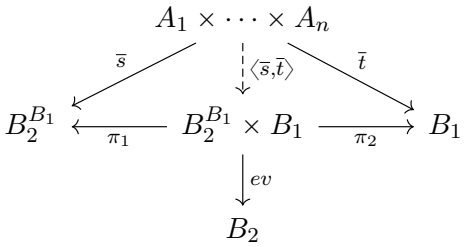
Types in the λ-calculus correspond to objects in the category.

Types $A \to B$ in the λ-calculus correspond to exponential objects B^A in the category.

Terms in the λ-calculus of type B, with free variables $x_1 : A_1, \ldots, x_n : A_n$, correspond to morphisms $A_1 \times \cdots \times A_n \to B$.

A free variable $x_i : A_i$ in a context with free variables $x_1 : A_1, \ldots, x_n : A_n$ corresponds to the projection morphism $\pi_i : A_1 \times \cdots \times A_n \to A_i$.

Given a term $s : B_1 \to B_2$ and a term $t : B_1$, both with free variables $x_1 : A_1, \ldots, x_n : A_n$, corresponding to morphisms $\overline{s} : A_1 \times \cdots \times A_n \to B_2$ and $\overline{t} : A_1 \times \cdots \times A_n \to B_1$, the application $st : B_2$ corresponds to the composite of the product morphism with the evaluation morphism $A_1 \times \cdots \times A_n \to B_2^{B_1} \times B_1 \to B_2$.



Given a term $t : B$ with free variables $x_1 : A_1, \ldots, x_n : A_n$, the abstraction $(\lambda x_n, t) : A_n \to B$ corresponds to using the adjunction corresponding to the exponential object of A_n:

$$C(A_1 \times \cdots \times A_{n-1} \times A_n, B) \simeq C(A_1 \times \cdots \times A_{n-1}, B^{A_n}).$$

4.2 The category of retracts

The next sections make extensive use of a category called **R**, which Hyland calls the 'category of retracts'. In this section, we will define the category, and show some properties about it.

Let L be a λ-theory. First of all, for $a_1, a_2 : L_0$, we define

$$\begin{aligned} a_1 \circ a_2 &= \lambda x_1, a_1(a_2 x_1); \\ (a_1, a_2) &= \lambda x_1, x_1 a_1 a_2; \\ \langle a_1, a_2 \rangle &= \lambda x_1, (a_1 x_1, a_2 x_1); \\ \pi_i &= \lambda x_1, x_1(\lambda x_2 x_3, x_{i+1}). \end{aligned}$$

Although, actually, since every one of these starts with a λ-abstraction, we need to lift the constants a_i to $\iota_{0,1}(a_i) : L_1$ to make the definitions above typecheck.

Note that $\pi_i(a_1, a_2) = a_i$ and $\pi_i \circ \langle a_1, a_2\rangle = \lambda x_1, a_i x_1$, which is exactly what we would expect of a projection.

Also, note that by replacing the x_i by x_{n+i} and the $\iota_{0,1}(a_i)$ by $\iota_{n,1}(a_i)$, we obtain definitions not only for elements of L_0, but for all L_n.

Later on, we will need not only pairs and their projects, but also n-tuples with projections. Therefore, we define

$$\begin{aligned}(a_i)_i &= (a_1, \dots, a_n) = ((\dots((c, a_1), a_2), \dots), a_n);\\ \langle a_i\rangle_i &= \langle a_1, \dots, a_n\rangle = \langle\langle\dots\langle\langle c, a_1\rangle, a_2\rangle, \dots\rangle, a_n\rangle;\\ \pi_{n,i} &= \pi_2 \circ \underbrace{\pi_1 \circ \dots \circ \pi_1}_{n-i}\end{aligned}$$

for some constant c, which usually is someting like $\lambda x_{n+1}, x_{n+1}$.

Definition 44. We define the category $\mathbf{R}$ as the Karoubi envelope of the monoid of the λ-terms without free variables under composition. That is, the category of idempotent functions:

$$\mathbf{R}_0 = \{A : L_0 \mid A \circ A = A\} \quad \text{and} \quad \mathbf{R}(A, B) = \{f : L_0 \mid B \circ f \circ A = f\},$$

with $\mathrm{id}_A = A$ and composition given by $\circ$.

Remark 35. Hyland instead defines $\mathbf{R}$ as

$$\mathbf{R}_0 = \{A : L_1 \mid A \bullet A = A\} \quad \text{and} \quad \mathbf{R}(A, B) = \{f : L_1 \mid B \bullet f \bullet A = f\},$$

writing $s \bullet t$ instead of $s \bullet (t)$ for $s, t : L_1$. Note that the following diagram commutes:

$$\begin{array}{ccc} L_1 \times L_1 & \xrightarrow{\bullet} & L_1 \\ {\scriptstyle (s,t)\mapsto((\lambda x_1, s),(\lambda x_1, t))}\downarrow & & \downarrow{\scriptstyle s\mapsto(\lambda x_1, s)} \\ L_0 \times L_0 & \xrightarrow{\circ} & L_0 \end{array}$$

Now, since for $A : \mathbf{R}$, $A = \lambda x_1, \iota_{0,1}(A)(\iota_{0,1}(A)x_1)$, we have $\lambda x_1, \iota_{0,1}(A)x_1 = A$. This shows that $s \mapsto \lambda x_1, s$ and $t \mapsto \iota_{0,1}(t)x_1$ (both on objects and morphisms) constitute an equivalence between Hyland's category of retracts and Scott's category of retracts.

Now, to give a bit more intuition for the objects of $\mathbf{R}$, we can pretend that an object $A : \mathbf{R}$ consists of the set of elements that satisfy $Aa = a$. Then a morphism $f : \mathbf{R}(A, B)$ gives $B(fa) = (B \circ f)a = fa$. This actually constitutes a functor from R to $\mathbf{Pshf}_L$:

Definition 45. We define a functor $\varphi : \mathbf{R} \to \mathbf{Pshf}_L$ by taking

$$\varphi(A)_n = \{a : L_n \mid \iota_{0,n}(A)a = a\} \quad \text{and} \quad \varphi(f)_n(a) = \iota_{0,n}(f)a$$

for $A, B : \mathbf{R}$, $f : \mathbf{R}(A, B)$ and $a : A$. The presheaf action on $\varphi(A)$ is given by the substitution of L:

$$(a, f) \mapsto a \bullet f$$

for $f : L_n^m$ and $a : L_m$ such that $\iota_{0,m}(A)a = a$.

It turns out that this embeds $\mathbf{R}$ as a full subcategory of $\mathbf{Pshf}_L$:

Lemma 42. *This functor φ is fully faithful.*

Proof. Take $A, B : \mathbf{R}$. We need to show that $f \mapsto \varphi(f)$ is an equivalence between $\mathbf{R}(A, B)$ and $\mathbf{Pshf}_L(\varphi(A), \varphi(B))$. We have a function

$$\psi : \mathbf{Pshf}_L(\varphi(A), \varphi(B)) \to \mathbf{R}(A, B), \quad g \mapsto (\lambda x_1, g_1(\iota_1(A)x_1))$$

which gives us the inverse. To see that this even typechecks, note that we have

$$\iota_1(A)x_1 : \varphi(A)_1 \quad \text{and} \quad g_1(\iota_1(A)x_1) : \varphi(B)_1 \subseteq L_1.$$

Using the fact that g is a presheaf morphism, we can show that

$$B \circ \psi(g) \circ A = \psi(g),$$

and that φ and ψ are inverses. □

Remark 36. Note that for all $A : \mathbf{R}$, we can 'reduce' any $x : L_n$ to

$$\iota_{0,n}(A)x : \varphi(A)_n$$

and in the same way, for all $A, B : \mathbf{R}$, we can turn any $f : L_0$ into a morphism $B \circ f \circ A : \mathbf{R}(A, B)$. In particular, we have $B \circ A : \mathbf{R}(A, B)$. Of course, all elements of $\varphi(A)_n$ and all elements of $\mathbf{R}(A, B)$ arise this way.

Remark 37. Note that if L is a nontrivial λ-theory, $\mathbf{R}$ is not a univalent category. To see this, note, for example, that we have an object $X := \langle \pi_1, \pi_2 \rangle : \mathbf{R}$ (corresponding to the type of 'pairs' of λ-terms). Since L_0 is a set, $X = X$ is a proposition. However, X has (at least) two automorphisms:

$$\langle \pi_1, \pi_2 \rangle \quad \text{and} \quad \langle \pi_2, \pi_1 \rangle.$$

These are the identity, and the automorphism (of order 2) that swaps the elements of the pair. To see that these are indeed different morphisms, note that applying them (or their lifted versions) to $(x_{2,1}, x_{2,2})$ gives respectively $x_{2,1}$ and $x_{2,2}$, which are distinct elements by Lemma 32.

In the next chapter, the 'universal object' $U : \mathbf{R}$, given by the identity $\lambda x_1, x_1$, plays a major role. Note that for all $A : \mathbf{R}$, we have morphisms $A : \mathbf{R}(U, A)$ and $A : \mathbf{R}(A, U)$, which exhibit A as a retract of U. Also note that $\varphi(U) = L$.

Note that $\mathbf{R}$ has many (isomorphic, but not equal for nontrivial L) terminal objects, given by $I_c := \lambda x_1, \iota_{0,1}(c) : \mathbf{R}$ for any $c : L_0$. These are terminal because for $f : A \to I_c$, we have $f = I_c \circ f = I_c$. Note that $\varphi(I_c)_n = \{\iota_{0,n}(c)\}$. We will choose $I = \lambda x_1 x_2, x_2 : \mathbf{R}$ as our main example of the terminal object.

We might wonder whether $\mathbf{R}$ also has an initial object O. However, for any $c : L_0$, we would have a constant morphism

$$I_c = \lambda x_1, \iota_{0,1}(c) : \mathbf{R}(O, U),$$

so if L is nontrivial, $\mathbf{R}$ has no initial object.

$\mathbf{R}$ has binary products with projections and product morphisms

$$A_1 \times A_2 = \langle p_1, p_2 \rangle, \quad p_i = A_i \circ \pi_i \quad \text{and} \quad \langle f, g \rangle.$$

Recall that for any object A, we have $A = \mathrm{id}_A$, which for $A_1 \times A_2$ is $\langle p_1, p_2 \rangle$ by the universal property of the product, which explains why the product is of this form.

$\mathbf{R}$ also has exponential objects

$$C^B = \lambda x_1, C \circ x_1 \circ B$$

with evaluation morphism $\epsilon_{BC} : C^B \times B \to C$ given by

$$\epsilon_{BC} = \lambda x_1, C(\pi_1 x_1(B(\pi_2 x_1))),$$

which is universal because we can lift a morphism $f : \mathbf{R}(A \times B, C)$ to a morphism $\psi(f) : \mathbf{R}(A, C^B)$ given by

$$\psi(f) = \lambda x_1 x_2, f(x_1, x_2).$$

Note that for $g : \mathbf{R}(A, C^B)$, the inverse $\psi^{-1}(g)$ is given by

$$\epsilon \circ \langle g \circ \pi_1, B \circ \pi_2 \rangle = \lambda x_1, g(\pi_1 x_1)(\pi_2 x_1) : \mathbf{R}(A \times B, C).$$

Also note that

$$\psi(\epsilon_{BC}) = \lambda x_1, C \circ x_1 \circ B = C^B = \mathrm{id}_{C^B}.$$

Note that $\varphi(C^B)_0 = \mathbf{R}(B, C)$, as we would expect.

Now, we might wonder whether there exists some $A : \mathbf{R}$ such that $\varphi(A)_0 = \varnothing$. However, for any $c : L_0$, we have $Ac : \varphi(A)_0$, because

$$Ac = (A \circ A)c = A(Ac).$$

Note that we can lift constants from $\varphi(A)_0$ to any $\varphi(A)_n$, so they are all nonempty.

Combining this with the embedding of $\mathbf{R} \hookrightarrow \mathbf{Pshf}_L$, we would expect $\mathbf{R}$ to not have all pullbacks. This is because in $\mathbf{Pshf}_L$, for a cospan $B \xrightarrow{f} A \xleftarrow{g} C$ with $f_n(B_n) \cap g_n(C_n) = \varnothing$ for some n, the pullback Q would have $Q_n = \varnothing$, which could never happen with an object coming from $\mathbf{R}$. Note, however, that this can not be made rigorous, because a fully faithful embedding reflects limits, but does not necessarily preserve them.

However, it is true that $\mathbf{R}$ does not have all pullbacks (if L is nontrivial). Consider for example the following cospan:

$$I \xrightarrow{f} U \xleftarrow{g} I$$

for different $f, g : \mathbf{R}(I, U)$. For example, $f = (\lambda x_1, \iota_{0,1}(\pi_1))$ and $g = (\lambda x_1, \iota_{0,1}(\pi_2))$. Now, take any object $Q : \mathbf{R}$. Note that we have a unique morphism $I : \mathbf{R}(Q, I)$. Then we have the following diagram:

$$\begin{array}{ccc} Q & \xrightarrow{I} & I \\ \downarrow{\scriptstyle I} & & \downarrow{\scriptstyle g} \\ I & \xrightarrow{f} & U \end{array}$$

with

$$f \circ I = f \neq g = g \circ I,$$

so the diagram does not commute, and Q is not a pullback of this cospan.

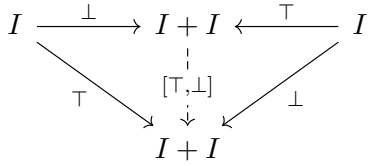
Taylor notes ([Tay86], Section 1.5) that the objects in $\mathbf{R}$ have very strong properties with respect to fixpoints. One of the properties also arises via Lawvere's fixed point theorem ([Law69], page 136): For all $B : \mathbf{R}$, since B^U is a retract of U, every endomorphism $f : B \to B$ has a fixpoint. That is, there exists $s : I \to B$ such that $f \circ s = s$. Working out the proof even yields an explicit term:

$$s = \lambda i, (\lambda x, f(xx))(\lambda x, f(xx)).$$

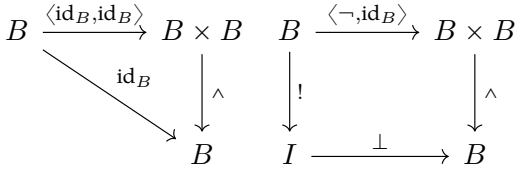
Indeed, $s = \lambda i, f((\lambda x, f(xx))(\lambda x, f(xx))) = f \circ s$, and $B \circ s = B \circ f \circ s = s = s \circ I$, so $s : \mathbf{R}(I, B)$.

From this, Taylor deduces ([Tay86], §1.5.12) that $\mathbf{R}$ does not have all coproducts if L is nontrivial, because suppose that $\mathbf{R}$ has all coproducts. Then $B = I + I$ is a 'boolean algebra object': We define $\bot, \top : I \to B$ to be the injections on the left and right components.

Since **R** is cartesian closed, binary products distribute over binary coproducts, so we have $B \times B \cong ((I \times I) + (I \times I)) + ((I \times I) + (I \times I))$, and note that $I \times I \cong I$. Using the universal property of the coproduct, we can define $\neg : B \to B$ componentwise as $\neg = [\top, \bot]$, the coproduct arrow



Note that by the definition of $\neg$, $\neg \circ \bot = \top$. Similarly, we define $\wedge, \vee : B \times B \to B$ as $\wedge = [[\bot, \bot], [\bot, \top]]$ and $\vee = [[\bot, \top], [\top, \top]]$. Using the same universal property, we can also verify some properties of $\bot, \top, \wedge$ and $\neg$, like the fact that the following diagrams commute:



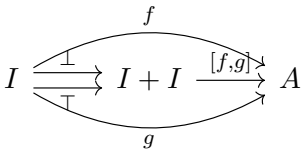
for $! : B \to I$ the terminal projection. We do this by checking, for example, that $\mathrm{id}_B \circ \top = \wedge \circ \langle \mathrm{id}_B, \mathrm{id}_B \rangle \circ \top$ and $\mathrm{id}_B \circ \bot = \wedge \circ \langle \mathrm{id}_B, \mathrm{id}_B \rangle \circ \bot$. Now, as mentioned above, every endomorphism has a fixed point. In particular, we have some $\star : I \to B$ such that $\neg \circ \star = \star$. Now, note that

$$\star = \wedge \circ \langle \mathrm{id}_B, \mathrm{id}_B \rangle \circ \star = \wedge \circ \langle \star, \star \rangle = \wedge \circ \langle \star, \neg \circ \star \rangle = \wedge \circ \langle \mathrm{id}_B, \neg \rangle \circ \star = \bot \circ ! \circ \star = \bot$$

and then

$$\bot = \star = \neg \circ \star = \neg \circ \bot = \top.$$

Now, for any object A : **R** and any two global elements $f, g : I \to A$, we have the following diagram:



and we have

$$f = [f, g] \circ \bot = [f, g] \circ \top = g.$$

In particular, for any two objects A, A' : **R**, since we have $A \circ A'$: $\mathbf{R}(A', A)$, so $\mathbf{R}(A', A)$ is nonempty, we have

$$\mathbf{R}(A', A) \simeq \mathbf{R}(I \times A', A) \simeq \mathbf{R}(I, A^{A'}) \simeq \{\star\},$$

so **R** is the trivial category and L is trivial.

4.3 Scott's Representation Theorem

The correspondence in Section 4.1 between simply-typed λ-calculi and cartesian closed categories raises a question whether such a correspondence also exists for untyped λ-calculi. Definition 41 shows that in fairly general circumstances we can take one object c in a category C and consider the morphisms $t : C(c^n, c)$ as terms in an untyped λ-calculus. Hyland calls this the 'endomorphism theory' of c.

Remark 38. To construct a *simply typed* λ-calculus from a category, we just need a cartesian closed category. In a simply typed λ-calculus, there is a lot of restriction on which terms we can apply to each other. A term of type $A \to B$ can only be applied to a term of type A, which gives a term of type B. In particular, a term can be applied only finitely many times to other terms, and every time, the result has a different type.

On the other hand, for an *untyped* λ-calculus, we need a cartesian closed category with a 'reflexive object'. This is because in the untyped λ-calculus, we can apply arbitrary terms to each other. For example, we can apply the term $(\lambda x_1, x_1 x_1)$ to itself, which would not be typable in the simply typed λ-calculus. Suppose that we have a category C and an object U such that the morphisms $C(U^n, U)$ give the untype λ-terms in n free variables, for all n. Now, given two terms $f, g : C(U^n, U)$, for the application fg, we need to consider f as a morphism in $C(U^n, U^U)$. We can do this by postcomposing with a morphism $\varphi : C(U, U^U)$. On the other hand, if $n > 0$, then $(\lambda x_n, f)$ is a morphism in $C(U^{n-1}, U^U)$, but it is a term in $n-1$ free variables, so it should be in $C(U^{n-1}, U)$. For this, we postcompose with a morphism $\psi : C(U^U, U)$. Now, for our untyped λ-calculus to have β-equality, we need $\psi \cdot \varphi = \mathrm{id}_{U^U}$, which means that the exponential U^U of U is a retract of U. This is exactly what it means for U to be a *reflexive object*: an object U in a cartesian closed category, that has a retraction onto its 'function space' U^U. Note that if we want our λ-theory to also have η-equality, the retraction must be an isomorphism.

Note that **Set** is a cartesian closed category, but that for sets X and Y, the function space X^Y has cardinality $|X|^{|Y|}$, and therefore U^U cannot be a retract of U, unless $U = \{\star\}$, in which case we have a very trivial λ-calculus: $\mathbf{Set}(U^n, U) = \{\star\}$.

During the 1960s, computer scientists sought for nontrivial examples of reflexive objects, there is a quote by Dana Scott that "Lambda-calculus has no mathematical models!"[Sco16]. However, in 1969, the same Dana Scott discovered that the category of (continuous) lattices with (Scott-)continuous functions between them has a nontrivial reflexive object, with the retraction onto the function spaces being even an isomorphism. Such a reflexive object D_∞ is obtained by starting with an arbitrary lattice D_0, iteratively taking $D_{n+1} = D_n^{D_n}$, with a retraction (in the 'wrong' direction) $r : D_n^{D_n} \to D_n$, and then passing to the limit $D_\infty = \lim_{\leftarrow} D_n$ (for the main result, see Theorem 4.4 in [Sco72], page 127).

Since Lambek showed an equivalence between simply typed λ-calculi and cartesian closed categories, and since we have a construction for an untyped λ-calculus from a cartesian closed category with a reflexive object, we can wonder whether this construction constitutes contitutes an equivalence between untyped λ-calculi and some class of categories. This question finds a partial answer in the following theorem, originally proven in a very syntactical way by Dana Scott (see [SH80], page 418).

Theorem 1. *We can obtain every untyped λ-calculus as the endomorphism theory of some object in some category.*

Proof. Let L be a λ-theory. Scott considers its category of retracts $\mathbf{R}$, with 'universal object' U.

Note that U^U is a retract of U, so U is a reflexive object.

Therefore, $E(U)$, the endomorphism theory of U, has a λ-theory structure. Note that the finite powers of U in $\mathbf{R}$ are given by $U^0 = I$ and $U^{n+1} = U^n \times U$.

We have $E(U)_n = \mathbf{R}(U^n, U) = \{f : L_0 \mid U \circ f \circ U^n = f\}$. The variables of $E(U)$ are the projections $\pi_{n,i}$ of U^n. The substitution is given by composition with the product morphism:

$$f \bullet g = f \circ \langle\langle\langle I, g_1\rangle, \dots\rangle, g_n\rangle.$$

We have $U^U = \lambda f, U \circ f \circ U = \lambda x_1 x_2, x_1 x_2$. Using the equivalence $\mathbf{R}(U^n \times U, U) \simeq \mathbf{R}(U^n, U^U)$ and the retraction $U^U : U \to U^U$, the abstraction and application λ and ρ are given by

$$\lambda(f) = \lambda x_1 x_2, \iota_{0,2}(f)(x_1, x_2), \quad \rho(g) = \lambda x_1, \iota_{0,1}(g)(\pi_1 x_1)(\pi_2 x_1).$$

for $f : \mathbf{R}(U^{n+1}, U)$ and $g : \mathbf{R}(U^n, U)$.

Now, we have bijections $\psi_0 : E(U)_0 \xrightarrow{\sim} L_0$, given by

$$\psi_0(f) = f(\lambda x_1, x_1) \quad \text{and} \quad \psi_0^{-1}(g) = \lambda x_1, \iota_{0,1}(g).$$

We can extend this to any n, by reducing any term to a constant by repeatedly using λ, then applying the bijection, and then lifting it again using ρ. Explicitly, we obtain

$$\psi_n(f) = \iota_{0,n}(f)(x_i)_i, \quad \text{and} \quad \psi_n^{-1}(g) = \lambda x_1, g \bullet (\pi_{n,i} x_1)_i.$$

It is not hard to verify that this is indeed a bijection, using at one point the fact that $f : \mathbf{R}(U^n, U)$ is defined by $f \circ U^n = f$, for

$$U^n = \langle \pi_{n,i} \rangle_i.$$

It is also pretty straightforward to check that

$$\begin{aligned} \psi(\pi_{n,i}) &= x_i, & \psi(f) \bullet (\psi(g_i))_i &= \psi(f \bullet g), \\ \psi(\lambda(h)) &= \lambda(\psi(h)), & \psi(\rho(h')) &= \rho(\psi(h')) \end{aligned}$$

for $f : E(U)_m$, $g : E(U)_n^m$, $h : E(U)_{n+1}$ and $h' : E(U)_n$. Therefore, ψ is an isomorphism of λ-theories. □

4.4 The Taylor Fibration

In his dissertation, Paul Taylor shows that **R** is not only cartesian closed, but also *relatively cartesian closed*.

In Section 1.7, we studied internal and external representations of families of objects in a category and how they behaved under substitutions (pullbacks). This was to arrive at a definition for dependent products and sums, as the right and left adjoints to the pullback (or substitution) functor $\alpha^* : (C \downarrow A) \to (C \downarrow B)$ along some morphism $\alpha : C(B, A)$.

Now, some categories are not locally cartesian closed. That is: not all pullback/substitution functors α^* exist or have a right adjoint. For example, **R** does not have all pullbacks, so the substitution functors do not always exist. In such a category C, the functor $C^2 \to C$ is not a fibration. One way to look at this, is that in such a category not every morphism $X \to A$ represents a family of objects. In such a category, we can carefully choose a subset of the morphisms to represent our indexed families. We will call a morphism that we choose to represent an indexed family a *display map*. In most cases, we have quite a bit of choice how big we want our subtype of display maps to be. However, to make sure that indexed families are well-behaved, the subtype of display maps needs to have some properties:

1. The pullback of a display map along any morphism exists and is a display map.
2. The composite of two display maps is a display map.
3. C has a terminal object and any terminal projection is a display map.

Remark 39. A 'maximal' example of a subtype of display maps is, for example, in the category **Set**, where we can take our subtype of display maps to equal the full type of all morphisms of C.

Remark 40. A 'minimal' example of a subtype of display maps is the subtype of (maps isomorphic to) product projections in a category with finite products. In this case, all indexed families are constant, and then dependent sums and products become binary products and exponential objects.

Now, let C be a category with a subtype of display maps $D \subseteq C$. We will denote the subtype of display maps from X to A with $D(X, A) \subseteq C(X, A)$. For any $A : C$, we define the category $(C \downarrow_D A)$ as a full subcategory of the slice category $(C \downarrow A)$, with as objects the display maps $f : D(X, A)$. The morphisms between two objects of $(C \downarrow_D A)$ are still all the morphisms of $(C \downarrow A)$ (i.e. the morphisms of C that commute with the display maps).

Note that for the terminal object $I : C$, $(C \downarrow_D I)$ is still equivalent to C, since every terminal projection is a display map.

Also note that since the pullback of display maps against any map exist (and are display maps again), we get a pullback functor $\alpha^* : (C \downarrow_D A) \to (C \downarrow_D B)$. This is the restriction of the pullback functor $\alpha^* : (C \downarrow A) \to (C \downarrow B)$, if it exists.

Note that since composing two display maps gives a display map again, and since the dependent sum is given by postcomposition, α^* has a left adjoint for all display maps α. That is: the fiber categories $(C \downarrow_D A)$ have dependent sums over display maps.

The question whether the fiber categories $(C \downarrow_D A)$ also have dependent products over display maps, brings us to the definition of relative cartesian closedness.

Definition 46. A category C is *cartesian closed relative* to a class of display maps D, if the substitution functors α^* along display maps have right adjoints.

Now, analogously to 12, Taylor shows:

Lemma 43. *If a category C is cartesian closed relative to a class of display maps D, then the fiber categories $(C \downarrow_D A)$ are cartesian closed and the substitution functors α^* preserve this structure.*

Proof. Taylor proves this in a series of lemmas leading up to §4.3.7 in [Tay86]. It is also proved as Proposition 6 of [HP89]. □

4.4.1 Taylor's proof

As Hyland remarks, Taylor shows that **R** is relatively cartesian closed using a very syntactical argument, in the spirit of Scott.

He starts out with a kind of 'external' representation of indexed families $(X_a)_a$ in **R**. He denotes these as 'functions' $A \to \mathbf{R}_0$. They are the elements $X : L_0$ with

$$X \circ A = X \quad \text{and} \quad (\lambda x_1, (X x_1) \circ (X x_1)) = X.$$

These X form a category $\mathbf{R}^A$, with the morphisms in $\mathbf{R}^A(X, Y)$ given by $f : L_0$ with $f \circ A = f$ and $(\lambda x_1, (Y x_1) \circ (f x_1) \circ (X x_1)) = f$. The identity on X is given by X, and the composition is given by $f \cdot g = \lambda x_1, g x_1 \circ f x_1$.

Taylor shows that these categories $\mathbf{R}^A$ are cartesian closed. He notes that assigning these categories $\mathbf{R}^A$ to objects $A : \mathbf{R}$ again constitutes a contravariant (pseudo)functor $\mathbf{R}^{\mathrm{op}} \to \mathbf{Cat}$, sending morphisms $A \to B$ to precomposition functors $\mathbf{R}^B \to \mathbf{R}^A$.

For $A : \mathbf{R}$, we introduce the combinator

$$\sum_A = \lambda x_1 x_2, (A(\pi_1 x_2), x_1(\pi_1 x_2)(\pi_2 x_2)).$$

For $A : \mathbf{Set}$, we have an equivalence between the elements of $\mathbf{Set}^A$ and the elements of the fiber $(\mathbf{Set} \downarrow A)$ of the fibration $\mathbf{Set}^2 \to \mathbf{Set}$. Now, for $A : \mathbf{R}$, $\sum_A$ gives a functor from $\mathbf{R}^A$ to $(\mathbf{R} \downarrow A)$. Note that it sends objects $X : \mathbf{R}^A$ to $\sum_A X$, together with a projection morphism $p_X = A \circ \pi_1 : \sum_A X \to A$. Note that with the embedding in Definition 45 into $\mathbf{Pshf}_L$, we can consider $\sum_A X$ as consisting of pairs (a, x) with $a : A$ and $x : Xa$, which is exactly what we would expect from a dependent sum.

Now, $\sum_A$ is fully faithful: given a morphism $g : \mathbf{R}\,(\sum_A X, \sum_A Y)$, we take

$$\psi(g) = \lambda x_1 x_2, \pi_2(g(x_1, x_2)).$$

For verifying that $\psi(g)$ is indeed a morphism in $\mathbf{R}^A(X,Y)$, it helps to recall that $g \circ \sum_A X = g = \sum_A Y \circ g$ and $p_Y \circ g = p_X$. Since $\psi(\sum_A f) = f$ for all $f : \mathbf{R}^A(X,Y)$ and $\sum_A \psi(g) = g$ for all $g : \mathbf{R}(\sum_A X, \sum_A Y)$, $\sum_A$ is indeed fully faithful.

On the other hand, $\sum_A$ is generally not surjective. However, we can choose our subtype D of display maps in such a way that the restriction $\mathbf{R}^A \to (C \downarrow_D A)$ becomes essentially surjective.

Definition 47. For $\mathbf{R}$, we will consider a morphism $f : X \to A$ to be a display map if we have some $Y : \mathbf{R}^A$ like above, and some isomorphism $g : (X, f) \xrightarrow{\sim} (\sum_A Y, p_Y)$ in $(\mathbf{R} \downarrow A)$.

Remark 41. Hyland actually gives a different characterization of Taylor's display maps. He claims that Taylor takes the display maps $X \to A$ to be the retracts in $(\mathbf{R} \downarrow A)$ of $p_1 : A \times U \to A$, the projection onto the first coordinate. The two characterizations are equivalent:

Given $Y : \mathbf{R}^A$, intuitively Ya is a retract of U for all a. Concretely, both morphisms $r : A \times U \to \sum_A Y$ and $s : \sum_A Y \to A \times U$ are given by $\sum_A Y$ and these commute with the projection to A. Therefore, if we have an isomorphism $g : X \xrightarrow{\sim} \sum_A Y$ in $(\mathbf{R} \downarrow A)$, then $\sum_A Y \cdot g^{-1}$ and $g \cdot \sum_A Y$ make X into a retract of $A \times U$.

Conversely, given a retraction $r : A \times U \to X$ and section $s : X \to A \times U$ in $(\mathbf{R} \downarrow A)$, we have

$$Y = \lambda x_1 x_2, \pi_2(s(r(x_1, x_2))) : \mathbf{R}^A.$$

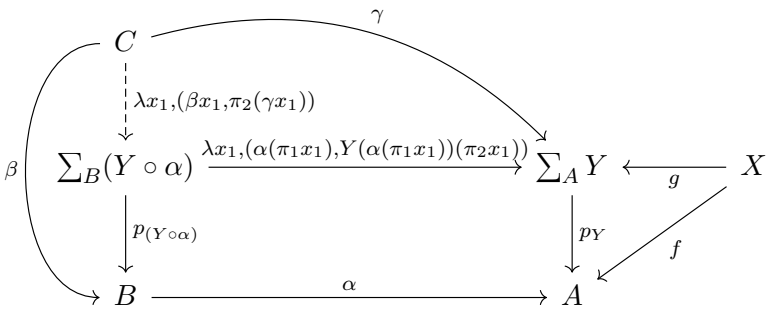
Using the properties of r and s and the definition of $\sum_A Y$, one can show that r gives a morphism $\sum_A Y \to X$ and s gives a morphism $X \to \sum_A Y$, and that $r \circ s = \mathrm{id}_{\sum_A Y}$. Combined with the fact that $s \circ r = \mathrm{id}_X$, this shows that X is isomorphic to $\sum_A Y$.

Remark 42. Recall that if L is nontrivial, $\mathbf{R}$ is not univalent, and the existence of $Y : \mathbf{R}^A$ with the isomorphism $g : X \xrightarrow{\sim} \sum_A Y$ is not a proposition. Take, for example, $Y : \mathbf{R}^I$ given by $(\lambda x_1, U \times U)$. Recall that $(\mathbf{R} \downarrow I)$ is equivalent to $\mathbf{R}$. Under this equivalence $\sum_I Y$ is equivalent to $U \times U$. As mentioned before, $\sum_I Y$ has (at least) two distinct isomorphisms to itself: the identity and the isomorphism that swaps both sides of the product.

In a similar way, being a retract of $A \times U$ is not a proposition. Therefore, if we want the class of display maps to really be a subclass of the class of morphisms, we need the existence of $Y : \mathbf{R}^A$ and the isomorphism $g : X \xrightarrow{\sim} \sum_A Y$, or the existence of the retraction $A \times U \to X$, to mean *mere existence* in this case. This means that we cannot use these in constructions, except for mere propositions.

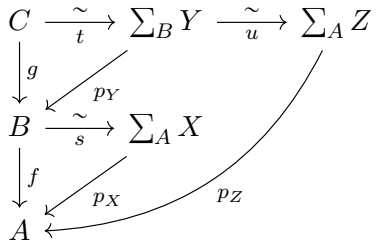
Taylor shows that these display maps indeed satisfy the three properties mentioned above. However, in univalent foundations, there are some subtleties in the case of pullbacks:

1. Given a display map $(X, f) : (\mathbf{R} \downarrow_D A)$ and a morphism $\alpha : \mathbf{R}(B, A)$. Recall that this means that there merely exists a $Y : \mathbf{R}^A$ and an isomorphism $X \xrightarrow{\sim} \sum_A Y$ in $(\mathbf{R} \downarrow A)$. Therefore, we have mere existence of a pullback $\sum_B (Y \circ \alpha)$ of $\sum_A Y$ along α. For all $C : \mathbf{R}$ with $\beta : \mathbf{R}(C, B)$ and $\gamma : \mathbf{R}(C, \sum_A Y)$ that make the square commute, we have the morphism $\lambda x_1, (\beta x_1, \pi_2(\gamma x_1)) : \mathbf{R}(C, \sum_B (Y \circ \alpha))$.



By the isomorphism between X and $\sum_A Y$, $\sum_B (Y \circ \alpha)$ is also a pullback of X: by postcomposing with g and its inverse, we see that morphisms from $\sum_B (Y \circ A)$ and C to X are equivalent to morphisms to $\sum_A Y$.

2. Given display maps $f : B \to A$ and $g : C \to B$. We have $X : \mathbf{R}^A$ and $Y : \mathbf{R}^B$ with isomorphisms $s : B \xrightarrow{\sim} \sum_A X$ and $t : C \xrightarrow{\sim} \sum_B Y$. We need to show that $f \circ g : C \to A$ is also a display map.



Intuitively, $\sum_B Y \cong \sum_A Z$ represents

$$\sum_{b:\sum_{a:A} Xa} Yb \cong \sum_{a:A} \sum_{x:Xa} Y(a,x),$$

by the isomorphism that relates $((a,x),y)$ to $(a,(x,y))$. Therefore, consider

$$Z = \lambda x_1 x_2, (X x_1 (\pi_1 x_2), Y(s^{-1}(x_1, \pi_1 x_2))(\pi_2 x_2)) : \mathbf{R}^A.$$

we have an isomorphism $u : \sum_B Y \xrightarrow{\sim} \sum_A Z$ given by:

$$u = \lambda x_1, (Aa, (Xax, Yby))$$

with

$$b = \pi_1 x_1; \quad a = \pi_1(sb); \quad x = \pi_2(sb); \quad y = \pi_2 x_1,$$

and

$$u^{-1} = \lambda x_1, (s^{-1}(Aa, Xax), Yby)$$

with

$$a = \pi_1 x_1; \quad x = \pi_1(\pi_2 x_1); \quad b = s^{-1}(a,x); \quad y = \pi_2(\pi_2 x_1)$$

and then $u \circ t$ is an isomorphism in $(\mathbf{R} \downarrow A)$ between $(C, f \circ g)$ and $(\sum_A Z, p_Z)$.

3. Let $B \to I$ be a terminal projection. Consider $Y = (\lambda x_1, B) : \mathbf{R}^I$. We have

$$\sum_I Y = \lambda x_1, ((\lambda x_2, x_2), B(\pi_2 x_1)) = \langle I \circ \pi_1, B \circ \pi_2 \rangle = I \times B,$$

which is isomorphic to B.

Remark 43. There is an awful lot of properties to verify here (e.g. that some terms are elements of some $\mathbf{R}^A$, that others are morphisms in some $\mathbf{R}(\sum_A X, \sum_B Y)$, that some diagrams commute), but most of it boils down to writing down the equations that should hold, reducing them via β-equality, and then using the facts that objects $A : \mathbf{R}$ are idempotent, that morphisms $f : \mathbf{R}(A,B)$ equal $B \circ f \circ A$ and that objects $X : \mathbf{R}^A$ satisfy $X \circ A = X$ and $Xa(Xax) = Xax$ for all a and x. Note that for a morphism $f : \sum_A X \to B$,

$$f(Ax_1, x_2) = f(x_1, x_2) = f(x_1, Xx_1x_2),$$

so f 'absorbs' such instances of A and X, so to speak.

Remark 44. To talk about 'relatively cartesian closed', we need a pullback functor $\alpha^* : (\mathbf{R} \downarrow_D A) \to (\mathbf{R} \downarrow_D B)$ for all (display maps) $\alpha : B \to A$. However, as remarked before, since the definition of display maps involves a propositional truncation, we only have mere existence of pullbacks, and to pass from the statement "for all f, there exists a pullback $\alpha^* f$" to the statement "there exists a function that sends every morphism $f : (\mathbf{R} \downarrow_D A)$ to its pullback $\alpha^* f : (\mathbf{R} \downarrow_D B)$", we need to assume the axiom of choice.

However, note that we have a weak equivalence between strict categories (categories in which the type of objects is a homotopy set) $\sum_A : \mathbf{R}^A \xrightarrow{\sim} (\mathbf{R} \downarrow_D A)$. If we assume the axiom of choice, $\sum_A$ becomes an equivalence of categories (see bullet (ii) in the introduction of Chapter 9 of [Uni13]) and then our pullback functor is given by

$$\sum_A^{-1} \bullet (\alpha \cdot -) \bullet \sum_B,$$

where $(\alpha \cdot -) : \mathbf{R}^A \to \mathbf{R}^B$ denotes the functor given by precomposition with α.

This brings us to the main theorem of this section ([Tay86], 5.1.8).

Theorem 2. *If we assume the axiom of choice,* $\mathbf{R}$ *is cartesian closed, relative to the given class of display maps.*

Proof. Let $\alpha : B \to A$ be a display map. We have mere existence of some $X : \mathbf{R}^A$ and an isomorphism $g : (X, \alpha) \xrightarrow{\sim} (\sum B, p_X)$. We need to show that there is a right adjoint to the pullback functor α^*. Since under assumption of the axiom of choice, α^* is defined as a composition of a precomposition functor with two equivalences of categories, this is equivalent to showing that there is a right adjoint to the precomposition functor $(\alpha \cdot -) : \mathbf{R}^A \to \mathbf{R}^B$ (note that $\alpha \cdot -$ sends f to $f \circ \alpha$).

Now, intuitively, any $Y : \mathbf{R}^B$ denotes an indexed family $((Y(a,b))_{b:Xa})_{a:A}$. We want the right adjoint $\alpha_* Y$ to be the indexed family of dependent products $(\prod_{b:B_a} Y(a,b))_a$. An element f of the dependent product $\alpha_* Ya$ is then a function from Xa, that satisfies $fb : Y(a,b)$ for all $b : Xa$. Therefore, for all $a : A$ and all $f : L_0$, we want $(\alpha_* Ya)f = f$ to mean "$f \circ Xa = f$ and, $fb : Y(a,b)$ for all $b : Xa$". We encode these two parts as

$$\lambda x_1, x_1 \circ Xa \quad \text{and} \quad \lambda x_1 x_2, Y(g^{-1}(a, x_2))(x_1 x_2).$$

It turns out that these are idempotents and they commute, so we can compose them into one object

$$\alpha_* Ya = \lambda x_1 x_2, Y(g^{-1}(a, x_2))(x_1(Xax_2)) : \mathbf{R}$$

and we define the combinator

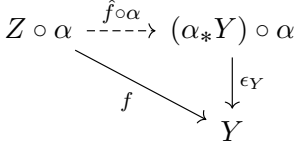
$$\alpha_* = \lambda x_1 x_2 x_3 x_4, x_1(g^{-1}(x_2, x_4))(x_3(Xx_2x_4)).$$

Now, applying α_* to both objects and morphisms in $\mathbf{R}^A$ gives a functor $\alpha_* : \mathbf{R}^B \to \mathbf{R}^A$.

Note that for all $Y : \mathbf{R}^B$ and all b, we have a morphism $(\lambda x_1, x_1(\pi_2(gb))) : \mathbf{R}(\alpha_* Y(\alpha b), Yb)$. Making this parametric in b gives us a counit

$$\epsilon_Y = \lambda x_1 x_2, Y x_1(x_2(\pi_2(gx_1))) : \mathbf{R}^B((\alpha_* Y) \circ \alpha, Y).$$

To show that $(\alpha \cdot -) \dashv \alpha_*$, we need to show that for all $Y : \mathbf{R}^B$, $(\alpha_* Y, \epsilon_Y)$ is a universal arrow from $(\alpha \cdot -)$ to Y. Recall the following diagram:



Now, suppose that we have some $Z : \mathbf{R}^A$ and some $f : \mathbf{R}^B(Z \circ \alpha, Y)$. We define

$$\hat{f} = \lambda x_1 x_2 x_3, f(g^{-1}(x_1, x_3))(Z x_1 x_2) : \mathbf{R}^A(Z, \alpha_* Y)$$

and we have $((\hat{f}) \circ \alpha) \cdot \epsilon_Y = f$. Now, suppose that we have some (other) $\hat{f}' : \mathbf{R}^A(Z, \alpha_*(Y))$ such that $(\hat{f}' \circ \alpha) \cdot \epsilon_Y = f$. Then substituting $(\hat{f}' \circ \alpha) \cdot \epsilon_Y$ for f in $\hat{f}$ yields

$$\hat{f} = \lambda x_1, (\alpha_* Y x_1) \circ (\hat{f}' x_1) = \hat{f}',$$

so $\hat{f}$ is unique and $(\alpha_* Y, \epsilon_Y)$ is a universal arrow, which concludes the proof that $(\alpha \cdot -) \dashv \alpha_*$ and we see that $\mathbf{R}$ is cartesian closed relative to the given class of display maps. □

Remark 45. Recall that in the definition of α_*, we composed commuting idempotents

$$\lambda x_1 x_2, x_1(X a x_2) \quad \text{and} \quad \lambda x_1 x_2, Y(a, x_2)(x_1 x_2).$$

In his PhD thesis, Taylor instead composes the idempotents

$$\lambda x_1, x_1(X a x_2) \quad \text{and} \quad \lambda x_1 x_2, Y x_2(x_1 x_2).$$

Note that in the first term, x_2 plays the role of an element of Xa, whereas in the second term, it plays the role of an element of B. Of course, if we worked in **Set**, we would indeed have $Xa \subseteq \bigsqcup_{a:A} X_a \cong B$. However, in $\mathbf{R}$, the 'terms' of $B \cong \sum_A X$ look like pairs (a, x) with $a : A$ and $x : Xa$, so we cannot consider terms of Xa to be terms of B.

Therefore, the idempotents that he uses do not commute, and the resulting term

$$\lambda x_1 x_2, Y(B x_2)(x_1(X a x_2))$$

(notice the redundant usage of B) is not idempotent.

Chapter 5

Hyland's paper

5.1 Scott's Representation Theorem

Theorem 3. *Any λ-theory L is isomorphic to the endomorphism λ-theory $E(L)$ of $L : \mathbf{Pshf}_L$, which is L, viewed as a presheaf in its own presheaf category.*

Proof. First of all, remember that L is indeed exponentiable and that $L^L = A(L, 1)$. Now, since L is a λ-theory, we have sequences of functions back and forth $\lambda_n : A(L, 1)_n \to L_n$ and $\rho_n : L_n \to A(L, 1)_n$. These commute with the L-actions, so they constitute presheaf morphisms and $E(L)$ is indeed a λ-theory.

Remark 34 gives a sequence of bijections $\varphi_n : \mathbf{Pshf}_L(L^n, L) \cong L_n$ for all n, sending $F : \mathbf{Pshf}_L(L^n, L)$ to $F(x_1, \ldots, x_n)$, and conversely sending $s : L_n$ to $((t_1, \ldots, t_n) \mapsto s\bullet(t_1, \ldots, t_n))$. It considers λ-terms in n variables as n-ary functions on the λ-calculus. Therefore, it should come as no surprise that φ preserves the x_i, $\bullet$, ρ and λ, which makes it into an isomorphism of λ-theories and this concludes the proof. □

5.2 Relations between the categories

As mentioned before, we have a fully faithful embedding of **R** into $\mathbf{Pshf}_L$.

Also, we have a functor from the Lawvere Theory **L** (see Lemma 70) associated to L into **R**, sending $n : \mathbf{L}$ to $U^n : \mathbf{R}$. By Scott's representation theorem, this functor can map morphisms as follows:

$$\mathbf{L}(m, n) = L_m^n \cong \mathbf{R}(U^m, U)^n \cong \mathbf{R}(U^m, U^n),$$

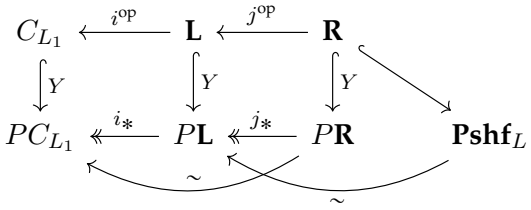
which immediately shows that this functor is a fully faithful embedding.

Note that if we consider L_1 as a monoid, with operation $\bullet$ and unit x_1, and if we consider this monoid as a category, we can embed this (fully faithfully) into **L**. This gives the following sequence of embeddings:

$$L_1 \hookrightarrow \mathbf{L} \hookrightarrow \mathbf{R} \hookrightarrow \mathbf{Pshf}_L$$

Note that the composition of the first two morphisms sends the object of the category L_1 to $(\lambda x_1, x_1) : \mathbf{R}$, which is exactly the usual embedding of L_1 into its Karoubi envelope.

Write $i : L_1^{\mathrm{op}} \hookrightarrow \mathbf{L}^{\mathrm{op}}$ and $j : \mathbf{L}^{\mathrm{op}} \hookrightarrow \mathbf{R}^{\mathrm{op}}$ for the embeddings on the opposite categories. By Corollary 1, the first two functors in this sequence essentially yield surjective functors on the presheaf categories:



The two equivalences in this diagram are due to Lemma 72 and Corollary 3.

Lemma 44. *The precomposition functors i_* and j_* are adjoint equivalences.*

Proof. We will show that j_* is an adjoint equivalence. From the 'two out of three' property, it follows then that i_* is also an adjoint equivalence.

Lemma 15 shows that $\eta : \mathrm{id}_{P\mathbf{L}} \Rightarrow \mathrm{Lan}_j \bullet j_*$ is a natural isomorphism. To complete the proof that j_* is an adjoint equivalence, we just have to show that $\epsilon : j_* \bullet \mathrm{Lan}_j \Rightarrow \mathrm{id}_{P\mathbf{L}}$ is a natural isomorphism as well.

To this end, take $F : P\mathbf{R}$. By Lemma 15, we have the isomorphism

$$\eta^{-1}_{j_*F} : j_*(\mathrm{Lan}_j(j_*F)) \cong j_*F.$$

Since functors preserve isomorphisms, we have $i_*(j_*(\mathrm{Lan}_j(j_*F))) \cong i_*(j_*F)$. However, since $j_* \bullet i_*$ is an equivalence and in particular fully faithful, this corresponds to an isomorphism

$$\mathrm{Lan}_j(j_*F) \cong F.$$

Now we only need to prove that its morphism is equal to ϵ_F. Equivalently, we need to prove that $i_*(j_*\epsilon)$ is equal to the morphism $i_*\eta^{-1}_{j_*F}$, or that $j_*\epsilon$ equals $\eta^{-1}_{j_*F}$.

Take $n : \mathbf{L}$. We need to show that $\epsilon_F(j(n)) = \eta^{-1}_{j_*F}(n)$ as functions from $(\mathrm{Lan}_j(j_*F))(j(n))$ to $F(j(n))$. Note that the diagram for $(\mathrm{Lan}_j(j_*F))(j(n))$ consists of $F(j(m))$ for all $f : \mathbf{R}(j(m), j(n))$. Now, as it turns out, both $\epsilon_F(j(n))$ and $\eta^{-1}_{j_*F}(n)$ are defined as the colimit arrow of $F(f) : \mathbf{Set}(F(j(m)), F(j(n)))$ for all $f : \mathbf{R}(j(m), j(n))$, which concludes the proof. □

Lemma 45. *The embeddings of $\mathbf{R}$ into $\mathbf{Pshf}_L$ and $P\mathbf{R}$ commute with the equivalence between these categories.*

Proof. Note:

$$(Y \bullet j_*)(A)(n) = (j \bullet (Y(A)))(n) = Y(A)(j(n)) = \mathbf{R}(U^n, A)$$

$$(\iota \bullet \sim)(A)(n) = \{a : L_n \mid \iota_{0,n}(A)a = a\}$$

By Scott's representation theorem, we have $\varphi_n : \mathbf{R}(U^n, U) \xrightarrow{\sim} L_n$, given by

$$\varphi_n(f) = \iota_{0,n}(f)(x_i)_i.$$

Restricting this equivalence to the morphisms to A yields

$$\mathbf{R}(U^n, A) \cong \{a : L_n \mid Aa = a\}.$$

Of course, this is natural in A, since for $g : \mathbf{R}(A, B)$, postcomposing elements of $\mathbf{R}(U^n, A)$ by g is equivalent to applying g to the elements of $\{a : L_n \mid Aa = a\}$.

Lastly, for $f : L^n_m$, the presheaf actions

$$(Y \bullet j_*)(A)(f) : \mathbf{R}(U^n, A) \to \mathbf{R}(U^m, A)$$

and

$$(\iota \bullet \sim)(A)(f) : \{a : L_n \mid \iota_{0,n}(A)a = a\} \to \{a : L_m \mid \iota_{0,n}(A)a = a\}$$

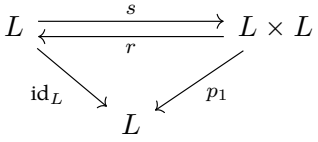
are in fact given by the substitution operations $\bullet$ in $E(U)$ and L. Since Scott's representation theorem shows that $E(U)$ is isomorphic to L as a λ-theory, these substitutions are compatible with the φ_n, which shows naturality in n. □

5.3 Locally cartesian closedness of the category of retracts

Recall that Paul Taylor constructed a category of 'dependent retracts' $\mathbf{R}^A$, and then defined a class of display maps for $\mathbf{R}$ in such a way that it gives a weak equivalence $\sum_A : \mathbf{R}^A \xrightarrow{\sim} (\mathbf{R} \downarrow_D A)$. I.e., a map $f : \mathbf{R}(X, A)$ is defined to be a display map if it is isomorphic to something in the image of the functor $\sum_A$. We saw that these display maps can also be characterized as the retracts of the projection $p_1 : A \times U \to A$ in $(\mathbf{R} \downarrow A)$.

Now, noticing that we can embed $\mathbf{R}$ as a full subcategory in $\mathbf{Pshf}_L$, Hyland attempts to do a similar thing for $\mathbf{Pshf}_L$. Noting that the universal object $U : \mathbf{R}$ is embedded as the theory presheaf $L : \mathbf{Pshf}_L$, he defines a full subcategory $\mathbf{R}(A) \subseteq (\mathbf{Pshf}_L \downarrow A)$, consisting precisely of the retracts of $\Delta_A L$ in $(\mathbf{Pshf}_L \downarrow A)$.

Remark 46. Again, being a retract of $\Delta_A L$ is not a mere proposition. For example, consider $\mathrm{id}_L : D(L, L)$:



There are multiple possible retraction-section pairs (r, s) that make the diagram commute. For example, we can take s to be the diagonal embedding $\Delta : L \to L \times L$ (sending l to (l, l)). Then we can take $r = p_1$ or $r = p_2$. We can also take $r = \pi_1$ and $s = \langle \mathrm{id}_L, g \rangle$ the product morphism of id_L and any (possibly constant) endomorphism $g : L \to L$.

Therefore, we need to take the propositional truncation of the existence of r and s.

Remark 47. Taylor would write $(\mathbf{Pshf}_L \downarrow_D A)$ instead of $\mathbf{R}(A)$, with $D(X, A)$ the retracts $f : X \to A$ of $\Delta_A L$. However, note that D is not a class of display maps. For example, for D to be a class of display maps, all terminal projections $! : X \to T$ need to be in D. Since for any category C with terminal object T, we have an equivalence $(C \downarrow T) \cong C$, having all terminal projections in D is equivalent to every object in $\mathbf{Pshf}_L$ being a retract of L.

However, if we take X to be the initial (empty) presheaf, we see that there are no morphisms $L \to X$, because the L_n are nonempty and the X_n are empty.

There also is a more category-theoretical argument: Suppose that every object of $\mathbf{Pshf}_L$ is a retract of L. Via the equivalence $\mathbf{Pshf}_L \cong L_1$, we see that every object of PC_{L_1} is a retract of $Y(\star)$. However then the embedding of $\hat{\mathbf{R}}$ (Definition 20) into PC_{L_1} is an adjoint equivalence. Since $Y_{\mathbf{R}} \bullet \iota^{\mathrm{op}}_{L_1 *} : \mathbf{R} \to \hat{\mathbf{R}}$ and $\iota^{\mathrm{op}}_{L_1 *} : P\mathbf{R} \to PC_{L_1}$ are weak equivalences, (Lemma 20, Corollary 3), $Y_{\mathbf{R}} : \mathbf{R} \to P\mathbf{R}$ is a weak equivalence as well, but the Yoneda embedding is never essentially surjective. For any category C, the constant empty presheaf $E : c \mapsto \varnothing$ is not representable, since for all $c : C$, $\mathrm{id}_c = Y(c)(c)$, so $Y(c)$ is not isomorphic to E.

Recall that $\mathbf{Pshf}_L$ is isomorphic to the presheaf category $P\mathbf{L}$, so it is an elementary topos, so it is locally cartesian closed (Example 5.2.5 and Theorem 5.8.4, [Bor94], Volume 3). Therefore, for a morphism $f : \mathbf{Pshf}_L(Y, X)$, we have adjunctions

$$(\mathbf{Pshf}_L \downarrow X) \underset{\prod_f}{\overset{\sum_f = - \cdot f}{\underset{-f^* \to}{\overset{\bot}{\underset{\bot}{}}}}} (\mathbf{Pshf}_L \downarrow Y)$$

Now, Hyland shows the following:

Theorem 4. *For $(Y, f) : \mathbf{R}(X)$, we can restrict $\sum_f$ and $\prod_f$ to functors from $\mathbf{R}(Y)$ to $\mathbf{R}(X)$.*

Proof. This proof proceeds by reducing, step by step, to a known case.

We take $(Z, g) : \mathbf{R}(Y)$. Since functors preserve retraction-section pairs, $\prod_f(Z, g)$ is a retract of $\prod_f \Delta_Y L$. Now, if this is a retract of $\Delta_X L$, then $\prod_f(Z, g)$ is a retract of $\Delta_X L$ as well: We can just compose retractions and sections. Therefore, to show that the image of $\mathbf{R}(Y)$ under $\prod_f$ lies within $\mathbf{R}(X)$, we just need to show that $\prod_f \Delta_Y L$ lies in $\mathbf{R}(X)$. In the same way, for $\sum_f$ we just need to show that $\sum_f \Delta_Y L$ lies in $\mathbf{R}(X)$.

By the definition of $\mathbf{R}(X)$, (Y, f) is a retract:

$$\begin{array}{ccccc} Y & \xrightarrow{s} & X \times L & \xrightarrow{r} & Y \\ & {\scriptstyle f}\searrow & \downarrow{\scriptstyle p_1} & \swarrow{\scriptstyle f} & \\ & & X & & \end{array}$$

The counits of the adjunctions $r_* \vdash \prod_r$ and $s_* \vdash \prod_s$, give maps

$$\prod_f \Delta_Y L \xrightarrow{\prod_f \eta_r} \prod_f \prod_r r^* \Delta_Y L \xrightarrow{\prod_f \prod_r \eta_s} \prod_f \prod_r \prod_s s^* r^* \Delta_Y L$$

Their composite is $\prod_f \eta_{s \cdot r}$. However, note that $r \cdot s = \mathrm{id}_Y$, so we have an isomorphism of functors

$$(s \cdot r)^* \cong \mathrm{id}_{\mathbf{Pshf}_L \downarrow Y} \quad \text{and} \quad \prod_{s \cdot r} \cong \mathrm{id}_{\mathbf{Pshf}_L \downarrow Y}$$

and transporting $\eta_{s \cdot r}$ along these isomorphisms, we get the identity natural transformation $\mathrm{id}_{\mathbf{Pshf}_L \downarrow Y} \Rightarrow \mathrm{id}_{\mathbf{Pshf}_L \downarrow Y}$.

Secondly, note that $r \cdot f = p_1$, so we have $\prod_r \bullet \prod_f \cong \prod_{p_1}$.

Lastly, by Remark 4, $r^* \Delta_Y L \cong \Delta_{X \times L} L$.

Therefore, we have the following diagram, showing that $\prod_f \Delta_Y L$ is a retraction of $\prod_{p_1} \Delta_{X \times L} L$:

$$\begin{array}{ccccccccc} & & & & \prod_{p_1} \Delta_{X \times L} L & & & & \\ & & & & \sim\uparrow & & & & \\ \prod_f \Delta_Y L & \xrightarrow{\prod_f \eta_r} & \prod_f \prod_r r^* \Delta_Y L & \xrightarrow{\prod_f \prod_r \eta_s} & \prod_f \prod_r \prod_s s^* r^* \Delta_Y L & \xrightarrow{\sim} & \prod_f \Delta_Y L & & \\ & \searrow & & \nearrow{\scriptstyle \prod_f \eta_{s \cdot r}} & & & & \searrow & \\ & & & & & {\scriptstyle \mathrm{id}_{\mathbf{Pshf}_L \downarrow X}} & & & \end{array}$$

In a similar way, $\sum_f \Delta_Y L$ is a retract of $\sum_{p_1} \Delta_{X \times L} L$:

$$\begin{array}{ccccccccc} & & & & \sum_{p_1} \Delta_{X \times L} L & & & & \\ & & & & \sim\uparrow & & & & \\ \sum_f \Delta_Y L & \xrightarrow{\sim} & \sum_f \sum_r \sum_s s^* r^* \Delta_Y L & \xrightarrow{\sum_f \sum_r \epsilon_s} & \sum_f \sum_r r^* \Delta_Y L & \xrightarrow{\sum_f \epsilon_r} & \sum_f \Delta_Y L & & \\ & \searrow & & \nearrow{\scriptstyle \sum_f \epsilon_{s \cdot r}} & & & & \searrow & \\ & & & & & {\scriptstyle \mathrm{id}_{\mathbf{Pshf}_L \downarrow X}} & & & \end{array}$$

So again, by the transitivity of retracts, we only have to show that $\prod_{p_1} \Delta_{X \times L} L$ and $\sum_{p_1} \Delta_{X \times L} L$ are retracts of $\Delta_X L$. By Lemma 13, we have $\prod_{p_1} \Delta_{X \times L} L \cong \Delta_X L^L$. By Lemma 11, the functor $\sum_{p_1}$ is given by postcomposition, so $\sum_{p_1} \Delta_{X \times L} L \cong \Delta_X (L \times L)$.

Since Δ_X is a functor, and in particular, preserves retracts, we only have to show that $L \times L$ and L^L are retracts of L.

But we already saw in Theorem 3 that $L^L \cong A(L, 1)$ is a retract of L.

Also, the (pairing and splitting) morphisms $s : \mathbf{Pshf}_L(L \times L, L)$ and $r : \mathbf{Pshf}_L(L, L \times L)$ given by

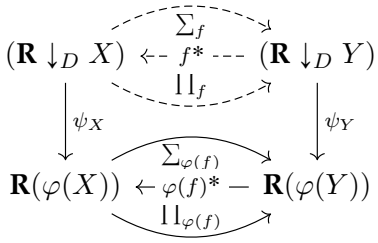
$$s_n(a,b) = \lambda x_{n+1}, x_{n+1}ab \quad \text{and} \quad r_n(a) = (a(\lambda x_{n+1}x_{n+2}, x_{n+1}), a(\lambda x_{n+1}x_{n+2}, x_{n+2}))$$

exhibit $L \times L$ as a retract of L. □

Remark 48. Now, recall that according to Hyland, Taylor endows the category of retracts $\mathbf{R}$ with a class of display maps D, such that $(\mathbf{R} \downarrow_D X)$ consists exactly of the retracts of $\Delta_X U$.

Hyland then claims that the fact that $\mathbf{R}$ is cartesian closed relative to this class of display maps follows from the theorem above. Recall that Taylor already shows relative cartesian closedness directly, which still works in univalent foundations if we assume the axiom of choice.

Recall that we have a weak equivalence $\varphi : \mathbf{R} \xrightarrow{\sim} \hat{\mathbf{R}}$ (Lemma 20) and one can imagine that we can extend this to weak equivalences $\psi_X : (\mathbf{R} \downarrow_D X) \xrightarrow{\sim} (\hat{\mathbf{R}} \downarrow_D \varphi(X))$. Hyland notes that for $X : \hat{\mathbf{R}}$, $\mathbf{R}(\varphi(X))$ is equivalent to $(\hat{\mathbf{R}} \downarrow_D \varphi(X))$. Implicitly, for $f : \mathbf{R}(X, Y)$, he wants to lift the functors $\sum_f$, f^* and $\prod_f$ along the equivalences ψ:



For this to work, we need to turn the weak equivalences ψ_X into adjoint equivalences. In particular, we know that for every object $A : \mathbf{R}(\varphi(X))$, there merely exists a preimage $\bar{A} : (\mathbf{R} \downarrow_D X)$ such that $\psi_X(\bar{A}) \cong A$, and we need to construct a functor $\psi_X^{-1} : \mathbf{R}(\varphi(X)) \to (\mathbf{R} \downarrow_D X)$.

For Taylor's proof, we did a similar thing when we knew that for every object $X : (\mathbf{R} \downarrow_D A)$, there existed some $Y : \mathbf{R}^A$, and we needed to construct a pullback functor $f^* : (\mathbf{R} \downarrow_D A) \to (\mathbf{R} \downarrow_D B)$. We could use the axiom of choice here because the objects of $\mathbf{R}$ (and therefore also the objects of the relative slices $(\mathbf{R} \downarrow_D A)$) form a set.

However, as mentioned in [AKS15] halfway through the introduction, and as explained more thoroughly in [Uni13], Lemma 3.8.5, if $\mathbf{R}(\varphi(X))$ is not a set (and if L is nontrivial, it is never a set), then there is no axiom of choice which can turn the 'for all ..., there exists ...' into a 'there exists a function'.

Therefore, in univalent foundations, the relative cartesian closedness is not a corollary of Theorem 4.

5.4 An elementary proof of the 'Fundamental Theorem'

The final theorem that Hyland works towards in his paper constructs an equivalence between λ-calculi and Λ-algebras. In this section, we will provide an elementary proof of this theorem. Then, in the next section, we will give Hyland's original proof, which uses more high-level categorical reasoning.

We will assume that Λ (and therefore, any λ-theory) satisfies β-equality.

In both cases, one part of the equivalence is very easy. Recall that the pure λ-calculus, the λ-theory Λ, is the initial object of **LamTh**. That means that **LamTh** is equivalent to $(\Lambda \downarrow \mathbf{LamTh})$. Now, using Definition 43, for any n we get a functor

$$-_n : \mathbf{LamTh} \to \mathbf{Alg}_\Lambda.$$

Here we are only interested in the case $n = 0$.

We will use Lemma 31, by showing that $-_0$ is a weak equivalence.

First of all, let A be an algebra for the initial λ-theory Λ.

The Λ-algebra structure gives the terms of A quite a lot of interesting behaviour. For example, we can define 'function application' and composition as

$$ab = (x_1x_2) \bullet (a, b) \quad \text{and} \quad a \circ b = (x_1 \circ x_2) \bullet (a, b),$$

and the same for the other constructions at the start of Section 4.2.

Remark 49. Recall that in Example 6, we constructed an algebraic theory T encapsulating the structure of a monoid. This allowed us to define a monoid operation on T-algebras as well. We then were able to transfer associativity of the operation on the T_n to associativity of the operation on the algebras. In exactly the same way, the function composition on A is associative because composition on Λ_n is associative. Similarly, we can show that $\pi_1(a, b) = a$, that $\pi_2 \circ \langle a, b\rangle = (\lambda x_2, x_1x_2) \bullet b$ etc.

In fact, we can repeat almost the entirety of Section 4.2 for A instead of for L_0:

Definition 48. We define the 'category of retracts' of A as the category $\mathbf{R}_A$ given by

$$(\mathbf{R}_A)_0 = \{X : A \mid X \circ X = X\} \quad \text{and} \quad \mathbf{R}_A(X, Y) = \{f : A \mid Y \circ f \circ X = f\}.$$

Just like in Section 4.2, $\mathbf{R}_A$ has 'universal object' and terminal object

$$U = (\lambda x_1, x_1) \bullet () \quad \text{and} \quad I = (\lambda x_1, c_1),$$

products

$$A \times B = \langle A \circ \pi_1, B \circ \pi_2\rangle$$

and exponential objects

$$B^A = (\lambda x_3, x_1 \circ x_3 \circ x_2) \bullet (B, A)$$

with the isomorphism $\psi : \mathbf{R}_A(C \times A, B) \xrightarrow{\sim} \mathbf{R}_A(C, B^A)$ given by

$$\psi(f) = (\lambda x_2x_3, x_1(x_2, x_3)) \bullet f \quad \text{and} \quad \psi^{-1}(f) = (\lambda x_2, x_1(\pi_1x_2)(\pi_2x_2)) \bullet f.$$

Therefore, we have a λ-theory $E_{\mathbf{R}_A}(U)$.

Remark 50. Note that if $A = L_0$, then the construction of $\mathbf{R}_A$ and $\mathbf{R}$ coincide, so we have an almost trivial equivalence $\mathbf{R}_A \cong \mathbf{R}$, and an isomorphism $E_{\mathbf{R}}(U) \cong E_{\mathbf{R}_A}(U)$.

The following lemma shows that our functor is essentially surjective:

Lemma 46. *We have a Λ-algebra isomorphism $\epsilon_A : E_{\mathbf{R}_A}(U)_0 \xrightarrow{\sim} A$.*

Proof. Take $c_n = (\lambda x_{n+1}, x_{n+1}) : \Lambda_n$. Note that

$$E(U)_0 = \{f : A \mid (\lambda x_2, x_1c_2) \bullet f = f\}.$$

Therefore, we have a bijection given by

$$\epsilon(a) = (x_1c_1) \bullet a \quad \text{and} \quad \epsilon^{-1}(a) = (\lambda x_2, x_1) \bullet a.$$

Now, to show that this is an isomorphism of Λ-algebras, recall that $\bullet : \mathbf{R}_A(U^n, U) \times \mathbf{R}_A(I, U)^n \to \mathbf{R}_A(U)$ is given by precomposition with the product morphism and that the Λ-algebra structure on $E(U)_0$ is given by the embedding of Λ into $E(U)_0$, given by $f \mapsto \iota_{\Lambda_n}(f) \bullet_A ()$ for some

collection $(\iota_{\Lambda n} : \Lambda_n \to \Lambda_0)_n$ (from now on, we will drop the n and just write ι_Λ). We have for all $f : \Lambda_n$, and all $a : E(U)_0^n$,

$$\begin{aligned}\epsilon(f \bullet_{E(U)} a) &= ((\iota_\Lambda(f) \bullet_A ()) \circ \langle a_i \rangle_i)(c_n) \\ &= (\iota_{0,n}(\iota_\Lambda(f))(x_i c_n)_i) \bullet_A a \\ &= (\iota_{0,n}(\iota_\Lambda(f))(x_i)_i) \bullet_A (\epsilon(a_i))_i,\end{aligned}$$

and we want this to equal $f \bullet_A (\epsilon(a_i))_i$. Leaving out the $\iota_{0,n}$, all we need to do is show that for all $f : \Lambda_n$, we have $\iota_\Lambda(f)(x_i)_i = f$. We will do this by structural induction on f.

- If $f = x_{n,i}$, we have $\iota_\Lambda(f) = \pi_{n,i}$, and

$$\iota_\Lambda(f)(x_i)_i = \pi_{n,i}(x_{n,i})_i = x_{n,i}.$$

- If $f = \lambda x_{n+1}, g$ for some $g : \Lambda_{n+1}$, we have

$$\iota_\Lambda(f) = (\lambda x_1 x_2, \iota_{0,2}(\iota_\Lambda(g))(x_1, x_2)),$$

  and if the induction hypothesis holds for g, then

$$\begin{aligned}\iota_\Lambda(f)(x_i)_i &= (\lambda x_{n+1} x_{n+2}, \iota_\Lambda(g)(x_{n+1}, x_{n+2}))(x_i)_i \\ &= \lambda x_{n+1}, \iota_\Lambda(g)((x_i)_i, x_{n+1}) \\ &= \lambda x_{n+1}, g.\end{aligned}$$

- Recall that application in a λ-theory is given by $g_1 g_2 = \rho(g_1) \bullet (x_1, \ldots, x_n, g_2)$. Now, if $f = g_1 g_2$ for $g_1, g_2 : \Lambda_{n+1}$, we have

$$\iota_\Lambda(f) = (\lambda x_1, \iota_\Lambda(g_1)(\pi_1 x_1)(\pi_2 x_1)) \circ \langle \mathrm{id}_{U^n}, \iota_\Lambda(g_2) \rangle = \lambda x_1, \iota_\Lambda(g_1) x_1 (\iota_\Lambda(g_2) x_1),$$

  and if the induction hypothesis holds for g_1 and g_2, then

$$\begin{aligned}\iota_\Lambda(f)(x_i)_i &= (\lambda x_{n+1}, \iota_\Lambda(g_1) x_{n+1} (\iota_\Lambda(g_2) x_{n+1}))(x_i)_i \\ &= \iota_\Lambda(g_1)(x_i)_i (\iota_\Lambda(g_2)(x_i)_i) \\ &= g_1 g_2.\end{aligned}$$

□

Now, take a morphism $F : \mathbf{Alg}_\Lambda(A, B)$. Recall that $\circ$, the categories $\mathbf{R}_A$ and $\mathbf{R}_B$, and their products and exponential objects are given using $f \bullet (a_2, \ldots, a_n)$ for some $f : \Lambda_n$ and $a_1, \ldots, a_n : A$. Since we have $F(f \bullet (a_i)_i) = f \bullet (F(a_i))$, F gives a functor $F : \mathbf{R}_A \to \mathbf{R}_B$, such that

$$F(I) = I, F(U) = U, F(A \times B) = F(A) \times F(B) \quad \text{and} \quad F(A^B) = F(A)^{F(B)},$$

and the same for the product projections and the the natural isomorphisms $\mathbf{R}_A(C, B^A) \cong \mathbf{R}_A(C \times A, B)$. In particular, we have for all $f : \mathbf{R}_A(U^n, U)$,

$$F(f) : \mathbf{R}_B(F(U^n), F(U)) = \mathbf{R}_B(U^n, U).$$

Therefore, F gives a Λ-theory morphism between the endomorphism theories of $U : \mathbf{R}_A$ and $U : \mathbf{R}_B$. This allows us to show:

Lemma 47. *The functor is full.*

Proof. For any λ-theory L, Theorem 1 gives an isomorphism of λ-theories $\eta_L : L \xrightarrow{\sim} E_{\mathbf{R}_{L_0}}(U)$, with

$$\eta_{L0}(f) = \lambda x_1, \iota_{0,1}(f) \quad \text{and} \quad {\eta_L^{-1}}_0(g) = g(\lambda x_1, x_1).$$

Now, for $L, L' : \mathbf{LamTh}$ and $F : \mathbf{Alg}_\Lambda(L_0, L'_0)$, we have

$$\eta_L \cdot F \cdot \eta_{L'}^{-1} : \mathbf{LamTh}(L, L'),$$

and we have for all $s : L_0$,

$$(\eta_{L0} \cdot F \cdot {\eta_{L'}^{-1}}_0)(s) = (\lambda x_1, \iota_{0,1}(F(s)))(\lambda x_1, x_1) = F(s),$$

which shows that the functor is full. □

Now, we only have to show:

Lemma 48. *The functor is faithful.*

Proof. Take morphisms $F, G : \mathbf{LamTh}(L, L')$. Suppose that $F_0 = G_0$. Then, for all $s : L_n$, we have

$$F_n(s) = \rho^n(F_0(\lambda^n(s))) = \rho^n(G_0(\lambda^n(s))) = G_n(s),$$

so $F = G$. □

Summarizing,

Theorem 5. *The functor that sends $L : \mathbf{LamTh}$ to $L_0 : \mathbf{Alg}_\Lambda$, is an adjoint equivalence.*

Proof. By Lemma 47 and Lemma 48, the functor is fully faithful. By Lemma 46, it is essentially surjective, so it is a weak equivalence. Since **LamTh** and $\mathbf{Alg}_\Lambda$ are univalent categories, Lemma 31 shows that F is an adjoint equivalence. □

5.5 Hyland's proof

Hyland gives a more category theoretical proof. That means that there is more high-level intuition why things work the way they work, but on the flip side, there is a lot more details to check.

5.5.1 Terms of a Λ-algebra

Definition 49. Take $\mathbf{1}_n = (\lambda x_1 \dots x_n, x_1 \dots x_n) \bullet () : A$.

Remark 51. Note that we have $\mathbf{1}_1 = U : \mathbf{R}_A$.

Definition 50. In this section, we consider sets of elements of A that behave like functions in n variables:

$$A_n = \{a : A \mid (\lambda x_2 x_3 \dots x_{n+1}, x_1 x_2 x_3 \dots x_{n+1}) \bullet a = a\}.$$

Remark 52. Some straightforward rewriting, shows that

$$A_n = \{a : A \mid \mathbf{1}_n \circ a = a\}.$$

Remark 53. Also note that $\mathbf{1}_n \circ a \circ \mathbf{1}_n = \mathbf{1}_n \circ a$, so for $a : A_n$, $a \circ \mathbf{1}_n = a$.

The following shows that 'functions in n variables' are indeed all in A_n:

Lemma 49. *For $t : \Lambda_{m+n}$ and $a_1, \ldots, a_m : A$, we have $(\lambda^n t) \bullet (a_1, \ldots, a_m) : A$ and we have*

$$\mathbf{1}_n \circ ((\lambda^n t) \bullet (a_1, \ldots, a_m)) = (\lambda^n t) \bullet (a_1, \ldots, a_m),$$

so $(\lambda^n t) \bullet (a_1, \ldots, a_m) : A_n$.

Proof. This follows by straightforward rewriting. □

Corollary 6. *By the previous remark,*

$$((\lambda^n t) \bullet (a_1, \ldots, a_m)) \circ \mathbf{1}_n = (\lambda^n t) \bullet (a_1, \ldots, a_m).$$

Corollary 7. *In particular, $\mathbf{1}_m \circ \mathbf{1}_n = \mathbf{1}_{\max(m,n)}$. From this, it follows that $A_m \subseteq A_n$ for $m \leqslant n$. It also follows that $a \mapsto \mathbf{1}_n \circ a$ gives a function from A to A_n (and also from $A_m \subseteq A$ to A_n).*

Definition 51 (The monoid of a Λ-algebra)**.** We make A_1, the 'functional elements' of A, into a monoid under composition $\circ$ with unit $\mathbf{1}_1$. The fact that this is a monoid follows from the remarks above.

Recall that we have an equivalence $[C_{A_1}^{\mathrm{op}}, \mathbf{Set}] \cong \mathbf{RAct}_{A_1}$.

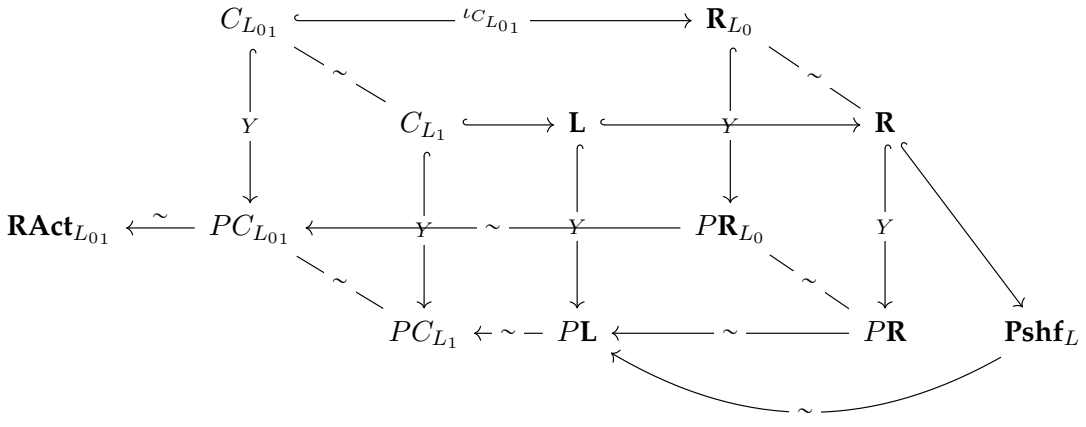
Remark 54. Note that, like in the last chapter, $\mathbf{R}_A$ pops up as the Karoubi envelope of the monoid category C_{A_1}, and fits into the following diagram:

$$\begin{array}{ccccc} & & C_{A_1} & \xhookrightarrow{\iota_{C_{A_1}}} & \mathbf{R}_A \\ & & \downarrow{\scriptstyle Y} & & \downarrow{\scriptstyle Y} \\ \mathbf{RAct}_{A_1} & \xleftarrow[\sim]{} & PC_{A_1} & \xleftarrow[\sim]{} & P\mathbf{R}_A \end{array}$$

Explicitly, this gives the embedding $\mathbf{R}_A(\mathbf{1}_1, -) : \mathbf{R}_A \hookrightarrow \mathbf{RAct}_{A_1}$ given by

$$X \mapsto \mathbf{R}_A(\mathbf{1}_1, X) = \{x : A \mid X \circ x = x\}.$$

Also, note that if $A = L_0$ for some λ-theory L, then the monoid A_1 is equivalent to the monoid L_1, and $\mathbf{R}_A$ is equivalent to our familiar category of retracts $\mathbf{R}$. Using 44, we have the following 2-commutative diagram:



5.5.2 Constructing a theory from an algebra

Definition 52. Composition $\circ$ gives a right A_1-action on the A_n, so we have $A_n : \mathbf{RAct}_{A_1}$. In particular, A_1 acts on itself, and we will call this set with right A_1-action U_A.

Lemma 50. *A_2 is isomorphic to the exponential object $U_A^{U_A}$ in $\mathbf{RAct}_{A_1}$.*

Proof. Recall that $U_A^{U_A}$ is the set of A_1-equivariant morphisms $U_A \times U_A \to U_A$. We have an isomorphism $\psi : A_2 \xrightarrow{\sim} U_A^{U_A}$, given by

$$\psi(f) = (b, b') \mapsto (\lambda x_4, x_1(x_2x_4)(x_3x_4)) \bullet (f, b, b'),$$

treating $f : A_2$ as a function in two variables, and simultaneously composing it with the functions $b, b' : A_1$. It has an inverse

$$\psi^{-1}(g) = (\lambda x_2 x_3, x_1(x_2, x_3)) \bullet (g(\pi_1, \pi_2)).$$

Note that the first pair (x_2, x_3) is a λ-term, whereas the second pair (π_1, π_2) is a pair in $U_A \times U_A$.

For $f : U_A^{U_A}$ and $(a_1, a_2) : U_A \times U_A$, we have

$$\psi(\psi^{-1}(f))(a_1, a_2) = f(\pi_1, \pi_2) \circ \langle a_1, a_2 \rangle = f(\pi_1 \circ \langle a_1, a_2 \rangle, \pi_2 \circ \langle a_1, a_2 \rangle) = f(a_1, a_2).$$

Here we use the A_1-equivariance of f. In the last step of this proof, we use, among other things, the fact that the $a_i : A_1$ and therefore $\lambda x_1, a_i x_1 = a_i$.

Some straightforward rewriting shows that for $a : A_2$, we have $\psi^{-1}(\psi(a)) = a$. In the last step of this proof, we use the fact that $a : A_2$ and therefore $\lambda x_1 x_2, a x_1 x_2 = a$.

Therefore, ψ is a bijection and, as it turns out, an isomorphism. □

Remark 55. Recall that the embedding $\mathbf{R}_A \hookrightarrow \mathbf{RAct}_{A_1}$ is the composition of a Yoneda embedding and two adjoint equivalences. These all preserve exponential objects (**(TODO)** : Citation needed). Now, note that A_1 is the image of $U : \mathbf{R}_A$, so the exponential $U_A^{U_A}$ is A_2, the image of $U^U = \lambda x_1 x_2, x_1 x_2 = \mathbf{1}_2$.

Definition 53 (Construction of the λ-theory)**.** Recall that $A_2 \subseteq U_A$, and that we have a retraction $a \mapsto \mathbf{1}_2 \circ a : U_A \to A_2$.

Therefore, U_A is a reflexive object in $\mathbf{RAct}_{A_1}$, and we get a λ-theory $E(U_A)$.

Remark 56. The cartesian closed embedding of $\mathbf{R}_A$ into $\mathbf{RAct}_{A_1}$ sends U to U_A. It sends the retraction $U^U : \mathbf{R}_A(U, U^U)$ exactly to the retraction $a \mapsto \mathbf{1}_2 \circ a : \mathbf{RAct}_{A_1}(U_A, A_2)$. Therefore, we have a λ-theory isomorphism $E_{\mathbf{R}_A}(U) \cong E_{\mathbf{RAct}_{A_1}}(U_A)$.

5.5.3 Constructing a theory morphism from an algebra morphism

Take a morphism $F : \mathbf{Alg}_\Lambda(A, B)$. Note that F preserves $\circ$ and the $\mathbf{1}_n$. Therefore, it induces a monoid morphism $A_1 \to B_1$, which gives a functor $C_{A_1} \to C_{B_1}$. Precomposing with this, we get a functor $\mathbf{RAct}_{B_1} \to \mathbf{RAct}_{A_1}$.

To get a morphism $E_{\mathbf{RAct}_{A_1}}(U_A) \to E_{\mathbf{RAct}_{B_1}}(U_B)$ however, we need a functor $\mathbf{RAct}_{A_1} \to \mathbf{RAct}_{B_1}$, which we obtain by Left Kan extension: see Lemma 26 (the "tensor product").

Hyland gives a very high-level argument why F_* induces a morphism $\bar{F} : \mathbf{LamTh}(E_{\mathbf{RAct}_{A_1}}(U_A), E_{\mathbf{RAct}_{B_1}}(U_B$ which we will discuss a bit more here.

Lemma 51. *The extension of scalars F_* sends U_A to U_A and preserves finite products.*

Proof. By Lemma 27, we have $F_*(U_A) \cong U_A$.

We will show that F_* preserves binary products and the terminal object, because from this it follows that F_* preserves all finite products.

We use Lemma 28 to show that F_* preserves the terminal object. We take (very similar to the terminal object in $\mathbf{R}$):

$$a_0 = (\lambda x_1 x_2, x_2) \bullet () : A_1 \quad \text{and} \quad b_0 = (\lambda x_1 x_2, x_2) \bullet () : B_1$$

and a_0 is weakly terminal because for all $a : B_1$, we have $F(a_0) \circ a = b_0$.

We use Lemma 29 to show that F_* also preserves the product. Therefore, given $a_1, a_2 : B_1$. Take $b = \langle a_1, a_2 \rangle$, together with the familiar projections $\pi_i : A_1$. We have $a_i = F(\pi_i) \circ a$.

Now, for some $b' : B_1$ and $\pi'_1, \pi'_2 : A_1$ such that $a_i = F(\pi'_i) \circ b'$, take $m = \langle \pi'_1, \pi'_2 \rangle$. Then $\pi_i \circ m = \pi'_i$ and $F(m) \circ a' = a$, so (a, π_1, π_2) is weakly terminal and F_* preserves binary products.

Since any finite product is (isomorphic to) a construction with a repeated binary product and the terminal object, the fact that F_* preserves binary products and the terminal object shows that F_* preserves all finite products. □

Now, we can start defining our lift of F:

Definition 54. We can send an element $g : E(U_A)_n = \mathbf{RAct}_A(U_A^n, U_A)$ to

$$F_*(g) : \mathbf{RAct}_{A'}(f(U_A^n), f(U_A)) \cong \mathbf{RAct}_{A'}((U_{A'})^n, U_{A'}) = E(U_{A'})_n$$

so we have a morphism $\bar{F} : \mathbf{LamTh}(E(U_A), E(U_{A'}))$.

Remark 57. The fact that $\bar{F}$ preserves the variables and substitution is not very hard, since these are just defined in terms of finite products of U_A and $\bar{F}$ preserves finite products and U_A.

However, showing that it is a λ-theory morphism is a different matter. Hyland claims

"F preserves $\mathbf{1}_n$, which determines the function space as a retract of the universal. So $\bar{F}$ preserves the retract and the result follows."

Although this covers the core of the argument, it is very complicated to actually verify that this works, because we need to pass through a lot of isomorphisms, and check that they work nicely together:

$$\begin{aligned} \alpha_A : \mathbf{RAct}_{A_1}(X \times Y, Z) &\xrightarrow{\sim} \mathbf{RAct}_{A_1}(X, Z^Y); \\ \beta : F_*(U_A) &\xrightarrow{\sim} U_{A'}; \\ \bar{\gamma} : F_*(A \times B) &\xrightarrow{\sim} F_*(A) \times F_*(B); \\ \gamma_n : F_*(X^n) &\xrightarrow{\sim} F_*(X)^n; \\ \delta_A : U_A^{U_A} &\xrightarrow{\sim} A_2. \end{aligned}$$

Since we already motivated this in a different way, we will leave it as an exercise for the enthousiastic reader.

Lemma 52. *For $F = id_A$, we have $\bar{F} = id_{E(U_A)}$.*

Proof. By the reasoning set forth in Lemma 48, we only need to check $\bar{F}_0$. Given $s : \mathbf{RAct}_{A_1}(I, U_A)$, we have

$$\bar{F}_0(s) = \gamma^{-1} \cdot F_*(s) \cdot \beta : \mathbf{RAct}_{A_1}(I, U_A)$$

for $\gamma : F_*(I) \cong I$ and $\beta : F_*(U_A) \cong U_A$. Then

$$\bar{\mathrm{id}}_A(s)(\star) = s(\star) \circ ((\lambda x_1 x_2, x_2) \bullet ()) = s(\star \circ ((\lambda x_1 x_2, x_2) \bullet ())),$$

so $\bar{\mathrm{id}}_{A0}(s) = s$ and we can conclude that $\bar{\mathrm{id}}_A = \mathrm{id}_{E(U_A)}$. □

Lemma 53. *For $F : \mathbf{Alg}_\Lambda(A, B)$ and $G : \mathbf{Alg}_\Lambda(B, C)$, we have $\bar{F} \cdot \bar{G} = \overline{F \cdot G}$.*

Proof. Again, we only have to check at the terms without free variables. Given $s : \mathbf{RAct}_{A_1}(I, U_A)$, we have

$$(\bar{F} \cdot \bar{G})(s) = \gamma_G^{-1} \cdot G_*(\gamma_F^{-1} \cdot F_*(s) \cdot \beta_F) \cdot \beta_G : \mathbf{RAct}_{C_1}(I, U_C)$$

for $\gamma : F_*(I) \cong I$ and $\beta : F_*(U_A) \cong U_A$. Then

$$\begin{aligned}(\bar{F} \cdot \bar{G})(s)(\star) &= G(F(s(\star)) \circ ((\lambda x_1 x_2, x_2) \bullet_B ())) \circ ((\lambda x_1 x_2, x_2) \bullet_C ()) \\ &= G(F(s(\star \circ ((\lambda x_1 x_2, x_2) \bullet_A ()) \circ ((\lambda x_1 x_2, x_2) \bullet_A ())))) \\ &= G(F(s(\star))) \\ &= (F \cdot G)^{-1}(s(\star \circ ((\lambda x_1 x_2, x_2) \bullet_A ()))) \\ &= \overline{(F \cdot G)}(s)(\star)\end{aligned}$$

so $(\bar{F} \cdot \bar{G})_0(s) = (\overline{F \cdot G})_0(s)$ and we can conclude that $\bar{F} \cdot \bar{G} = \overline{F \cdot G}$. □

Definition 55. We get a functor from $\mathbf{Alg}_\Lambda$ to **LamTh**, sending objects A to $E_{\mathbf{RAct}_{A_1}}(U_A)$ and morphisms $F : \mathbf{Alg}_\Lambda(A, B)$ to $\bar{F} : E(U_A) \to E(U_B)$.

Remark 58. Note that for $X : \mathbf{R}_A$, we have a natural isomorphism of sets with a right B_1-action

$$\psi : \{f : B \mid F(X) \circ f = f\} \xrightarrow{\sim} \{f : A \mid X \circ f = f\} \times B_1/\sim$$

given by

$$\psi(f) = (X, f) \quad \text{and} \quad \psi^{-1}(f, b) = F(f) \circ b.$$

Therefore, the following diagram 2-commutes:

$$\begin{array}{ccc} \mathbf{R}_A & \xrightarrow{F} & \mathbf{R}_B \\ \downarrow{\scriptstyle \mathbf{R}_A(\mathbf{1}_1, -)} & & \downarrow{\scriptstyle \mathbf{R}_B(\mathbf{1}_1, -)} \\ \mathbf{RAct}_{A_1} & \xrightarrow{F_*} & \mathbf{RAct}_{B_1} \end{array}$$

Since F preserves all the structure of $\mathbf{R}_A$, as do the embeddings of $\mathbf{R}_A$ and $\mathbf{R}_B$ into $\mathbf{RAct}_{A_1}$ and $\mathbf{RAct}_{B_1}$, one would expect F_* to preserves the structure of the full subcategory $\mathbf{R}_A$ of $\mathbf{RAct}_{A_1}$, including U_A, its finite products and their exponentials. This is another way to argue that $\bar{F}$ is indeed a morphism of λ-theories.

5.5.4 The unit

Definining the unit of the adjunction boils down to a version of Scott's representation theorem.

Lemma 54. *For a λ-theory L, we can define a λ-theory isomorphism*

$$\eta_L : L \xrightarrow{\sim} E(U_{L_0}).$$

Proof. Recall from Remark 54 that we have a chain of equivalences

$$\mathbf{Pshf}_L \xrightarrow{\sim} P\mathbf{L} \xrightarrow{\sim} PC_{L_1} \xrightarrow{\sim} PC_{L_{01}} \xrightarrow{\sim} \mathbf{RAct}_{L_{01}}.$$

It sends the presheaf L^n to the set L_1^n with a right action sending $s : L_1^n$ and $t : (L_0)_1$ to $(s_i \bullet_L \rho(t))_i : L_1^n$. We have an isomorphism $s \mapsto (\lambda(s_i))_i : (L_1)^n \to U_{L_0}$, with inverse $(\rho_i)_i : s \mapsto (\rho(s_i))_i$.

Then η_L arises by combining this with the isomorphism from Scott's representation theorem (Theorem 3)

$$\eta_L : L \xrightarrow{\sim} E_{\mathbf{Pshf}_L}(L) \xrightarrow{\sim} E_{\mathbf{RAct}_{L_{01}}}(U_{L_0}).$$

It is quite easy to work out that we can make this explicit as (dropping the L and just writing η_L)

$$\eta_n : L_n \xrightarrow{f \mapsto ((s:\mathbf{Pshf}_{L_m^n}) \mapsto f \bullet s)_m} \mathbf{Pshf}_L(L^n, L) \xrightarrow{f \mapsto f_1} \mathbf{RAct}_{(L_0)_1}(L_1^n, L_1), \xrightarrow{\lambda \circ - \circ (\rho_i)_i} \mathbf{RAct}_{(L_0)_1}(U_{L_0}^n, U_{L_0}),$$

so $\eta_n(f)(s) = \lambda(f \bullet (\rho(s_i))_i)$ for $f : L_n$ and $s : U_{L_0}^n$. □

Remark 59. Note that we can do the same with Scott's version of his representation theorem, using the chain of equivalences and an embedding

$$\mathbf{R} \xrightarrow{\sim} \mathbf{R}_{L_0} \xrightarrow{Y} P\mathbf{R}_{L_0} \xrightarrow{\sim} PC_{L_{01}} \xrightarrow{\sim} \mathbf{RAct}_{L_{01}},$$

Here, it is a bit harder to get an explicit formula for η, because we need to do a bit more conversion between U_A^n and the image of U^n after the embedding $\varphi : \mathbb{R} \hookrightarrow \mathbf{RAct}_{L_{01}}$. If we quickly define λ-terms

$$\langle x_i \rangle_i = \langle x_1, \dots, x_n \rangle = \langle \langle \dots \langle (\lambda x_{n+1} x_{n+2}, x_{n+2}), x_1 \rangle, \dots \rangle, x_n \rangle$$

and corresponding tuples $(x_i)_i$ and projections π_i, we have

$$L_n \xrightarrow{f \mapsto \lambda(f \bullet (\pi_i x_1)_i)} \mathbb{R}(U^n, U) \xrightarrow{f \mapsto f \circ -} \mathbf{RAct}_{L_{01}}(\varphi(U^n), U_{L_0}) \xrightarrow{(s \mapsto \langle s_i \rangle_i) \circ -} \mathbf{RAct}_{L_{01}}(U_{L_0}^n, U_{L_0}).$$

and in the end, we obtain the very same explicit formula

$$\eta_n(f)(s) = \iota_{0,n}(f)(x_i)_i \circ (\langle s_i \rangle_i) = \lambda(f \bullet (\rho(s_i))_i).$$

Lemma 55. *η is natural in L. That is, for all $F : \mathbf{LamTh}(L, L')$, the following diagram commutes:*

$$\begin{array}{ccc} L & \xrightarrow{\quad F \quad} & L' \\ \downarrow{\scriptstyle \eta_L} & & \downarrow{\scriptstyle \eta_{L'}} \\ E_{\mathbf{RAct}_{L_{01}}}(U_{L_0}) & \xrightarrow{\bar{F}_0} & E_{\mathbf{RAct}_{L'_{01}}}(U_{L'_0}) \end{array}$$

Proof. We must show

$$\eta_L \cdot \bar{F}_0 = F \cdot \eta_{L'}.$$

Note that for all $s : L_0$,

$$\begin{aligned} \bar{F}_{00}(\eta_L(s)) &= \gamma_{F_0}^{-1} \cdot F_{0*}(\star \mapsto (\lambda x_1, \iota_{0,1}(s))) \cdot \beta_F \\ &= \star \mapsto F_0(\lambda x_1, \iota_{0,1}(s)) \circ ((\lambda x_1 x_2, x_2) \bullet ()) \\ &= \star \mapsto (\lambda x_1, \iota_{0,1}(F_0(s))) \\ &= \eta_{L'}(F_0(s)) \end{aligned}$$

and by Lemma 47, this concludes the proof. □

5.5.5 The counit

Definition 56. For $A : \mathbf{Alg}_\Lambda$, we define a bijection $\epsilon_A : E_{\mathbf{RAct}_{A_1}}(U_A)_0 \cong A$ as

$$\epsilon_A(s) = x_1(\lambda x_2, x_2) \bullet (s(\star)) \quad \text{and} \quad \epsilon_A^{-1}(a)(\star) = (\lambda x_2, x_1) \bullet a.$$

These are inverses because $s(\star)$ is A_1-equivariant (Lemma 23), and then

$$(\lambda x_2, x_1(\lambda x_3, x_3)) \bullet s(\star) = s(\star) \circ ((\lambda x_1 x_2, x_2) \bullet ()) = s(\star).$$

We want to show that ϵ_A is an isomorphism of Λ-algebras. We use Lemma 38 for this, so we need to show that it preserves the application and the Λ-definable constants.

Lemma 56. *We have for all $a, b : E(U_A)_0$,*

$$\epsilon_A((x_1 x_2) \bullet (a, b)) = (x_1 x_2) \bullet (\epsilon_A(a), \epsilon_A(b)).$$

Proof. For $a, b : E(U_A)_0$, we have, using at some point the isomorphism $\delta : \mathbf{RAct}_{A_1}(U_A^{U_A}, A_2)$,

$$\begin{aligned}\epsilon_A((x_1x_2) \bullet (a, b)) &= (x_1(\lambda x_3, x_3)(x_2(\lambda x_3, x_3))) \bullet (a(\star), b(\star)) \\ &= (x_1x_2) \bullet (\epsilon_A(a), \epsilon_A(b))\end{aligned}$$

and this concludes the proof. □

To show that ϵ preserves the Λ-definable constants, we first need to show two properties of ϵ and η:

Lemma 57. *ϵ is natural in A. That is, for all $F : \mathbf{Alg}_\Lambda(A, B)$, the following diagram commutes:*

$$\begin{array}{ccc} E(U_A)_0 & \xrightarrow{\bar{F}_0} & E(U_B)_0 \\ \downarrow{\scriptstyle \epsilon_A} & & \downarrow{\scriptstyle \epsilon_B} \\ A & \xrightarrow{F} & B \end{array}$$

Proof. The functor $F_* : \mathbf{RAct}_{A_1} \to \mathbf{RAct}_{B_1}$ sends $X : \mathbf{RAct}_{A_1}$ to $X \times B_1/\sim : \mathbf{RAct}_{B_1}$. We have isomorphisms

$$\beta : F_*(U_A) \xrightarrow{\sim} U_B \quad \text{and} \quad \gamma : F_*(I) \xrightarrow{\sim} I,$$

with

$$\beta(a, b) = F(a) \circ b \quad \text{and} \quad \gamma^{-1}(\star) = (\star, (\lambda x_1 x_2, x_2) \bullet ()).$$

Then $\bar{F} : \mathbf{RAct}_{A_1}(I, U_A) \to \mathbf{RAct}_{B_1}(I, U_B)$ is given by

$$\bar{F}(s)(\star) = \gamma^{-1} \cdot (s \times \mathrm{id}_{B_1}) \cdot \beta = (\lambda x_2, x_1(\lambda x_3, x_3)) \bullet F(s(\star)),$$

so

$$(\bar{F} \cdot \epsilon_B)(s) = (x_1(\lambda x_2, x_2)) \bullet F(s(\star)) = (\epsilon_A \cdot F)(s),$$

which concludes the proof. □

Lemma 58. *ϵ and η satisfy one of the zigzag identities.*

Proof. In this case, the zigzag identity on $L \mapsto L_0$ boils down to the following diagram commuting for all L : **LamTh**:

$$\begin{array}{ccc} L_0 & \xrightarrow{\mathrm{id}_{L_0}} & L \\ & {\scriptstyle \eta_{L_0}} \searrow \quad \nearrow {\scriptstyle \epsilon_{L_0}} & \\ & E(U_{L_0})_0 & \end{array}$$

Now, note that for all $f : L_0$,

$$(\eta_{L0} \cdot \epsilon_{L_0})(f) = (x_1(\lambda x_2, x_2)) \bullet (\lambda x_1, \iota_{0,1}(f)) = f,$$

which shows that the diagram commutes. □

Now, finally, we are ready to show that ϵ is an isomorphism of Λ-algebras:

Lemma 59. *We have for all $s : \Lambda_0$,*

$$\epsilon_A(s \bullet ()) = s \bullet ().$$

Proof. Consider the following diagram, with $F : \mathbf{Alg}_\Lambda(\Lambda_0, A)$ given by $F(s) = s \bullet ()$:

$$\begin{array}{ccc} E(U_{\Lambda_0})_0 & \xrightarrow{\bar{F}_0} & E(U_A)_0 \\ {\scriptstyle \eta_0}\circlearrowleft\downarrow{\scriptstyle \epsilon_{\Lambda_0}} & & \downarrow{\scriptstyle \epsilon_A} \\ \Lambda_0 & \xrightarrow{F} & A \end{array}$$

By Lemma 57, the square commutes and by Lemma 58, we have $\eta_0 \cdot \epsilon_{\Lambda_0} = \mathrm{id}_{\Lambda_0}$.

Recall that there exists a unique morphism $\iota_\Lambda : \mathbf{LamTh}(\Lambda, E(U_A))$, and that for all $s : \Lambda_0$, by definition $s \bullet_{E(U_A)_0} () = \iota_\Lambda(s) \bullet_{E(U_A)} () = \iota_\Lambda(s)$. Since we have $\eta \cdot \bar{F} : \mathbf{LamTh}(\Lambda, E(U_A))$, we must have $\iota_\Lambda = \eta \cdot \bar{F}$ and

$$\begin{aligned} \epsilon_A(s \bullet_{E(U_A)} ()) &= \epsilon_A(\bar{F}_0(\eta_0(s))) \\ &= F(\epsilon_{\Lambda_0}(\eta_0(s))) \\ &= F(s) \\ &= s \bullet_A (), \end{aligned}$$

which concludes the proof. □

5.5.6 The equivalence

By Lemma 3.2 in [nLa24a] and Lemma 58, η and ϵ satisfy both zigazg identities, and we can state the fundamental theorem of the λ-calculus:

Theorem 6. *There is an adjoint equivalence* $\mathbf{LamTh} \cong \mathbf{Alg}_\Lambda$*, sending a λ-theory L to the Λ-algebra L_0, with an inverse functor that sends a Λ-algebra A to the theory $E_{\mathbf{RAct}_{A_1}}(U_A)$.*

Remark 60. Hyland remarks that the isomorphism $U_A^{U_A} \cong A_2$ can be generalized to isomorphisms $U_A^{U_A^n} \cong A_{n+1}$.

$$\begin{aligned} \mathbf{RAct}_{A_1}(U_A^n, U_A) &\cong \mathbf{RAct}_{A_1}(I, U_A^{U_A^n}) \\ &\cong \mathbf{RAct}_{A_1}(I, A_{n+1}) \\ &\cong \{f : A_{n+1} \mid \forall a : A_1, f \circ a = f\} \\ &\cong A_n. \end{aligned}$$

Explicitly, we get an isomorphism $\psi : \mathbf{RAct}_{A_1}(U_A^n, U_A) \xrightarrow{\sim} A_n$, given by

$$\psi(f) = (\lambda x_2 \dots x_{n+1}, x_1(x_2, \dots, x_{n+1})) \bullet (f(\pi_1, \dots, \pi_n))$$

and

$$\psi^{-1}(g)(a_1, \dots, a_n) = (\lambda x_{n+2}, x_1(x_2 x_{n+2}) \dots (x_{n+1} x_{n+2})) \bullet (g, a_1, \dots, a_n).$$

Note that to show this, we need to use the A_1-equivariance of f at some point:

$$\begin{aligned} &f(\pi_2, \dots, \pi_{n+1})((\lambda x_{n+1}, x_{n+1}), x_1, \dots, x_n) \\ &= (f(\pi_1, \dots, \pi_n) \circ \langle \pi_2, \dots, \pi_{n+1} \rangle)((\lambda x_{n+2}, x_{n+2}), x_1, \dots, x_n) \\ &= f(\pi_1, \dots, \pi_n)(x_1, \dots, x_n). \end{aligned}$$

This gives a λ-theory structure on $(A_n)_n$, with

$$\begin{aligned} y_{n,i} &= (\lambda x_1 \dots x_n, x_{n,i}) \bullet (); \\ f \bullet g &= (\lambda x_{m+2} \dots x_{m+n+1}, x_1(x_2 x_{m+2} \dots x_{m+n+1}) \dots (x_{m+1} x_{m+2} \dots x_{m+n+1})) \bullet (f, g_1, \dots, g_m); \\ \rho(f) &= \mathbf{1}_m \circ f; \\ \lambda(h) &= f. \end{aligned}$$

for $f : A_m$, $g : A_n^m$ and $h : A_{m+1}$.

Then any $F : \mathbf{Alg}_\Lambda(A, B)$ gives functions $F : A_n \to B_n$, and these give a λ-theory morphism in $\mathbf{LamTh}((A_n)_n, (B_n)_n)$.

The natural isomorphism $\epsilon_A : A_0 \to A$ is just id_A. Note that, even though $A_0 = A$ as sets, their Λ-algebra structures are defined differently, so it takes some work to show that ϵ_A is a Λ-algebra morphism. Also, $\eta_{Ln} : L_n \to (L_0)_n$ is given by λ^n, with ρ^n as its inverse. The zigzag identities are trivial, so this gives another, very elemental proof of the fundamental theorem.

5.6 Theory of extensions

The fundamental theorem of the λ-calculus that Hyland shows is actually not of the form shown above. To get there, we first need to show that the category of T-algebras for an algebraic theory T has coproducts, and define the 'theory of extensions'.

Let $[\![n]\!]$ denote the finite set $\{1, 2, \ldots, n\}$. For T an algebraic theory, let $\mathbf{L}$ be its corresponding Lawvere theory (Lemma 70).

(**TODO**) Move the coproducts and theory of extensions to chapter 3?

Lemma 60. *Let T be an algebraic theory. The category of T-algebras has coproducts.*

Proof. This is shown in [ARV10], in the lemmas leading up to Theorem 4.5.

Explicitly, we can express the coproduct of algebras, and especially its set, as the following coend ([HYL17], Proposition 2.5) (see also Section 1.10 for more on coends)

$$A + B = \int^{(m,n):\mathbf{L}\times\mathbf{L}} T_{m'+n'} \times A^m \times B^n,$$

considering A as a covariant functor on $\mathbf{L}$ (see Lemma 71) and the theory presheaf T as a presheaf (see Lemma 72).

Note that we do not need the exact definition of $A+B$ for the rest of this section. Nonetheless, it is interesting to see how it is defined and why this definition works.

One can think of $A + B$ consisting of elements $t \bullet (a + b)$ for $t : T_{m+n}$, $a : A^m$ and $b : B^n$ (writing $(a+b)$ for $(a_1, \ldots, a_m, b_1, \ldots, b_n)$), 'substituting' the a_i and b_j for the x_i and x_{m+j} in t.

However, the coend is a quotient of $\coprod T_{m+n} \times A^m \times B^n$ along some relations. These relations then give 'associativity' of this substitution $t \bullet (a + b)$. In particular, they assure that reordering or duplicating x_i and their corresponding a_i and b_j do not yield different elements. For $f : \mathbf{L}(m, m') = T_m^{m'}$, $g : \mathbf{L}(n, n') = T_n^{n'}$, associating the images on the left and right of

$$T_{m'+n'} \times A^{m'} \times B^{n'} \leftarrow T_{m'+n'} \times A^m \times B^n \to T_{m+n} \times A^m \times B^n$$

gives

$$t \bullet ((f_i \bullet a)_i + (g_j \bullet b)_j) = (t \bullet ((f_i \bullet (x_{m+n,j})_j)_i + (g_i \bullet (x_{m+n,m+j})_j)_i)) \bullet (a + b)$$

for $t : T_{m'+n'}$, $a : A^m$ and $b : B^n$.

Then it becomes clear what the action of T will be on $A+B$, although the precise definition looks a bit complex because we have to juggle a bit with the variables in the different T_{m+n}. For $s : T_l$, $t_i : T_{m_i+n_i}$, $a_i : A^{m_i}$ and $b_i : B^{n_i}$, define the disjoint embeddings

$$d_i : [\![m_i + n_i]\!] \cong [\![m_i]\!] \sqcup [\![n_i]\!] \hookrightarrow \left[\!\!\left[\sum_j m_j\right]\!\!\right] \sqcup \left[\!\!\left[\sum_j n_j\right]\!\!\right] \cong \left[\!\!\left[\sum_j m_j + \sum_j n_j\right]\!\!\right],$$

which we will use to make sure that the x_j in the different t_i are mapped to distinct variables. Then we can define

$$s \bullet (t_i \bullet a_i + b_i) = (s \bullet (t_i \bullet (x_{d_i(j)})_j)_i) \bullet (a_1 + b_1 + \cdots + a_l + b_l).$$

More formally (using the coend injections $A^{\sum_k m_k} \times B^{\sum_k n_k} \times T_{\sum_k m_k + \sum_k n_k} \to A + B$), this gives functions

$$T_l \to (A^{m_1} \times B^{n_1} \times T_{m_1+n_1} \to (\cdots \to (A^{m_l} \times B^{n_l} \times T_{m_l+n_l} \to A + B) \dots)),$$

commuting with the relations between the different $(A^m \times B^n \times T_{m+n})$, which, by repeatedly using the universal property of the coend, then correspond to functions

$$T_l \to (A + B \to (\cdots \to (A + B \to A + B) \dots)),$$

or, equivalently, a function

$$T_l \times (A + B)^l \to A + B.$$

We have left and right injections $A \to A + B$ and $B \to A + B$, given respectively by the maps $A^1 \times B^0 \times T_{1+0} \to A + B$ and $A^0 \times B^1 \times T_{0+1} \to A + B$:

$$a \mapsto x_1 \bullet a \quad \text{and} \quad b \mapsto x_1 \bullet b$$

and every element $A + B$ arises by the action of t on combinations of these embedded elements:

$$t \bullet (a + b) = t \bullet ((x_1 \bullet a_i)_i + (x_1 \bullet b_j)_j),$$

which ultimately can be used to show that $A + B$ indeed has the universal property of the coproduct. □

Definition 57 (Theory of extensions)**.** Let T be an algebraic theory and A a T-algebra. We can define an algebraic theory T_A called the *theory of extensions* of A' with $(T_A)_n = A + T_n$. The right injection of the variables $x_i : T_n$ gives the variables.

For $h : (A + T_n)^m$, sending $g : T_m$ to $g \bullet h$ gives a T-algebra morphism $T_m \to T_n + A$. Together with the right injection morphism of A into $T_n + A$, this gives us a T-algebra morphism from the coproduct: $T_m + A \to T_n + A$. Doing this for every $h : (A + T_n)^m$ gives us the substitution $(T_m + A) \times (T_n + A)^m \to T_n + A$.

Showing that this is indeed an algebraic theory involves invoking the universal property of the coproduct and using properties of T-algebras and T-algebra morphisms.

Remark 61. We can turn the map $A \mapsto T_A$ into a functor $T_- : \mathbf{Alg}_T \to \mathbf{AlgTh}$. For a morphism $f : \mathbf{Alg}_T(A, B)$, we get maps

$$f + \mathrm{id}_{T_n} : \mathbf{Alg}_T(A + T_n, B + T_n).$$

We can combine these into a morphism $T_f = (f + \mathrm{id}_{T_n})_n$, and this makes T_- into a functor from $\mathbf{Alg}_T$ to $\mathbf{AlgTh}$.

To actually show that T_f is a morphism and that T_- is a functor, we use the properties of the coproduct a couple of times, as well as the fact that $f + \mathrm{id}_{T_n} : \mathbf{Alg}_T(A + T_n, B + T_n)$ preserves the T-action.

Remark 62. Note that for a T-algebra morphism $f : A \to B$, we have morphisms $f : A + T_n \xrightarrow{f+\mathrm{id}_{T_n}} B + T_n$.

Remark 63. Note that the right embeddings $r_n : T_n \to A + T_n$ give an algebraic theory morphism $(r_n)_n : T \to T_A$, so we can think of T as lying inside T_A.

The following result explains why we are interested in the theory of extensions:

Lemma 61. *For T an algebraic theory and A a T-algebra, we have an adjoint equivalence $\mathbf{Alg}_{T_A} \cong (A \downarrow \mathbf{Alg}_T)$ between algebras for T_A and the coslice category under A.*

Proof. Let B be a T_A-algebra. Pullback along the embedding $(r_n)_n : T \to T_A$ gives $(r_n)^*_n(B) : \mathbf{Alg}_T$. Also, we have a T-algebra morphism $A \to B$ given by the composition

$$A \to (T_A)_0 \xrightarrow{f \mapsto f \bullet ()} B.$$

Conversely, take $f : \mathbf{Alg}_T(A, B)$. For $b : B^n$, we have a T-algebra morphism $T_n \to B$ given by $f \mapsto f \bullet b$. This, together with f, gives a morphism from the coproduct $A + T_n \to B$, and doing this for every $b : B^n$ gives a T_A-action on B as functions $(A + T_n) \times B^n \to B$.

Now, showing that the function $A \to (T_A)_0 \to B$ defined above is indeed a T-algebra morphism and that the other B, together with the given T_A-action is indeed a T_A-algebra, and furthermore showing that these extend to functors that together form an adjoint equivalence, involves checking a lot of details. One can indeed check that all of this holds, using the properties of algebraic theories, algebras, algebra morphisms and coproducts, as well as the fact that for all $b : B^n$, $f \mapsto f \bullet b$ is a morphism in $\mathbf{Alg}_T(A + T_n, (r_n)^*_n(B))$. However, for the sake of brevity, we will omit these and point to the formalization for the details. □

Example 12. Take $T_n = Z[X_1, \ldots, X_n]$, the polynomial rings in n variables. T-algebras correspond to commutative rings, so we can call T 'the theory of commutative rings'. Now, for a commutative ring R, $\mathbf{Alg}_{T_R}$ is equivalent to the coslice category $(R \downarrow \mathbf{Alg}_T)$, which is the category of R-algebras. Therefore, the theory of extensions T_R can be considered to be the theory of R-algebras.

Lemma 62. *Any algebraic theory morphism $f : \mathbf{AlgTh}(S, T)$ factorizes through the embedding of S into the theory of extensions of the pullback of the algebra T_0:*

$$S \longrightarrow S_{f^*(T_0)} \longrightarrow T, \qquad S \xrightarrow{\quad f \quad} T$$

Proof. For any n, we have a map of S-algebras $[t \mapsto t \bullet (), f_n] : f^*(T_0) + S_n \to f^*(T_n)$, given on $f^*(T_0)$ by $t \mapsto t \bullet ()$ and on S_n by f_n. By the universal property of the coproduct, the following diagram commutes for all n:

$$S_n \xrightarrow{r_n} f^*(T_0) + S_n \xrightarrow{[t \mapsto t \bullet (), f_n]} T, \qquad S_n \xrightarrow{\quad f_n \quad} T$$

which shows that $(r_n)_n \cdot [f \mapsto f \bullet (), f_n]_n = f$.

Now, to show that $[f \mapsto f \bullet (), f_n]_n$ indeed constitutes an algebraic theory morphism is a bit more work. It involves using the universal property of the coproduct a couple of times, as well as showing that for all $g : S^n_{f^*(T_0)}$,

$$s \mapsto ([t \mapsto t \bullet (), f_m]x) \bullet_T ([t \mapsto t \bullet (), f_n]g_i)_i$$

is a morphism in $\mathbf{Alg}_S(f^*(T_0) + T_m, f^*(T_n))$. For more details, we again point to the formalization. □

Now, given a Λ-algebra A, applying the above to the initial morphism $\iota_\Lambda : \Lambda \to E_{\mathbf{R}_A}(U)$, we get the following diagram:

$$\begin{array}{ccccc} \Lambda & \longrightarrow & \Lambda_{E_{\mathbf{R}_A}(U)_0} & \xrightarrow{[t \mapsto t \bullet_{E_{\mathbf{R}_A}(U)}(), \iota_{\Lambda_n}]_n} & E_{\mathbf{R}_A}(U) \\ & & \downarrow{\scriptstyle \Lambda_{\epsilon_A}} & & \\ & & \Lambda_A & & \end{array}$$

For the final form of the fundamental theorem, we need to show that

$$\Lambda_{\epsilon_A^{-1}} \cdot [t \mapsto t \bullet_{E_{\mathbf{R}_A}(U)} (), \iota_{\Lambda_n}]_n = [a \mapsto \epsilon_A^{-1}(a) \bullet_{E_{\mathbf{R}_A}(U)} (), \iota_{\Lambda_n}]_n$$

is an isomorphism of algebraic theories. By Lemma 62, this is equivalent to its pullback

$$[a \mapsto \epsilon_A^{-1}(a) \bullet_{E_{\mathbf{R}_A}(U)} (), \iota_{\Lambda_n}]_n^* : \mathbf{Alg}_{E_{\mathbf{R}_A}(U)} \to \mathbf{Alg}_{\Lambda_A}$$

being an equivalence of categories.

Lemma 63. *The isomorphisms $\epsilon_A : E_{\mathbf{R}_A}(U)_0 \xrightarrow{\sim} A$ form a natural transformation. That is, for all $h : \mathbf{Alg}_\Lambda(A, B)$, the following diagram commutes:*

$$\begin{array}{ccc} E_{\mathbf{R}_A}(U)_0 & \xrightarrow{h} & E_{\mathbf{R}_B}(U)_0 \\ \downarrow{\scriptstyle \epsilon_A} & & \downarrow{\scriptstyle \epsilon_B} \\ A & \xrightarrow{h} & B \end{array}$$

Proof. This follows from simple unfolding and using the property of algebra morphisms: For all $f : E_{\mathbf{R}_A}(U)_0 = \{f : A \mid (x_1 \circ I_c) \bullet_A f = f\}$,

$$\begin{aligned} \epsilon_B(h(f)) &= (x_1 c_1) \bullet_B h(f) \\ &= h((x_1 c_1) \bullet_A f) \\ &= h(\epsilon_A(f)). \end{aligned}$$

□

Lemma 64. *The pullback functor*

$$[a \mapsto \epsilon_A^{-1}(a) \bullet_{E_{\mathbf{R}_A}(U)} (), \iota_{\Lambda_n}]_n^*$$

is essentially surjective.

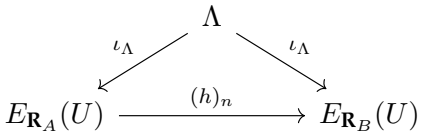
Proof. Take $B : \mathbf{Alg}_{\Lambda_A}$. By Lemma 61 we can consider B to be a object in the coslice category $h : \mathbf{Alg}_\Lambda(A, B)$. As shown in Section 5.4, h sends elements in $\mathbf{R}_A(U^n, U)$ to elements in $\mathbf{R}_B(U^n, U)$, so we can regard it as a morphism $(h)_n : \mathbf{AlgTh}(E_{\mathbf{R}_A}(U), E_{\mathbf{R}_B}(U))$. We then have a $E_{\mathbf{R}_A}(U)$-algebra $(h)_n^*(E_{\mathbf{R}_B}(U)_0)$. Now we need to prove that we have an isomorphism of Λ_A-algebras

$$([a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n \cdot (h)_n)^*(E_{\mathbf{R}_B}(U)_0) \cong B.$$

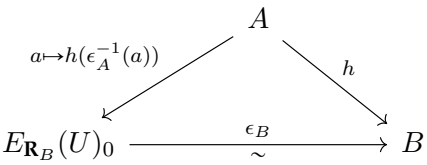
Under the equivalence in Lemma 61, this pullback of $E_{\mathbf{R}_B}(U)_0$ corresponds to some $(B', h') : (A \downarrow \mathbf{Alg}_\Lambda)$. The set of B' is $\mathbf{R}_B(I, U)$, its Λ-action is given by

$$(f, b) \mapsto h(\iota_{\Lambda_n}(f)) \bullet_{E_{\mathbf{R}_B}(U)} b,$$

and the morphism h' is given by $h'(a) = h(\epsilon_A^{-1}(a))$. Note that by initiality of Λ, the following diagram of algebraic theories commutes



Therefore, $h(\iota_{\Lambda_n}(f)) = \iota_{\Lambda_n}(f)$, so the Λ-action on B' is exactly the action on $\iota_\Lambda^*(E_{\mathbf{R}_B}(U)_0)$, which means that $B' = \iota_\Lambda^*(E_{\mathbf{R}_B}(U)_0)$. We have the following diagram in $\mathbf{Alg}_\Lambda$:



By naturality of ϵ, this diagram commutes, which shows that ϵ_B is an isomorphism in the coslice category under A. Pulling this isomorphism back along the equivalence from Lemma 61 gives an isomorphism of Λ_A-algebras

$$([a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n \cdot h)^*(E_{\mathbf{R}_B}(U)_0) \cong B$$

and this concludes the proof. □

Lemma 65. *For $\iota_\Lambda : \mathbf{LamTh}(\Lambda, E_{\mathbf{R}_A}(U))$ and $s_i : \Lambda_n$, we have*

$$\iota_\Lambda((s_i)_i) = \langle x_i \rangle_i \bullet_A (\iota_\Lambda(s_i))_i.$$

Proof. By the recursive nature of the definitions of $(s_i)_i$ and $\langle x_i \rangle_i$, it suffices to show that for $a, b : \Lambda_n$, $\iota_\Lambda((a,b)) = \langle x_1, x_2 \rangle \bullet_A (\iota_\Lambda(a), \iota_\Lambda(b))$ and that $\iota_\Lambda(c_n) = I_c \bullet_A ()$. Since ι_Λ is defined via structural induction, this is just a matter of straightforward but tedious unfolding and rewriting, at some point using the fact that $(x_1 \circ U^n) \bullet_A \iota_\Lambda(a) = \iota_\Lambda(a)$:

$$\begin{aligned}
\iota_\Lambda((a,b)) &= \iota_\Lambda(\lambda x_{n+1}, x_{n+1}\iota_{n,n+1}(a)\iota_{n,n+1}(b)) \\
&= (\lambda x_2 x_3, x_1(x_2, x_3)) \bullet_A \iota_\Lambda(x_{n+1}\iota_{n,n+1}(a)\iota_{n,n+1}(b)) \\
&= (\lambda x_4 x_5, (\lambda x_6, x_1 x_6 (x_2 x_6)(x_3 x_6))(x_4, x_5)) \bullet_A (\iota_\Lambda(x_{n+1}), \iota_\Lambda(a \bullet_\Lambda (x_{n+1,i})_i), \iota_\Lambda(b \bullet_\Lambda (x_{n+1,i})_i)) \\
&= (\lambda x_3 x_4, \pi_{n+1,n+1}(x_3, x_4)((x_1 \circ \langle \pi_{n+1,i} \rangle)_i(x_3, x_4))((x_2 \circ \langle \pi_{n+1,i} \rangle)_i(x_3, x_4))) \bullet_A (\iota_\Lambda(a), \iota_\Lambda(b)) \\
&= (\lambda x_3 x_4, x_4((x_1 \circ \langle \pi_{n,i} \rangle_i)x_3)((x_2 \circ \langle \pi_{n,i} \rangle_i)x_3)) \bullet_A (\iota_\Lambda(a), \iota_\Lambda(b)) \\
&= (\lambda x_3 x_4, x_4(x_1 x_3)(x_2 x_3)) \bullet_A (\iota_\Lambda(a), \iota_\Lambda(b)) \\
&= \langle x_1, x_2 \rangle \bullet_A (\iota_\Lambda(a), \iota_\Lambda(b))
\end{aligned}$$

and

$$\begin{aligned}
\iota_\Lambda(c_n) &= \iota_\Lambda(\lambda x_{n+1}, x_{n+1}) \\
&= (\lambda x_2 x_3, x_1(x_2, x_3)) \bullet_A \iota_\Lambda(x_{n+1}) \\
&= (\lambda x_1 x_2, \pi_{n+1,n+1}(x_1, x_2)) \bullet_A () \\
&= (\lambda x_1 x_2, x_2) \bullet_A () \\
&= I_c \bullet_A ().
\end{aligned}$$

□

Remark 64. Note that for any $E_{\mathbf{R}_A}(U)$-algebra B, we can view the pullback Λ_A-algebra $[a \mapsto \epsilon_A^{-1}(a) \bullet_{E_{\mathbf{R}_A}(U)} (), \iota_{\Lambda_n}]^* B$ as an object in the coslice category under A, given by

$$a \mapsto \epsilon_A^{-1}(a) \bullet_B () : \mathbf{Alg}_\Lambda(A, \iota_\Lambda^* B).$$

Functoriality of the endomorphism theory construction of $U : \mathbf{R}_A$ and $U : \mathbf{R}_{\iota_\Lambda^* B}$, gives a λ-theory morphism

$$a \mapsto \epsilon_A^{-1}(a) \bullet_B () : \mathbf{LamTh}(E_{\mathbf{R}_A}(U), E_{\mathbf{R}_{\iota_\Lambda^* B}}(U)).$$

Then, we can pull back the theory algebra along this morphism, to get, again a $E_{\mathbf{R}_A}(U)$-algebra

$$(a \mapsto \epsilon_A^{-1}(a) \bullet_B ())^*(E_{\mathbf{R}_{\iota_\Lambda^* B}}(U)_0).$$

Lemma 66. *For $B : \mathbf{Alg}_{E_{\mathbf{R}_A}(U)}$, we have an isomorphism of $E_{\mathbf{R}_A}(U)$-algebras given by*

$$\epsilon_{\iota_\Lambda^* B} : (a \mapsto \epsilon_A^{-1}(a) \bullet_B ())^*(E_{\mathbf{R}_{\iota_\Lambda^* B}}(U)_0) \xrightarrow{\sim} B.$$

Proof. Note that the underlying set of $(a \mapsto \epsilon_A^{-1}(a) \bullet_B ())^*(E_{\mathbf{R}_{\iota_\Lambda^* B}}(U)_0)$ is $\mathbf{R}_{\iota_\Lambda^* B}(I, U)$, and that $\epsilon_{\iota_\Lambda^* B}$ is a bijection between this set and B. Now we only need to show that it is a morphism of $E_{\mathbf{R}_A}(U)$-algebras. Take $s : E_{\mathbf{R}_A}(U)_n$ and all $b : E_{\mathbf{R}_{\iota_\Lambda^* B}}(U)_0^n$. We have

$$\begin{aligned}
\epsilon_{\iota_\Lambda^* B}(s \bullet b) &= \iota_\Lambda(x_1 c_1) \bullet_B ((\epsilon_A^{-1}(s) \bullet_B ()) \bullet_{E_{\mathbf{R}_{\iota_\Lambda^* B}}(U)_0} b) \\
&= \iota_\Lambda(x_1 c_1) \bullet_B (\iota_\Lambda(x_1 \circ \langle x_2, \ldots, x_{n+1}\rangle) \bullet_B (\epsilon_A^{-1}(s) \bullet_B (), b_1, \ldots, b_n)) \\
&= \iota_\Lambda(x_1 c_1) \bullet_B ((\iota_\Lambda(x_1 \circ \langle x_{i+1}\rangle_i) \bullet_{E_{\mathbf{R}_A}(U)} (\epsilon_A^{-1}(s), \iota_\Lambda(x_1), \ldots, \iota_\Lambda(x_n))) \bullet_B b) \\
&= (\iota_\Lambda(x_1 c_1) \bullet_{E_{\mathbf{R}_A}(U)} (\iota_\Lambda(x_1 \circ \langle x_{i+1}\rangle_i) \bullet_{E_{\mathbf{R}_A}(U)} (\epsilon_A^{-1}(s), \iota_\Lambda(x_1), \ldots, \iota_\Lambda(x_n)))) \bullet_B b \\
&= (\iota_\Lambda(x_1(x_{i+1}c_{n+1})_i) \bullet_{E_{\mathbf{R}_A}(U)} (\epsilon_A^{-1}(s), \iota_\Lambda(x_1), \ldots, \iota_\Lambda(x_n))) \bullet_B b \\
&= (\iota_\Lambda(x_1 x_2) \bullet_{E_{\mathbf{R}_A}(U)} (\epsilon_A^{-1}(s), \iota_\Lambda((x_i c_n)_i))) \bullet_B b \\
&= ((\iota_\Lambda(x_1 x_2) \bullet_{E_{\mathbf{R}_A}(U)} (\epsilon_A^{-1}(s), \iota_\Lambda((x_i)_i))) \bullet_{E_{\mathbf{R}_A}(U)} (x_i c_n)_i) \bullet_B b
\end{aligned}$$

Also,

$$\begin{aligned}
s \bullet_B (\epsilon_B(b_i))_i &= s \bullet_B (\iota_\Lambda(x_1 c_1) \bullet_B b_i)_i \\
&= s \bullet_B (\iota_\Lambda(x_i c_n) \bullet_B (b_j)_j)_i \\
&= (s \bullet_{E_{\mathbf{R}_A}(U)} (\iota_\Lambda(x_i c_n))_i) \bullet_B b.
\end{aligned}$$

Then the result follows from the fact that

$$\begin{aligned}
\iota_\Lambda(x_1 x_2) \bullet_{E_{\mathbf{R}_A}(U)} (\epsilon_A^{-1}(s), \iota_\Lambda((x_i)_i)) &= \iota_\Lambda(x_1 x_2) \bullet_{E_{\mathbf{R}_A}(U)} ((\lambda x_2, x_1) \bullet_A s, \langle x_i\rangle_i \bullet_A (\iota_\Lambda(x_i))_i) \\
&= (\lambda x_3, x_1 x_3 x_2 x_3) \bullet_A ((\lambda x_2, x_1) \bullet_A s, \langle x_i\rangle_i \bullet_A (\pi_{n,i} \bullet_A ())_i) \\
&= (\lambda x_3, x_1 x_2 x_3) \bullet_A (s, \langle \pi_{n,i}\rangle_i \bullet_A ()) \\
&= (x_1 \circ x_2) \bullet_A (s, U^n \bullet_A ()) \\
&= s.
\end{aligned}$$

□

Lemma 67. *$[a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n^*$ is fully faithful.*

Proof. First of all, note that the pullback functor is always faithful, since it preserves all the 'data' (i.e. the functions) of the algebra morphisms.

Now, to show that it is full, take $B, C : \mathbf{Alg}_{E_{\mathbf{R}_A}(U)}$. Also, take a morphism

$$h : \mathbf{Alg}_{\Lambda_A}([a \mapsto \epsilon_A^{-1}(a) \bullet_B (), \iota_{\Lambda_n}]_n^* B, [a \mapsto \epsilon_A^{-1}(a) \bullet_C (), \iota_{\Lambda_n}]_n^* C).$$

By Remark 64, and by functoriality of the endomorphism theory construction, we get a commutative diagram

$$\begin{array}{ccc} & E_{\mathbf{R}_A}(U) & \\ {\scriptstyle a\mapsto \epsilon_A^{-1}(a)\bullet_B()}\swarrow & & \searrow{\scriptstyle a\mapsto \epsilon_A^{-1}(a)\bullet_C()} \\ E_{\mathbf{R}_{\iota_\Lambda^* B}}(U) & \xrightarrow{(h)_n} & E_{\mathbf{R}_{\iota_\Lambda^* C}}(U) \end{array}$$

Now, using Definition 43 for the coslice category $(E_{\mathbf{R}_A}(U) \downarrow \mathbf{AlgTh})$, and using the previous lemma, we get the following diagram of $E_{\mathbf{R}_A}(U)$-algebras:

$$\begin{array}{ccc} (a \mapsto \epsilon_A^{-1}(a) \bullet_B ())^*(E_{\mathbf{R}_{\iota_\Lambda^* B}}(U)_0) & \xrightarrow{(a\mapsto \epsilon_A^{-1}(a)\bullet())^* h} & (a \mapsto \epsilon_A^{-1}(a) \bullet_C ())^*(E_{\mathbf{R}_{\iota_\Lambda^* C}}(U)_0) \\ \sim\downarrow{\scriptstyle \epsilon_{\iota_\Lambda^* B}} & & \sim\downarrow{\scriptstyle \epsilon_{\iota_\Lambda^* C}} \\ B & \xrightarrow{\bar{h}} & C \end{array}$$

This gives us the lift

$$\bar{h} = \epsilon^{-1}_{\iota_\Lambda^* B} \cdot (a \mapsto \epsilon_A^{-1}(a) \bullet ())^* h \cdot \epsilon_{\iota_\Lambda^* C} : \mathbf{Alg}_{E_{\mathbf{R}_A}(U)}(B, C)$$

Now, when we again pull back this map to

$$[a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n^* \bar{h} : \mathbf{Alg}_{\Lambda_A}([a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n^* B, [a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n^* C).$$

Note that by naturality of ϵ, we have

$$\bar{h} = \epsilon^{-1}_{\iota_\Lambda^* B} \cdot h \cdot \epsilon_{\iota_\Lambda^* C} = \epsilon^{-1}_{\iota_\Lambda^* B} \cdot \epsilon_{\iota_\Lambda^* B} \cdot h = h$$

as functions, and therefore $[a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n^* \bar{h} = h$, so the pullback is full. □

Lemma 68. *Λ_A is isomorphic to $E_{\mathbf{R}_A}(U)$ as an algebraic theory.*

Proof. By Lemmas 64 and 67, the pullback functor $[a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n^*$ is a weak equivalence. Since algebra categories are univalent, this means that the pullback functor is an equivalence of categories. Then, by Lemma 33, $[a \mapsto \epsilon_A^{-1}(a) \bullet (), \iota_{\Lambda_n}]_n : \mathbf{AlgTh}(\Lambda_A, E_{\mathbf{R}_A}(U))$ is an isomorphism. □

Now, to show that we can replace the functor $A \mapsto E_{\mathbf{R}_A}(U)$ by $A \mapsto \Lambda_A$, we can show that the functors are isomorphic:

Lemma 69. *The isomorphisms $[a \mapsto \epsilon_A^{-1}(a) \bullet_{E_{\mathbf{R}_A}(U)} (), \iota_{\Lambda_n}]_n : \mathbf{AlgTh}(\Lambda_A, E_{\mathbf{R}_A}(U))$ form a natural isomorphism between the functors*

$$A \mapsto E_{\mathbf{R}_A}(U), \quad h \mapsto (h)_n \quad \textit{and} \quad A \mapsto \Lambda_A, \quad h \mapsto \Lambda_h.$$

Proof. We must show that for all $h : \mathbf{Alg}_\Lambda(A, B)$ and for all n, the following diagram of Λ-algebras commutes:

$$\begin{array}{ccc} A + \Lambda_n & \xrightarrow{[a\mapsto \epsilon_A^{-1}(a)\bullet(), \iota_{\Lambda_n}]} & \iota_\Lambda^*(E_{\mathbf{R}_A}(U)_n) \\ {\scriptstyle h+\mathrm{id}_{\Lambda_n}}\downarrow & & \downarrow{\scriptstyle h} \\ B + \Lambda_n & \xrightarrow[[a\mapsto \epsilon_B^{-1}(a)\bullet(), \iota_{\Lambda_n}]]{} & \iota_\Lambda^*(E_{\mathbf{R}_B}(U)_n) \end{array}$$

Now, by the universal property of the coproduct $A + \Lambda_n$, it suffices to check that for all $a : A$ and $t : \Lambda_n$,

$$h(\epsilon_A^{-1}(a) \bullet_{E_{\mathbf{R}_A}(U)} ()) = \epsilon_B^{-1}(h(a)) \bullet_{E_{\mathbf{R}_B}(U)} () \quad \text{and} \quad h(\iota_{\Lambda_n}(a)) = \iota_{\Lambda_n}(a).$$

The former follows from the fact that $(h)_n : \mathbf{LamTh}(E_{\mathbf{R}_A}(U), E_{\mathbf{R}_B}(U))$ respects the substitution and from the naturality of ϵ^{-1}:

$$h(\epsilon_A^{-1}(a) \bullet_{E_{\mathbf{R}_A}(U)} ()) = h(\epsilon_A^{-1}(a)) \bullet_{E_{\mathbf{R}_B}(U)} () = \epsilon_B^{-1}(h(a)) \bullet_{E_{\mathbf{R}_B}(U)} (),$$

whereas the latter follows from the initiality of Λ, and the fact that $(h)_n : E_{\mathbf{R}_A}(U) \to E_{\mathbf{R}_B}(U)$ is a λ-theory morphism. □

Remark 65. Now, note that for any Λ-algebra A, Λ_A is an algebraic theory. However, for our equivalence of categories, we need a functor to the category of λ-theories. By the natural isomorphism above (which respects the algebraic theory structures), we see that the objects and morphisms in the images of $E_{\mathbf{R}_-}(U)$ and Λ_- have 'the same' algebraic theory structures, and we can transfer the additional λ-theory structures from $E_{\mathbf{R}_-}(U)$ to Λ_-. With some abuse of notation, this yields a functor $\Lambda_- : \mathbf{Alg}_\Lambda \to \mathbf{LamTh}$.

The final form of Hyland's representation theorem is the following:

Theorem 7. *The functor that sends a λ-theory L to the Λ-algebra L_0 and the functor that sends a Λ-algebra A to the theory of extensions Λ_A form an adjoint equivalence*

$$\mathbf{LamTh} \cong \mathbf{Alg}_\Lambda.$$

Proof. By 6, we have an adjoint equivalence given by

$$L \mapsto L_0 \quad \text{and} \quad A \mapsto E_{\mathbf{R}_A}(U).$$

By the previous lemma, the second functor is isomorphic to $\Lambda_- : \mathbf{Alg}_\Lambda \to \mathbf{LamTh}$. Therefore, we can replace one by the other.

There are two ways to see this: We may notice that we can transfer the unit, the counit and the two zigzag identities of the adjunction along the natural isomorphism and show that this all works together. As an alternative, we can also notice that the category of λ-theories is univalent, so the functor category $\mathbf{Alg}_\Lambda \to \mathbf{LamTh}$ is univalent and the natural isomorphism between the functors is an equality, and we can replace one by the other. □

Chapter 6

The formalization

6.1 Statistics

6.2 Components

6.3 Displayed categories

6.3.1 Fibrations

6.3.2 Defininig objects by their category instead of the other way around

6.3.3 _ax for categories and their objects

6.3.4 Cartesian vs cartesian′ for products

6.3.5 Limits

Limits in a fibered category

6.4 Inductive types

6.5 The formalization of the λ-calculus

Defining Lambda Calculus in a different way (not as an axiomatized HIT) - As set quotient instead of HIT - With a signature

6.6 Tuples

$stnm \to A$ vs vec A

6.7 Products

$T \times (T \times \cdots \times T)$ vs $T \times T^n vs T^(Sn)$ Terminal as product over empty set over any set with a function to empty.

6.8 The $n + p$-presheaf

L (S n) (for lambda) vs L (n + 1) (stemming from the naive implementation of the L (n + p) presheaf)

6.9 Quotients

Quotients (by hrel or eqrel) vs coproducts (generalizing to arbitrary category with coproduct) vs a category with some structure

6.10 The Karoubi envelope

KanExtension instead of specific construction at KaroubiEnvelope

6.11 Univalence

Univalence bewijzen via isweqhomot vs direct

6.12 Equality, Iso's and Equivalence (of categories)

It is important to choose the right kind of equality to prove.

Bibliography

[ACU14] Thosten Altenkirch, James Chapman, and Tarmo Uustalu. "Monads need not be endofunctors". In: (2014). DOI: 10.2168/LMCS-11(1:3)2015. eprint: arXiv:1412.7148.

[AKS15] Benedikt Ahrens, Krzysztof Kapulkin, and Michael Shulman. "Univalent categories and the Rezk completion". In: *Mathematical Structures in Computer Science* 25.5 (Jan. 2015), pp. 1010–1039. ISSN: 1469-8072. DOI: 10.1017/s0960129514000486. URL: http://dx.doi.org/10.1017/S0960129514000486.

[ARV10] J. Adámek, J. Rosický, and E. M. Vitale. *Algebraic Theories: A Categorical Introduction to General Algebra*. Cambridge Tracts in Mathematics. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511760754.

[AW23] Benedikt Ahrens and Kobe Wullaert. *Category Theory for Programming*. 2023. eprint: arXiv:2209.01259.

[Bez+20] Marc Bezem et al. *Construction of the Circle in UniMath*. 2020. arXiv: 1910.01856 [math.LO].

[Bor94] Francis Borceux. *Handbook of Categorical Algebra*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1994.

[Fre72] Peter Freyd. "Aspects of topoi". In: *Bulletin of the Australian Mathematical Society* 7.1 (Aug. 1972), pp. 1–76. ISSN: 1755-1633. DOI: 10.1017/S0004972700044828. URL: http://dx.doi.org/10.1017/S0004972700044828.

[Hay85] Susumu Hayashi. "Adjunction of semifunctors: Categorical structures in nonextensional lambda calculus". In: *Theoretical Computer Science* 41 (1985), pp. 95–104. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(85)90062-3. URL: https://www.sciencedirect.com/science/article/pii/0304397585900623.

[HP89] J. Martin E. Hyland and Andrew M. Pitts. "The theory of constructions: Categorical semantics and topos-theoretic models". In: *Contemporary Mathematics* 92 (1989), pp. 137–199. DOI: 10.1090/conm/092/1003199. URL: http://dx.doi.org/10.1090/conm/092/1003199.

[HS93] Raymond Hoofman and Harold Schellinx. *Models of the untyped lambda-calculus in semi cartesian closed categories*. ILLC Prepublication Series for Mathematical Logic and Foundations, ML-93-05. Feb. 1993. eprint: https://eprints.illc.uva.nl/id/eprint/1329/1/ML-1993-05.text.pdf.

[htt] Mark Kamsma (https://math.stackexchange.com/users/661457/mark-kamsma). *Show that the Yoneda embedding preserves exponential objects*. Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/3511278 (version: 2022-12-20). eprint: https://math.stackexchange.com/q/3511278. URL: https://math.stackexchange.com/q/3511278.

[Hur95] Antonius J. C. Hurkens. "A simplification of Girard's paradox". In: *Typed Lambda Calculi and Applications*. Ed. by Mariangiola Dezani-Ciancaglini and Gordon Plotkin. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 266–278. ISBN: 978-3-540-49178-1.

[Hyl14] Martin Hyland. "Towards a Notion of Lambda Monoid". In: *Electronic Notes in Theoretical Computer Science* 303 (2014). Proceedings of the Workshop on Algebra, Coalgebra and Topology (WACT 2013), pp. 59–77. ISSN: 1571-0661. DOI: https://doi.org/10.1016/j.entcs.2014.02.004. URL: https://www.sciencedirect.com/science/article/pii/S1571066114000309.

[HYL17] J.M.E. HYLAND. "Classical lambda calculus in modern dress". In: *Mathematical Structures in Computer Science* 27.5 (2017), pp. 762–781. DOI: 10.1017/S0960129515000377.

[KS06] Masaki Kashiwara and Pierre Schapira. *Categories and sheaves*. Vol. 332. Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer-Verlag, Berlin, 2006, pp. x+497. ISBN: 978-3-540-27949-5; 3-540-27949-0. DOI: 10.1007/3-540-27950-4. URL: https://doi.org/10.1007/3-540-27950-4.

[Law69] F. William Lawvere. "Diagonal arguments and cartesian closed categories". In: *Category Theory, Homology Theory and their Applications II*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1969, pp. 134–145. ISBN: 978-3-540-36101-5. DOI: https://doi.org/10.1007/BFb0080769.

[Mac98] Saunders Mac Lane. *Categories for the working mathematician*. Second. Vol. 5. Graduate Texts in Mathematics. Springer-Verlag, New York, 1998, pp. xii+314. ISBN: 0-387-98403-8.

[Mar71] Per Martin-Löf. "A theory of types". Preprint, Stockholm University. 1971.

[nLa24a] nLab authors. *adjoint equivalence*. https://ncatlab.org/nlab/show/adjoint+equivalence. Revision 17. May 2024.

[nLa24b] nLab authors. *Grothendieck fibration*. https://ncatlab.org/nlab/show/Grothendieck+fibration. Revision 113. Feb. 2024.

[Rie14] Emily Riehl. *Categorical Homotopy Theory*. New Mathematical Monographs. Cambridge University Press, 2014.

[Sco16] Dana S. Scott. *Greetings to the Participants at "Strachey 100"*. https://www.cs.ox.ac.uk/strachey100/Strachey_booklet.pdf. A talk read out at the Strachey 100 centenary conference. 2016.

[Sco72] Dana Scott. "Continuous lattices". In: *Toposes, Algebraic Geometry and Logic*. Ed. by F. W. Lawvere. Berlin, Heidelberg: Springer Berlin Heidelberg, 1972, pp. 97–136. ISBN: 978-3-540-37609-5. URL: http://dx.doi.org/10.1007/BFb0073967.

[SH80] J. P. Seldin and J. R. Hindley, eds. *To H.B. Curry: Essays on Combinatory Logic and Formalism*. en. San Diego, CA: Academic Press, Sept. 1980.

[Tay86] Paul Taylor. "Recursive Domains, Indexed Category Theory and Polymorphism". PhD thesis. University of Cambridge, 1986.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book, 2013.

Appendix A

Alternative definitions

The literature, there are many different but equivalent definitions, carrying many different names, for the objects that called 'algebraic theories' in Section 3.1. Also, many of these different names are attached to differently defined objects in different sources. This section will showcase some of the various definitions.

In this section, we will denote the finite set $[\![n]\!] = \{1, \dots, n\}$.

A.1 Abstract Clone

Definition 58. An algebraic theory as presented in Section 3.1, is usually called an *abstract clone*. In this thesis, outside of this specific section, we will call it algebraic theory to be consistent with the names that Hyland attaches to objects.

Remark 66. The definition of algebraic theory that Hyland gives is closest to that of an abstract clone. However, instead of a sequence of sets $(T_n)_n$, he requires a functor $T : F \to \mathbf{Set}$ (with $F \subseteq \mathbf{FinSET}$ the skeleton category of finite sets $F_0 = \{[\![0]\!], [\![1]\!], \dots\}$), and he requires $\bullet : T_m \times T_n^m \to T_n$ to be dinatural in n and natural in m.

Using naturality, one can show that with such a functor we have $x_{n,i} = f(a)(x_{1,1})$ for the function $a(1) = i : [\![n]\!]$.

Alternatively, using the same naturality, one can show that this functor sends a morphism $a : [\![m]\!] \to [\![n]\!]$ to the function $T_m \to T_n$ given by

$$f \mapsto f \bullet (x_{n,a(i)})_i.$$

If we take this to be the definition of our functor on morphisms, the (di)naturality in m and n can be shown using the associativity and the laws about the interaction between $\bullet$ and the x_i.

Since any additional properties mean extra complexity when formalizing, and since the proofs rarely use the functor structure, we decided to reduce the functor $T : \mathbf{FinSET} \to \mathbf{Set}$ to a sequence of sets $(T_n)_n$.

A.2 Lawvere theory

Definition 59. An *algebraic theory* as presented in [ARV10] is a small category with finite products.

Definition 60. An *algebra* for an algebraic theory T is a finite-product-preserving functor $T \to \mathbf{Set}$.

This definition is more general than the definition of algebraic theory in Section 3.1. To make it equivalent, we have to be more specific about the objects of the category:

Definition 61. A *Lawvere theory*, or *one-sorted algebraic theory* is a category L, with $L_0 = \{0, 1, \dots\}$, such that $n = 1^n$, the n-fold product.

Lemma 70. *There is an equivalence between abstract clones and Lawvere theories.*

Proof. Let C be an abstract clone. We construct a Lawvere theory L as follows: We have objects $L_0 = \{0, 1, \dots\}$ and morphisms $L(m, n) = C_m^n$. The identity morphism is $\mathrm{id}_n = (x_i)_i : L(n, n)$ and for $f : L(l, m)$, $g : L(m, n)$, we have composition

$$f \cdot g = (g_i \bullet f)_i : L(l, n).$$

Lastly, we have product projections $\pi_{n,i} = x_{n,i} : L(n, 1)$ for all $1 \leqslant i \leqslant n$.

Conversely, if L is a Lawvere theory, we construct an abstract clone C as follows: We take $C_n = L(n, 1)$. We take the $x_{n,i}$ to be the product projections $\pi_i : L(n, 1)$ and we define the substitution $f \bullet g = \langle g_i \rangle_i \cdot f$ the composite of the product morphism $\langle g_i \rangle_i$ with f. □

A.2.1 Algebras for Lawvere Theories

Lemma 71. *Let C be an abstract clone, and L be its associated Lawvere theory by the equivalence given above. A C-algebra is equivalent to an algebra for L.*

Proof. Let A be a C-algebra. We will construct a functor $F : L \to \mathbf{Set}$ as follows: We take $F(n) = A^n$. We define the action on morphisms as

$$F(f)(a) = (f_i \bullet a)_i$$

for $f : L(m, n) = L(m, 1)^n$ and $a : A^m$.

Conversely, let $F : L \to \mathbf{Set}$ be a functor. We take the C-algebra A, with $A = F(1)$ and for $f : C_n = L(n, 1)$ and $a : A^n$, $f \bullet a = F(f)(a)$. □

A.2.2 Presheaves for Lawvere Theories

Lemma 72. *Let C be an abstract clone, and L be its associated Lawvere theory by the equivalence given above. A C-presheaf is equivalent to a presheaf over L.*

Proof. The correspondence between C-presheaves P and presheaves Q over L is as follows:

The sets P_n correspond to the action of Q on objects $Q(n)$.

The P-action $P_m \times C_n^m \to P_n$ corresponds to the action of Q on morphisms $L(n, m) \times Q(m) \to Q(n)$. □

A.3 Cartesian Operad

Definition 62. A *cartesian operad* T is a functor $T : F \to \mathbf{Set}$, together with an 'identity' element $\mathrm{id} : T(1)$ and for all $m, n_1, \dots, n_m : \mathbb{N}$, a composition map

$$T(m) \times \prod_i T(n_i) \to T\left(\sum_i n_i\right),$$

written $(f, g_1, \dots, g_m) \mapsto f[g_1, \dots, g_m]$, satisfying some identity, associativity and naturality conditions (see [Hyl14], Definition 2.1, for more details).

Remark 67. One can arrive at this concept as a standalone definition, or one can view a cartesian operad as a *cartesian multicategory* with one element. A multicategory is a category in

which morphisms have type $C((X_1, X_2, \ldots, X_n), Y)$ instead of $C(X, Y)$. A cartesian multicategory is a multicategory in which one can permute the X_i of a morphism and has 'contraction' and 'weakening' operations:

$$C((X_1, \ldots, X_i, X_i, \ldots, X_n), Y) \to C((X_1, \ldots, X_i, \ldots, X_n), Y),$$
$$C((X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n), Y) \to C((X_1, \ldots, X_{i-1}, X_i, X_{i+1}, \ldots, X_n), Y).$$

Lemma 73. *There is an equivalence between abstract clones and cartesian operads.*

Proof. Let C be an abstract clone. We define a cartesian operad T with $T(n) = C_n$ and $T(f)(t) = t \bullet (x_{f(1)}, \ldots, x_{f(m)})$ for $f : [\![m]\!] \to [\![n]\!]$ and $t : C_m$. The identity element is $x_{1,1}$ and the composition $f[g_1, \ldots, g_n]$, for $f : C_m$ and $g_i : C_{n_i}$, is given by lifting all terms to $C_{\sum_i n_i}$ and then substituting:

$$f[g_1, \ldots, g_n] = f \bullet (T(\iota_i)(g_i))$$

for $\iota_i : [\![n_i]\!] \hookrightarrow [\![\sum_i n_i]\!]$ the pairwise disjoint injections.

Conversely, given a cartesian operad T, we construct an abstract clone C with $C_n = T([\![n]\!])$. The variables are $x_{n,i} = \iota_{n,i}(\mathrm{id})$, for $\iota_{n,i} : [\![1]\!] \hookrightarrow [\![n]\!]$ the morphism that sends 1 to i. The substitution $f \bullet (g_i)_i$ for $g_1, \ldots, g_m : T(n)$ is given by composing and then identifying some variables:

$$f \bullet (g_i)_i = T(\pi)(f[g_1, \ldots, g_m])$$

for $\pi : [\![mn]\!] \to [\![n]\!]$ the function that sends $i + 1$ to $(i \mod n) + 1$. □

A.4 Relative Monad

Definition 63. Let $S : C \to D$ be a functor. A *relative monad* on S is a functor $T : C \to D$, together with a natural transformation $\eta : S \Rightarrow T$ and a 'kleisli extension' $(-)^* : D(S(X), T(Y)) \to D(T(X), T(Y))$, natural in both S and T, such that for all $f : D(S(X), T(Y))$ and $g : D(S(Y), T(Z))$,

$$\eta^*_X = \mathrm{id}_{TX}, \qquad f = \eta_X \cdot f^* \qquad \text{and} \qquad (f \cdot g^*)^* = f^* \cdot g^*.$$

Remark 68. Note that for an adjunction $F \dashv G$, $F \bullet G$ gives a monad. In the same way, there exists a notion of *relative adjunction*, from which we can obtain a relative monad (see [ACU14], Theorem 2.10).

Remark 69. Now, there is a result that states: There is an equivalence between abstract clones and relative monads on the embedding $\iota :$ **FinSET** $\hookrightarrow$ **Set**.

Note that the objects of **FinSET** are defined to be sets X, together with a proof that there exists some $n : \mathbb{N}$ and some bijection $f : X \xrightarrow{\sim} [\![n]\!]$. This existence of n and f is given by the propositional truncation $\left\| \sum_{n:\mathbb{N}}, X \xrightarrow{\sim} [\![n]\!] \right\|$.

Classically, this construction starts with "fix, for all $X :$ **FinSET**, a bijection $f : X \xrightarrow{\sim} [\![n]\!]$". However, since X only provides *mere existence* of such a bijection without choosing one (by the propositional truncation), there is no way to obtain f without using the axiom of choice.

Now, we can partially circumvent this problem by noting that we have a fully faithful and essentially surjective embedding $F \to$ **FinSET** (for F a skeleton of finite sets), which induces an adjoint equivalence $[F,$ **Set**$] \xrightarrow{\sim} [$**FinSET**, **Set**$]$, so we can lift the functor part of a relative monad on $F \to$ **Set** to **FinSET** $\to$ **Set**. We can also lift the natural transformation using this equivalence. However, the kleisli extension cannot be lifted using this and we are stuck.

Therefore, we will prove a modified statement:

Lemma 74. *There is an equivalence between abstract clones and relative monads on the embedding* $\iota : F \hookrightarrow$ **Set**.

Proof. Let C be an abstract clone. We define a relative monad T as follows: We take $T(\llbracket n \rrbracket) = C_n$. For a morphism $a : F(\llbracket m \rrbracket, \llbracket n \rrbracket)$. We take $T(a)(f) = f \bullet (x_{a(i)})_i$. We define $\eta_{\llbracket n \rrbracket}(i) = x_{n,i}$. Finally, for $g : \mathbf{Set}(\llbracket m \rrbracket, T(\llbracket n \rrbracket))$, we define $g^*(f) = f \bullet g$.

Conversely, let $(T, \eta, (\cdot)^*)$ be a relative monad on the embedding $\iota : F \hookrightarrow \mathbf{Set}$. We define an abstract clone C with $C_n = T(\llbracket n \rrbracket)$. Substitution is defined as $f \bullet g = g^*(f)$ for $f : C_m$ and $g : C_n^m$ and we have variables $x_{n,i} = \eta_{\llbracket n \rrbracket}(i)$. □

A.5 Monoid in a skew-monoidal category

For details and a general treatment, see [ACU14], Section 3 and specifically Theorem 3.4.

Definition 64. Consider the category $[F, \mathbf{Set}]$. Note that we have a functor $\iota : F \hookrightarrow \mathbf{Set}$ and that **Set** has colimits. Therefore, we can define a 'tensor product' on functors $[F, \mathbf{Set}]$ as

$$F \otimes G = G \bullet \mathrm{Lan}_\iota F.$$

Together with the 'unit' ι, this gives $[F, \mathbf{Set}]$ a 'skew-monoidal category' structure.

Definition 65. Given a (skew-)monoidal category $(C, \otimes, I)$, a *monoid* in this category is an object $T : C$, together with a 'multiplication' $\mu : C(T \otimes T, T)$ and a 'unit' morphism $\eta : C(I, T)$ satisfying a couple of laws (see [Mac98], Section III.6).

Lemma 75. *There exists an equivalence between relative monads on $\iota : F \hookrightarrow \mathbf{Set}$ and monoids in the skew-monoidal category $[F, \mathbf{Set}]$.*

Proof. A monoid in $[F, \mathbf{Set}]$ consists of an object $T : [F, \mathbf{Set}]$, together with natural transformations $\mu : T \otimes T \Rightarrow T$ and $\eta : \iota \Longrightarrow T$. We can immediately see the functor T and natural transformation η of the relative monad pop up here. The kleisli extension corresponds to μ; they are related to each other via the properties of the left Kan extension of T. □

Appendix B

Weak Cartesian Closed Categories

In Definition 41, we defined the endomorphism λ-theory of a reflexive object in a cartesian closed category. However, as it turns out, we do not need all of the properties of a cartesian closed category. It suffices to consider a *weak* cartesian closed category. In fact, even a 'semi cartesian closed category' will suffice, but in this section, we will restrict ourselves to a weak cartesian closed category, as this notion suits our purposes well enough.

It seems that semi cartesian closed categories were first defined in [Hay85] and then [HS93] gave a slightly different definition, as well as the very related definition of a weak cartesian closed category. The latter paper also contains the construction of the endomorphism λ-theory.

Definition 66. Let C be a category with binary products (we will call the projections π_1 and π_2), and take $A, B : C$. A *semi-exponential object* is an object $A^B : C$, with a morphism $\text{ev} : C(A^B \times B \to A)$ and a hayashi-1985-semifunctorsfunction

$$\text{cur}_X : C(X \times B \to A) \to C(X, A^B)$$

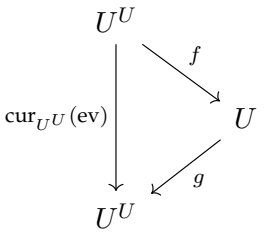
for all X, such that

$$\langle g \cdot \text{cur}_X(f), h\rangle \cdot \text{ev} = \langle g, h\rangle \cdot f \quad \text{and} \quad \text{cur}_Y((g \times \text{id}_B) \cdot f) = g \cdot \text{cur}_X(f)$$

for all $f : C(X \times B, A)$, $g : C(Y, X)$ and $h : C(Y, B)$.

Definition 67. A *weak cartesian closed category* is a category C with a terminal object and binary products, together with a choice for a semi-exponential object A^B for all $A, B : C$.

Definition 68. Let C a weak cartesian closed category. A *reflexive object* in C is an object U with morphisms f and g, forming the following commutative diagram:



Definition 69. If we have a reflexive object (U, f, g) in a weak cartesian closed category C, we can endow the endomorphism algebraic theory $E_C(U)$ (Definition 41) with a λ-theory structure with β-equality, with

$$\rho(s) = ((s \cdot g) \times \text{id}_U) \cdot \text{ev} : C(U^{n+1}, U)$$

and

$$\lambda(t) = \mathrm{cur}_{U^n}(t) \cdot f : C(U^n, U)$$

for $s : E(U)_n$ and $t : E(U)_{n+1}$.

Lemma 76. *The definition above indeed yields a λ-theory with β-equality.*

Proof. We have

$$\begin{aligned}\rho(s \bullet t) &= ((\langle t_i \rangle_i \cdot (s \cdot g)) \times \mathrm{id}_U) \cdot \mathrm{ev} \\ &= (\langle t_i \rangle_i \times \mathrm{id}_U) \cdot ((s \cdot g) \times \mathrm{id}_U) \cdot \mathrm{ev} \\ &= \rho(s) \bullet ((\iota_{n,1}(t_i))_i + (x_{n+1,n+1})),\end{aligned}$$

$$\begin{aligned}\lambda(s) \bullet t &= \langle t_i \rangle_i \cdot \mathrm{cur}(t) \cdot f \\ &= \mathrm{cur}_{U^n}((\langle t_i \rangle_i \times \mathrm{id}_U) \cdot t) \cdot f \\ &= \lambda(s \bullet ((\iota_{n,1}(t_i))_i + (x_{n+1,n+1})))\end{aligned}$$

and

$$\begin{aligned}\rho(\lambda(s)) &= \langle \pi_1 \cdot \mathrm{cur}_{U^n}(t) \cdot \mathrm{cur}_{U^U}(\mathrm{ev}), \pi_2 \rangle \cdot \mathrm{ev} \\ &= \langle \pi_1 \cdot \mathrm{cur}_{U^n}(t), \pi_2 \rangle \cdot \mathrm{ev} \\ &= t.\end{aligned}$$

□

Now, we will show that the semi-exponential objects satisfy some properties that justify the use of the exponential notation. These properties allow us to construct semi-exponentials on products, from the semi-exponentials of their components.

Lemma 77. *Take $A, A', B : C$. If A^B and A'^B are semi-exponential objects, then $A^B \times (A')^B$ is the semi-exponential object $(A \times A')^B$.*

Proof. We have

$$\begin{aligned}\mathrm{ev} : C(A^B \times B \to A) \quad &\text{and} \quad \mathrm{ev}' : C(A'^B \times B \to A'), \\ \mathrm{cur}_X : C(X \times B, A) \to C(X, A^B) \quad &\text{and} \quad \mathrm{cur}'_X : C(X \times B, A') \to C(X, A'^B).\end{aligned}$$

Define

$$\mathrm{ev}'' = \langle \langle \pi_1 \cdot \pi_1, \pi_2 \rangle \cdot \mathrm{ev}, \langle \pi_1 \cdot \pi_2, \pi_2 \rangle \cdot \mathrm{ev} \rangle : C(A^B \times A'^B, A \times A')$$

and for $f : C(X \times B, A \times A')$,

$$\mathrm{cur}''_X(f) = \langle \mathrm{cur}_X(f \cdot \pi_1), \mathrm{cur}_X(f \cdot \pi_2) \rangle : C(X, A^B \times A^{B'}).$$

These satisfy the required equations. □

Lemma 78. *Take $A, B, B' : C$. If A^B and $(A^B)^{B'}$ are semi-exponential objects, then $(A^B)^{B'}$ is the semi-exponential object $A^{B \times B'}$.*

Proof. We have

$$\begin{aligned}\mathrm{ev} : C(A^B \times B \to A) \quad &\text{and} \quad \mathrm{ev}' :: C((A^B)^{B'} \times B' \to A^B), \\ \mathrm{cur}_X : C(X \times B, A) \to C(X, A^B) \quad &\text{and} \quad \mathrm{cur}'_X : C(X \times B', A^B) \to C(X, (A^B)^{B'}).\end{aligned}$$

Define

$$\mathrm{ev}'' = \langle (\mathrm{id}_{(A^B)^{B'}} \times \pi_2) \cdot \mathrm{ev}', \pi_2 \cdot \pi_1 \rangle \cdot \mathrm{ev} : C((A^B)^{B'} \times (B \times B'), A)$$

and for $f : C(X \times (B \times B'), A)$,

$$\mathrm{cur}''_X(f) = \mathrm{cur}'_X(\mathrm{cur}_{X \times B'}(\langle \pi_1 \cdot \pi_1, \langle \pi_2, \pi_1 \cdot \pi_2 \rangle \rangle \cdot f)) : C(X, (A^B)^{B'}).$$

Then, for $f : C(X \times (B \times B'), A)$, $g : C(Y, X)$ and $h : C(Y, B \times B')$,

$$\begin{aligned}
\langle g \cdot \mathrm{cur}''_X(f), h \rangle \cdot \mathrm{ev}'' &= \langle \langle g \cdot \mathrm{cur}'_X(\mathrm{cur}_{X \times B'}(\langle \pi_1 \cdot \pi_1, \langle \pi_2, \pi_1 \cdot \pi_2 \rangle \rangle \cdot f)), h \cdot \pi_2 \rangle \cdot \mathrm{ev}', h \cdot \pi_1 \rangle \cdot \mathrm{ev} \\
&= \langle \langle \langle g, h \cdot \pi_2 \rangle, h \cdot \pi_1 \rangle \cdot \pi_1 \cdot \pi_1, \langle \langle \langle g, h \cdot \pi_2 \rangle, h \cdot \pi_1 \rangle \cdot \pi_2, \langle \langle g, h \cdot \pi_2 \rangle, h \cdot \pi_1 \rangle \cdot \pi_1 \cdot \pi_2 \rangle \rangle \cdot f \\
&= \langle g, h \rangle \cdot f
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{cur}''_Y((g \times \mathrm{id}_B) \cdot f) &= \mathrm{cur}'_Y(\mathrm{cur}_{Y \times B'}(((g \times \mathrm{id}_{B'}) \times \mathrm{id}_B) \cdot \langle \pi_1 \cdot \pi_1, \langle \pi_2, \pi_1 \cdot \pi_2 \rangle \rangle \cdot f)) \\
&= \mathrm{cur}'_Y((g \times \mathrm{id}_{B'}) \cdot \mathrm{cur}_{X \times B'}(\langle \pi_1 \cdot \pi_1, \langle \pi_2, \pi_1 \cdot \pi_2 \rangle \rangle \cdot f)) \\
&= g \cdot \mathrm{cur}'_X(\mathrm{cur}_{X \times B'}(\langle \pi_1 \cdot \pi_1, \langle \pi_2, \pi_1 \cdot \pi_2 \rangle \rangle \cdot f)),
\end{aligned}$$

which concludes the proof. □

Now, we can show that giving an algebraic theory a λ-theory structure amounts to endowing its corresponding Lawvere theory (Lemma 70) with a weak cartesian closed structure.

Lemma 79. *Let L be an algebraic theory. A weak cartesian closed category structure on its associated Lawvere theory gives a λ-theory structure on L.*

Proof. (**TODO**) □

Lemma 80. *Let L be a λ-theory. Then its corresponding Lawvere theory C is a weak cartesian closed category.*

Proof. Note that by definition of a Lawvere theory, every $n : C$ is the n-fold product of 1. Therefore, binary products are given by addition on the natural numbers, and 0 is the terminal object.

(**TODO**) □

Index