

Categorical semantics of the reduction of GATs to two-sorted GATs.

Notes on my 4.5-month internship at the Laboratoire d'Informatique de l'École Polytechnique
(Palaiseau, France)

Samy Avrillon, supervised by
Ambroise Lafont (LIX, Palaiseau, France)

General Context

Generalized Algebraic Theories a.k.a. GATs are syntactic objects introduced in 1986 by Cartmell [CartmellGATs]. We can see them as a generalization of Type Theory's inductive types as they enable us to describe algebraic structures. For example, we can describe the models of first order logic using a GAT.

A GAT is made of a list of “sorts” called a “sort specification” that describes sets, usually followed by a list of “constructors”.

Problem Studied

There is a process that transforms any GAT into another GAT with only two sorts. This transformation has been used by Filippo Sestini in their PhD, to restrict their study to GATs with only two sorts. However, this systematic transformation has not yet been formally justified.

Proposed Contribution

During this internship, my goal was to formalize this transformation semantically. I focused directly on the categories of models without looking at the syntactic aspects of GATs. I only focused on “sort specification”, i.e. GATs that are just a list of sort declarations with no constructors. I started with applying the transformation on simple examples to identify properties that would prove the transformation to be correct, working in a categorical framework. Then, I formally stated those properties and proved them as generically as possible.

Arguments Supporting Their Validity

The overall proof was made semantically and at each step, we tried to generalize objects as much as possible. The ways of constructing the objects are based on published papers [Altenkirch2018] [Fiore2008].

My contribution is also a first step towards solving the conjecture stated in Sestini's PhD.

Summary and Future Work

We formalized semantically this transformation, enabling us to be more general and avoid the syntactical aspects of GATs. This transformation has been proven so that studying only GATs with two sorts can be done without losing generality.

The next step could be to add term constructors to our formalization. We could describe their transformation and adapt the properties we stated for sort specification to them.

We could also try to further generalize GATs using this transformation. We could for example try to describe sorts that are mutually included one into the other. More work needs to be done studying them and looking at how the transformation affects them to discover new properties.

We could also aim to encode this transformation and the associated proofs in a proof assistant like Agda or Coq.

Contents

1	Introduction	3
2	Examples	4
2.1	The empty sort specification	5
2.2	Type theory example	6
3	Constructing the coreflection	10
3.1	Preliminaries	11
3.2	Initialization	13
3.3	Inductive Step: Constructing the categories	13
3.4	Inductive step: Constructing the functors F_i and G_i	14
3.5	Inductive Step: Proof of the coreflection	16
3.6	Inductive step: Constructing \triangleleft^i	16
4	Other Properties	17
4.1	Sort specifications as simple direct categories	17
4.2	Creating semantics object from the syntax	18
5	Summary	20
6	Bibliography	20
	Appendices	21

1 Introduction

A Generalized Algebraic Theory (or GAT), first introduced by Cartmell [**CartmellGATs**], is a syntactic specification of an algebraic structure. From a GAT, one can define a category of models describing the models of the algebraic structure. A GAT typically starts with a sort specification i.e. a list of sort declarations, which may be followed by a list of constructors. In this document, we will not ask ourselves about the specific syntax of GATs. A “vague” definition is enough to understand the rest of the document.

Sort specification A sort specification is a list of *sort declarations*, composed of *parameters* and *Set* as their codomain.

We give an example of a sort specification for Type Theory. In the right column, we give the interpretation of the sort declarations as components of the models of the GAT.

$\text{Con} : \text{Set}$	A set of contexts X_{Con}
$\text{Ty} : (\Gamma : \text{Con}) \rightarrow \text{Set}$	For each context $\Gamma \in X_{\text{Con}}$, a set $X_{\text{Ty}}(\Gamma)$ of types in this context
$\text{Tm} : (\Delta : \text{Con}) \rightarrow (A : \text{Ty } \Delta) \rightarrow \text{Set}$	For each context $\Gamma \in X_{\text{Con}}$ and each type $A \in X_{\text{Ty}}(\Gamma)$ in this context, a set $X_{\text{Tm}}(\Gamma, A)$ of terms of this type.

A sort specification therefore specifies the different families of sets contained in a model, and how they relate to each other in terms of indexing.

Term specification Once we have a sort specification, we can add constructors to it to make a complete GAT. Constructors are composed of parameters (the same kind as for sort declarations) and of a codomain which is a sort defined by a previous sort declaration. Those constructors specify elements of the sets contained in the model. For example, to the previous sort specification, one can add the following constructors:

$\text{unit} : (\Gamma : \text{Con}) \rightarrow \text{Ty } \Gamma$	In any context $\Gamma \in X_{\text{Con}}$, a type $\text{unit}(\Gamma) \in X_{\text{Ty}}(\Gamma)$.
$\text{tt} : (\Gamma : \text{Con}) \rightarrow \text{Tm } \Gamma (\text{unit}(\Gamma))$	In any context $\Gamma \in X_{\text{Con}}$, an element $\text{tt}(\Gamma) \in X_{\text{Tm}}(\Gamma, \text{unit}(\Gamma))$.

Two-sortification It was observed [**AmbrusSzumiXie2sort**] that one can transform any GAT into a GAT with only two sorts. Let us present this transformation.

The sort specification of the transformed GAT is always the same, and contains these two sort declarations:

$\mathcal{U} : \text{Set}$	The set of sorts
$\mathcal{E}l : \mathcal{O} \rightarrow \text{Set}$	For every sort object o in the set of sorts, a set called $\mathcal{E}l(o)$ of objects corresponding to the sort object o .

Then, we replace all occurrences of *Set* to \mathcal{O} , and we apply $\mathcal{E}l$ to every parameter. We write \underline{o} rather than $\mathcal{E}l(o)$ to ease reading. For example, the GAT of Type Theory presented above becomes that which follows:

$\text{Con} : \mathcal{O}$ $\text{Ty} : (\Gamma : \underline{\text{Con}}) \rightarrow \mathcal{O}$ $\text{Tm} : (\Delta : \underline{\text{Con}}) \rightarrow (A : \underline{\text{Ty } \Delta}) \rightarrow \mathcal{O}$ $\text{unit} : (\Gamma : \underline{\text{Con}}) \rightarrow \underline{\text{Ty } \Gamma}$ $\text{tt} : (\Gamma : \underline{\text{Con}}) \rightarrow \underline{\text{Tm (unit } \Gamma)}$	<p>One sort object is called “Con”</p> <p>For each object Γ corresponding to the sort object Con, another sort object called “Ty Γ”</p> <p>For each object Δ corresponding to the sort object Con, and for every object A corresponding to the sort object Ty Δ, another sort object called “Tm ΔA”</p> <p>For each object Γ corresponding to the sort object Con, an object called “unit Γ” corresponding to the sort object Ty Γ</p> <p>...</p>
---	---

This process is useful when studying GATs, as it enables one to restrict the scope of their study to GATs with only two sorts without loss of generality. Filippo Sestini noticed that in his thesis [SestiniPhD]:

Many instances of multi-sorted IITs [IITs are a variant of GATs] can be reduced to equivalent two-sorted IITs, via a systematic reduction method originally observed by Zongpu (Szumi) Xie. We are not aware of a formal proof of this construction for arbitrary IITs, but we conjecture that it does apply to all instances of induction-induction and consequently that it shows two-sorted IITs are enough to represent any specifiable IIT.

Goal The goal of my internship was to study and understand the relationship between the categories of models of an original GAT and the category of models of the transformed “two-sortified” GAT, to legitimate this transformation. In this document, we will only study sort specifications (i.e. lists of sort declarations, with no constructors).

We constructed a coreflection between those two categories, whose formal definition is given in section 3. It consists of an adjunction $F \dashv G$ between the category \mathcal{C} of the models of the GAT and the category \mathcal{B} of the models of the two-sortified GAT, where G is full and faithful. This coreflection justifies the transformation as the initial object of \mathcal{C} can be computed as the image by F of the initial object of \mathcal{B} .

The category \mathcal{B} is equipped with a forgetful functor R to \mathcal{B}_0 , the category of models of the two-sort sort specification $(\mathcal{O}, \mathcal{E}1)$.

$$\begin{array}{ccc}
 & & G \\
 & \swarrow & \searrow \\
 \mathcal{B} & \xleftrightarrow{\perp} & \mathcal{C} \\
 & \nwarrow & \nearrow \\
 & & F \\
 R \downarrow & & \\
 \mathcal{B}_0 & &
 \end{array}$$

2 Examples

Before making the formal proof, we will study our construction on two examples, in order to ease understanding of the formal proof.

Structure of the proof The formal proof will be an induction on the number of sorts taken into account. At each step, we will add a new sort declaration, represented by a functor H_i described later (subsection 3.3).

At the i th recursion step, we will build the following objects :

- The category of models of the GAT \mathcal{C}_i
- The category of models of the transformed GAT \mathcal{B}_i

- A functor $F_i : \mathcal{B}_i \rightarrow \mathcal{C}_i$
- A functor $G_i : \mathcal{C}_i \rightarrow \mathcal{B}_i$
- An adjunction between them $F_i \vdash G_i$
- A proof that $F_i G_i \cong \text{Id}_{\mathcal{C}_i}$ (i.e. $F_i \vdash G_i$ make up a coreflection)

We will now present those objects on two specific examples to help get an understanding of the logic behind the formal proof.

2.1 The empty sort specification

Our first example is the empty sort specification. As our proof will be an induction on the number of sorts, its first step will always be creating the objects for this empty sort specification.

Category of models of the empty sort specification The category of models of the empty sort specification will simply be $\mathbf{1}$, the category with only one object \star and one trivial (identity) morphism (i.e. the terminal category of $\mathcal{C}\text{at}$). This is because the empty sort specification has only one model. We will denote this category as \mathcal{C}_0 .

Category of models of the transformed GAT The transformed GAT for the empty sort specification is the sort specification $(\mathcal{O}, \mathcal{E}\mathbf{1})$ (no constructors as there is no sort in the initial sort specification):

$$\boxed{\begin{array}{l} \mathcal{U} : \text{Set} \\ \mathcal{E}\mathbf{1} : \mathcal{O} \rightarrow \text{Set} \end{array}}$$

The usual way of defining the category of models of this sort specification is by taking the category of families of sets $\mathcal{F}\text{amSet}$, whose objects are pairs of a set X and a family of sets indexed by X : $(Y_x)_{x \in X}$. However, we will rather use another category : $[\mathcal{T}, \text{Set}]$, the category of presheaves over the category \mathcal{T} , the category with two objects and one non-trivial arrow between them. The objects and morphism of \mathcal{T} are described below:

$$\begin{array}{c} \mathcal{E}\mathbf{1} \\ \downarrow p \\ \mathcal{U} \end{array}$$

So an object of $[\mathcal{T}, \text{Set}]$ is composed of two sets $X_{\mathcal{U}}$ and $X_{\mathcal{E}\mathbf{1}}$ and an arrow $X_p : X_{\mathcal{E}\mathbf{1}} \rightarrow X_{\mathcal{U}}$. Morphisms are natural transformations $f : X \rightarrow X'$ i.e. two functions $f_{\mathcal{U}} : X_{\mathcal{U}} \rightarrow X'_{\mathcal{U}}$ and $f_{\mathcal{E}\mathbf{1}} : X_{\mathcal{E}\mathbf{1}} \rightarrow X'_{\mathcal{E}\mathbf{1}}$ that commutes with X_p and X'_p .

This category is equivalent to the category $\mathcal{F}\text{amSet}$ with an equivalence we will see later ([subsection 3.1](#))

With this formalization, a model of this sort specification is a functor $X : \mathcal{T} \rightarrow \text{Set}$, such that

- $X_{\mathcal{U}}$ is the set of the “sort objects”
- For each sort object $A \in X_{\mathcal{U}}$, the set of objects corresponding to the sort object is $X_p^{-1}(\{A\}) \subset X_{\mathcal{E}\mathbf{1}}$

We will denote this category of models of the empty sort specification as \mathcal{B}_0 .

Functors We want to show that there is a coreflection between those two categories \mathcal{B}_0 and \mathcal{C}_0 as follows:

$$[\mathcal{T}, \text{Set}] = \mathcal{B}_0 \begin{array}{c} \xleftarrow{G_0} \\ \perp \\ \xrightarrow{F_0} \end{array} \mathcal{C}_0 = \mathbf{1}$$

$F_0 : [\mathcal{T}, \mathbf{Set}] \rightarrow \mathbf{1}$ is simply the terminal functor of \mathbf{Cat} , i.e. the functor sending all objects to \star , the only object of $\mathbf{1}$, and all morphisms to id_\star .

$G_0 : \mathbf{1} \rightarrow [\mathcal{T}, \mathbf{Set}]$ is the functor that sends the only object of $\mathbf{1}$ to the initial object of $[\mathcal{T}, \mathbf{Set}]$: $0_{[\mathcal{T}, \mathbf{Set}]}$ (i.e. such that G_0 preserves colimits)

We can check that those two functors create an adjunction, as

$$\mathcal{H}om(G_0\star, X) = \mathcal{H}om(0_{[\mathcal{T}, \mathbf{Set}]}, X) \cong 1_{\mathbf{Set}} \cong \mathcal{H}om(\star, F_0X)$$

We also have that they create a coreflection, as F_0G_0 goes from $\mathbf{1}$ to $\mathbf{1}$, and so $F_0G_0 = \text{Id}_\mathbf{1}$ as $\mathbf{1}$ is terminal in \mathbf{Cat} . So F_0G_0 is an isofunctor.

2.2 Type theory example

Our next example is the sort specification of type theory presented in the introduction. We recall that the sort specification and its transformation are as follows:

	$\mathcal{U} : \mathbf{Set}$
	$\mathcal{E}l : \mathcal{O} \rightarrow \mathbf{Set}$
$\text{Con} : \mathbf{Set}$	$\text{Con} : \mathcal{O}$
$\text{Ty} : (\Gamma : \text{Con}) \rightarrow \mathbf{Set}$	$\text{Ty} : (\Gamma : \underline{\text{Con}}) \rightarrow \mathcal{O}$
$\text{Tm} : (\Delta : \text{Con}) \rightarrow (A : \text{Ty } \Delta) \rightarrow \mathbf{Set}$	$\text{Tm} : (\Delta : \underline{\text{Con}}) \rightarrow (A : \text{Ty } \Delta) \rightarrow \mathcal{O}$

We will build the following objects, step by step, adding one sort at a time:

$$\begin{array}{ccc}
 \mathcal{B}_3 & \begin{array}{c} \xleftarrow{G_3} \\ \perp \\ \xrightarrow{F_3} \end{array} & \mathcal{C}_3 \\
 \downarrow R_2^3 & & \\
 \mathcal{B}_2 & \begin{array}{c} \xleftarrow{G_2} \\ \perp \\ \xrightarrow{F_2} \end{array} & \mathcal{C}_2 \\
 \downarrow R_1^2 & & \\
 \mathcal{B}_1 & \begin{array}{c} \xleftarrow{G_1} \\ \perp \\ \xrightarrow{F_1} \end{array} & \mathcal{C}_1 \\
 \downarrow R_0^1 & & \\
 \mathcal{B}_0 & \begin{array}{c} \xleftarrow{G_0} \\ \perp \\ \xrightarrow{F_0} \end{array} & \mathcal{C}_0
 \end{array}$$

Where the categories \mathcal{C}_i are the categories of models of the sort specification limited to the i first sort declarations, and \mathcal{B}_i are the categories of models of the transformed GAT limited to the i first sort declarations. For example, \mathcal{C}_2 is the category of models of the two first sorts of the sort specification (Con, Ty) . $\mathcal{B}_0, \mathcal{C}_0, F_0$, and G_0 have been defined in the last subsection.

The R_{i-1}^i functors between the categories are forgetful functors that forget about the added sorts.

I will only give here the explanation of functors F_3 and G_3 , omitting the intermediate functors F_1, G_1, F_2, G_2 . Their construction is given in the complete proof.

Category of models We have seen in the introduction a description of a model of the type theory sort specification. We will rebuild the category of those models inductively, adding one sort at a time.

To construct those categories generically, we will define functors $H_i : \mathcal{C}_{i-1} \rightarrow \mathbf{Set}$ that will describe the “parameters” of the sort declaration.

$$\boxed{\text{Con} : \mathbf{Set}}$$

We begin with the following functor, corresponding to the sort declaration above

$$H_{\text{Con}}(\star) = 1 \in \mathbf{Set}$$

This functor means that Con takes no parameter (or one parameter of type unit) i.e. there is only *one* way of constructing a Con .

We build \mathcal{C}_1 to be a pair of

- An object X_0 of \mathcal{C}_0
- A family of sets X_{Con} indexed by $H_{\text{Con}}(x) = 1$

As there is only one object in \mathcal{C}_0 and as a family of sets indexed by 1 is simply a set, we have that this category \mathcal{C}_1 is isomorphic to Set (i.e. the category of sets). This is as expected: a model X consists of a set X_{Con} .

$$\boxed{\text{Ty} : (\Gamma : \text{Con}) \rightarrow \text{Set}}$$

Then, we take the functor $H_2(X) = X_{\text{Con}}$, corresponding to the sort declaration above. This functor means that Ty takes one parameter of type Con i.e. There is exactly one Ty sort for every Con there is in the model.

Again, we build \mathcal{C}_{Ty} to be a pair of

- A model X_{Con} from \mathcal{C}_1
- A family of sets X_{Ty} indexed by $H_{\text{Ty}}(X_{\text{Con}}) = X_{\text{Con}}$

This category is, as expected, the category \mathcal{FamSet} of families of sets i.e. objects are pairs of a set X_{Con} and a family indexed by X_{Con} : $(X_{\text{Ty}}(\Gamma))_{\Gamma \in X_{\text{Con}}}$.

$$\boxed{\text{Tm} : (\Delta : \text{Con}) \rightarrow (A : \text{Ty } \Delta) \rightarrow \text{Set}}$$

The functor $\mathcal{C}_2 \rightarrow \text{Set}$ we will use here is defined as follows:

$$H_{\text{Tm}}(X_{\text{Con}}, X_{\text{Ty}}) = \coprod_{\Delta \in X_{\text{Con}}} X_{\text{Ty}}(\Delta) \in \text{Set}$$

\coprod denotes the coproduct or disjoint union in Set . This functor means that Tm takes one parameter Δ of sort Con and one parameter of sort $\text{Ty } \Delta$, i.e. there is exactly one Tm for every pair of a Con called Δ in the model, and of a Ty associated to Δ in the model.

We build \mathcal{C}_3 to be a pair of

- A model $(X_{\text{Con}}, X_{\text{Ty}})$ from \mathcal{C}_2
- A family of sets X_{Tm} indexed by $H_{\text{Tm}}(\text{Con})$

This category is isomorphic to the category composed of triples of

- A set X_{Con}
- A family of sets indexed by X_{Con} : $(X_{\text{Ty}}(\Gamma))_{\Gamma \in X_{\text{Con}}}$
- A family of sets indexed by $(\Delta : X_{\text{Con}})$ and $X_{\text{Ty}}(\Delta)$: $\left((X_{\text{Tm}}(\Delta, A))_{A \in X_{\text{Ty}}(\Delta)} \right)_{\Delta \in X_{\text{Con}}}$

Category of models of the transformed GAT We have also seen in the introduction how was constructed the category of models of the transformed GAT. We will reconstruct it sort by sort.

$$\boxed{\begin{array}{l} \mathcal{O} : \text{Set} \\ \mathcal{E}l : \mathcal{O} \rightarrow \text{Set} \end{array}}$$

We have seen in the previous subsection that the category of models of this GAT is $\mathcal{B}_0 = [\mathcal{T}, \text{Set}]$. We will only add new *constructors* to these models, and no new sorts.

$$\boxed{\text{Con} : \mathcal{O}}$$

We are adding a constructor to the model. This constructor takes no parameters, which means we want to select one sort out of our model to be the Con sort. And we recall that the sorts of a model $X : [\mathcal{T}, \text{Set}]$ are the elements of $X_{\mathcal{U}}$.

Therefore a model of the above GAT with only one constructor is a model $X : [\mathcal{T}, \text{Set}]$ along with an object $\text{Cstr}_{\text{Con}}^X \in X_{\mathcal{U}}$. An object of \mathcal{B}_1 is a pair of

- A functor $X : \mathcal{T} \rightarrow \text{Set}$
- An element $\text{Cstr}_{\text{Con}}^X \in X_{\mathcal{U}}$

A morphism f from $(X, \text{Cstr}_{\text{Con}}^X)$ to $(X', \text{Cstr}_{\text{Con}}^{X'})$ is simply a morphism of $[\mathcal{T}, \text{Set}]$ from X to X' that respects the constructor, i.e. that is such that

$$f_{\mathcal{U}}(\text{Cstr}_{\text{Con}}^X) = \text{Cstr}_{\text{Con}}^{X'}$$

$$\boxed{\text{Ty} : \text{Con} \rightarrow \mathcal{O}}$$

Now the constructor we are adding takes one parameter. It means that for every element of \mathcal{E}_1 associated with Con, there is an object in \mathcal{O} .

This constructor means that a model X comes with a function as follows:

$$\text{Cstr}_{\text{Ty}}^X : X_p^{-1}(\{\text{Cstr}_{\text{Con}}^X\}) \rightarrow X_{\mathcal{U}}$$

The codomain of this constructor means “Every object of $X_{\mathcal{E}_1}$ that is associated to the Con object of $X_{\mathcal{U}}$ i.e. the object $\text{Cstr}_{\text{Con}}^X$ ” This codomain is quite verbose, but we have another way of expressing it. In the final proof, we will have already constructed the functor $F_1 : \mathcal{B}_1 \rightarrow \mathcal{C}_1$, and we had a functor $H_{\text{Ty}} : \mathcal{C}_1 \rightarrow \text{Set}$ that described the parameters of the sort declaration of Ty for the category \mathcal{C}_1 . It turns out that $H_{\text{Ty}} \circ F_1 : \mathcal{B}_1 \rightarrow \text{Set}$ describes exactly the codomain expressed above.

$$(H_{\text{Ty}} \circ F_1)(X, \text{Cstr}_{\text{Con}}^X) = X_p^{-1}(\text{Cstr}_{\text{Con}}^X)$$

The fact that $H_X \circ F_{X-1}$ describes exactly the codomain of the constructor we want to add will hold for every sort declaration we can add.

Therefore, the category \mathcal{B}_2 consists of a pair of

- A model $(X, \text{Cstr}_{\text{Con}}^X)$ from \mathcal{B}_1
- A constructor $\text{Cstr}_{\text{Ty}}^X : H_i F_{i-1}(X, \text{Cstr}_{\text{Con}}^X) \rightarrow X_{\mathcal{U}}$

Once again, a morphism f from $(X, \text{Cstr}_{\text{Con}}^X, \text{Cstr}_{\text{Ty}}^X)$ to $(X', \text{Cstr}_{\text{Con}}^{X'}, \text{Cstr}_{\text{Ty}}^{X'})$ is simply a morphism of \mathcal{B}_1 from $(X, \text{Cstr}_{\text{Con}}^X)$ to $(X', \text{Cstr}_{\text{Con}}^{X'})$ that does respect the Cstr_{Ty} constructor, i.e. that verifies, for every Γ in $H_i F_{i-1}(X, \text{Cstr}_{\text{Con}}^X)$ that

$$f_{\mathcal{U}}(\text{Cstr}_{\text{Ty}}^X(\Gamma)) = \text{Cstr}_{\text{Ty}}^{X'}(f_{\mathcal{E}_1}(\Gamma))$$

$$\boxed{\text{Tm} : (\Delta : \text{Con}) \rightarrow (A : \text{Ty } \Delta) \rightarrow \mathcal{O}}$$

This one is made like the others, a model of \mathcal{B}_3 is a pair of

- A model $(X, \text{Cstr}_{\text{Con}}^X, \text{Cstr}_{\text{Ty}}^X)$ of \mathcal{B}_2
- A constructor $\text{Cstr}_{\text{Tm}}^X$ that goes from $H_{\text{Tm}} F_2 X$ to $X_{\mathcal{U}}$

And morphisms are restricted with the Tm constructor rule that states that for every (Δ, A) in $H_i F_{i-1}(X, \text{Cstr}_{\text{Con}}^X, \text{Cstr}_{\text{Ty}}^X)$

$$f_{\mathcal{U}}(\text{Cstr}_{\text{Tm}}^X(\Delta, A)) = \text{Cstr}_{\text{Tm}}^{X'}(f_{\mathcal{E}_1}(\Delta), f_{\mathcal{E}_1}(A))$$

Constructing G_3 We want to make functors connecting those two categories

$Y : \mathcal{C}_3$	$X : \mathcal{B}_3$
$Y_{\text{Con}} : \mathcal{S}\text{et}$	$X : [\mathcal{T}, \mathcal{S}\text{et}]$
$(Y_{\text{Ty}}(\Gamma))_{\Gamma \in Y_{\text{Con}}}$	$\mathcal{C}\text{str}_{\text{Con}}^X : X_{\mathcal{U}}$
$((Y_{\text{Tm}}(\Delta, A))_{A \in Y_{\text{Ty}}(\Delta)})_{\Delta \in Y_{\text{Con}}}$	$\mathcal{C}\text{str}_{\text{Ty}}^X : H_{\text{Ty}}F_1(X, \mathcal{C}\text{str}_{\text{Con}}^X) \rightarrow X_{\mathcal{U}}$
	$\mathcal{C}\text{str}_{\text{Tm}}^X : H_{\text{Tm}}F_2(X, \mathcal{C}\text{str}_{\text{Con}}^X, \mathcal{C}\text{str}_{\text{Ty}}^X) \rightarrow X_{\mathcal{U}}$

Let's first look at the first component of the «result» $X = G_3(Y)$. It will be a functor of $[\mathcal{T}, \mathcal{S}\text{et}]$. We recall how $[\mathcal{T}, \mathcal{S}\text{et}]$ corresponds to the sort specification $(\mathcal{O}, \mathcal{E}1)$:

- $X_{\mathcal{U}}$ is the set of all sorts
- $X_{\mathcal{E}1}$ is the set of all objects of those sorts
- X_p sends an object of a sort to the sort it corresponds to

Therefore, we will construct $X_{\mathcal{U}}$ to be the disjoint union (denoted \oplus) of all the indices of the sets inside Y , $X_{\mathcal{E}1}$ will be the disjoint union of all the sets inside Y , and X_p will keep track of what is the index of a set.

$$\begin{array}{ccccccc}
 X_{\mathcal{E}1} & = & Y_{\text{Con}} & \oplus & \coprod_{\Gamma \in Y_{\text{Con}}} Y_{\text{Ty}}(\Gamma) & \oplus & \coprod_{\Delta \in Y_{\text{Con}}} \coprod_{A \in Y_{\text{Ty}}(\Delta)} Y_{\text{Ty}}(\Delta, A) \\
 \downarrow p & & \downarrow & & \downarrow \text{proj}_1 & & \downarrow \text{proj}_{2,1} \\
 X_{\mathcal{U}} & = & 1 & \oplus & Y_{\text{Con}} & \oplus & \coprod_{\Delta \in Y_{\text{Con}}} Y_{\text{Ty}}(\Delta)
 \end{array}$$

where $\text{proj}_1 : \coprod_{a \in A} B(a) \rightarrow A$ is the first projector of the coproduct, and $\text{proj}_{2,1}$ get the index $\coprod_{a \in A} B(a)$ of a double coproduct $\coprod_{a \in A} \coprod_{b \in B(a)} C(a, b)$

Now, the constructors only have to give the right sort according to their arguments. So

$$\begin{aligned}
 \mathcal{C}\text{str}_{\text{Con}}^X &= \text{inj}_1(\star) \\
 \mathcal{C}\text{str}_{\text{Ty}}^X(\Gamma) &= \text{inj}_2(\Gamma) \\
 \mathcal{C}\text{str}_{\text{Tm}}^X(\Delta, A) &= \text{inj}_3(\Delta, A)
 \end{aligned}$$

where inj_i is the i th injector of the disjoint union \oplus .

Remark 1 *With this functor, every element of $X_{\mathcal{U}}$ is reached by some constructor. Therefore, every object of $X_{\mathcal{E}1}$ points to a sort that is reached by some constructor.*

Constructing F_i We will now create an object $Y = F_i X$ of \mathcal{C}_3 from an object X of \mathcal{B}_3 .

- Y_{Con} should be the set of all objects of sort Con. In the transformed GAT, it means all objects of $\mathcal{E}1$ which have been created with the sort $\mathcal{C}\text{str}_{\text{Con}}^X$. Therefore we have

$$Y_{\text{Con}} = X_p^{-1}(\{\mathcal{C}\text{str}_{\text{Con}}^X\}) \subseteq X_{\mathcal{E}1}$$

- $Y_{\text{Ty}}(\Gamma)$ should be the set of all objects of sort Ty Γ , which means all objects of $\mathcal{E}1$ which have been created with the sort $\mathcal{C}\text{str}_{\text{Ty}}^X(\Gamma)$. Therefore we have, for every Γ in Y_{Con}

$$Y_{\text{Ty}}(\Gamma) = X_p^{-1}(\{\mathcal{C}\text{str}_{\text{Ty}}^X(\Gamma)\})$$

This is well-defined, as

$$\Gamma : Y_{\text{Con}} = X(p)^{-1}(\{\mathcal{C}\text{str}_{\text{Con}}^X\}) = H_{\text{Ty}}(X(p)^{-1}(\{\mathcal{C}\text{str}_{\text{Con}}^X\})) = H_{\text{Ty}}F_1(X, \mathcal{C}\text{str}_{\text{Con}}^X)$$

- In the same way, we define

$$Y_{\text{Tm}}(\Delta, A) = X_p^{-1}(\mathcal{C}\text{str}_{\text{Tm}}^X(\Delta, A))$$

which is also well-defined

Remark 2 We can see that the F_3 functor does not take into account elements of $X_{\mathcal{U}}$ that are not “created” by one of the constructors, nor does it take into account the elements of $X_{\mathcal{E}_1}$ that are associated with those elements of $X_{\mathcal{U}}$.

This will later justify that when one adds elements to the base object X of a model of \mathcal{B}_i , then the image model by F_i is unchanged, which we will later express by “ $F_i \text{inj}_i^i$ is an isomorphism” in [the global proof](#).

An adjunction I will not do the proof that $F_3 \vdash G_3$ in this specific case, as it will be proven in the general case in the global proof, but I will give intuitions of why the adjunction holds.

An adjunction would be described by the following isomorphism:

$$\mathcal{H}\text{om}_{\mathcal{B}_3}(G_3Y, X) \simeq \mathcal{H}\text{om}_{\mathcal{C}_3}(Y, F_3X)$$

Recall that sorts of G_3Y are all in the domain of some constructor, so therefore, a morphism of $\mathcal{H}\text{om}_{\mathcal{B}_3}(G_3Y, X)$ only acts on constructible sorts. Also, the only sorts that remain in F_3X are the ones that were reached by some constructor of X , therefore a morphism of $\mathcal{H}\text{om}_{\mathcal{C}_3}(Y, F_3X)$ only acts on constructible sorts.

Therefore, both morphisms are only descriptions of actions on constructible sorts into constructible sorts. That’s how we create the isomorphism.

A coreflection If we look at F_3G_3Y for a model Y of \mathcal{C}_3 , we remark that

$$\begin{aligned} F_3G_3(Y)_{\text{Con}} &= G_3(Y)_p^{-1}(\{\mathcal{C}\text{str}_{\text{Con}}^{G_3(Y)}\}) \\ &= G_3(Y)_p^{-1}(\{\text{inj}_1 \star\}) \\ &= Y_{\text{Con}} \end{aligned}$$

and

$$\begin{aligned} F_3G_3(Y)_{\text{Ty}}(\Gamma) &= G_3(Y)_p^{-1}(\{\mathcal{C}\text{str}_{\text{Ty}}^{G_3(Y)}(\Gamma)\}) \\ &= G_3(Y)_p^{-1}(\{\text{inj}_2 \Gamma\}) \\ &= \text{proj}_1^{-1}(\Gamma) \\ &= \left\{ (\Gamma', A) \in \coprod_{\Gamma' \in Y_{\text{Con}}} Y_{\text{Ty}}(\Gamma') \mid \Gamma' = \Gamma \right\} \\ &\simeq Y_{\text{Ty}}(\Gamma) \end{aligned}$$

and finally, with the same method, we get that

$$F_3G_3(Y)_{\text{Tm}}(\Delta, A) \simeq Y_{\text{Tm}}(\Delta, A)$$

And therefore $F_3G_3 \cong \text{Id}_{\mathcal{C}_3}$.

3 Constructing the coreflection

The proof will take the form of an induction on the number of sorts taken into account. At each induction step, we will build the following objects:

- The category of models of the GAT \mathcal{C}_i
- The category of models of the transformed GAT \mathcal{B}_i

- A forgetful functor $R_{i-1}^i : \mathcal{B}_i \rightarrow \mathcal{B}_{i-1}$.

Those functors compose themselves into a functor $R_0^i : \mathcal{B}_i \rightarrow \mathcal{B}_0$:

$$R_0^i := R_0^1 \circ \dots \circ R_{i-1}^i$$

- An operator $\triangleleft^i : \mathcal{B}_i \times \mathcal{B}_0 \rightarrow \mathcal{B}_i$ along with two morphisms $\text{inj}_1^i : X \rightarrow X \triangleleft^i Y$ and $\text{inj}_2^i : Y \rightarrow R_0^i(X \triangleleft^i Y)$ for every X, Y in $\mathcal{B}_i \times \mathcal{B}_0$. This operation follows a specific universal property: For every morphism $g : X \rightarrow Z$ and $h : Y \rightarrow R_0^i Z$, there is a unique morphism $\{g; h\}$ such that the following two diagrams respectively in \mathcal{B}_i and \mathcal{B}_0 commute:

$$\begin{array}{ccc} X & \xrightarrow{\text{inj}_1^i} & X \triangleleft^i Y \\ & \searrow g & \downarrow \{g, h\} \\ & & Z \end{array} \quad \begin{array}{ccc} R_0^i X & \xrightarrow{R_0^i \text{inj}_1^i} & R_0^i(X \triangleleft^i Y) \xleftarrow{\text{inj}_2^i} Y \\ & \searrow R_0^i g & \downarrow R_0^i \{g, h\} \\ & & R_0^i Z \end{array}$$

The operator can be extended to act on morphisms by exploiting the universal property, so that for any morphism $g : X \rightarrow X'$ in \mathcal{B}_i and $h : Y \rightarrow Y'$ in \mathcal{B}_0 , there is a morphism $g \triangleleft^i h : X \triangleleft^i Y \rightarrow X' \triangleleft^i Y'$ in \mathcal{B}_i .

- A functor $F_i : \mathcal{B}_i \rightarrow \mathcal{C}_i$
- A functor $G_i : \mathcal{C}_i \rightarrow \mathcal{B}_i$
- An adjunction between them $F_i \vdash G_i$
- A proof that $F_i G_i \cong \text{Id}_{\mathcal{C}_i}$ (i.e. $F_i \vdash G_i$ make up a coreflection)
- A proof that $F_i \text{inj}_1^i$ is an isomorphism. The inverse isomorphism will be denoted as $(F_i \text{inj}_1^i)^{-1}$
- A proof that the canonical morphism $\mathfrak{H}_{i-1}^i : (R_{i-1}^i X) \triangleleft^{i-1} Y \rightarrow R_{i-1}^i(X \triangleleft^i Y)$ is an isomorphism.

We will denote the composition of those \mathfrak{H} morphisms as such:

$$\mathfrak{H}_0^i := R_0^{i-1} \mathfrak{H}_{i-1}^i \circ \dots \circ R_0^1 \mathfrak{H}_1^1 \circ \mathfrak{H}_0^1 : (R_0^i X) \triangleleft^0 Y \rightarrow R_0^i(X \triangleleft^i Y)$$

Here is a figure that describes the recursive construction of some of the above objects

$$\begin{array}{ccc} & & \mathcal{C}_i \\ & \swarrow G_i & \\ \mathcal{B}_i & \xrightarrow{\perp} & \mathcal{C}_i \\ & \searrow F_i & \\ & & \mathcal{C}_{i-1} \\ & \swarrow G_{i-1} & \\ \mathcal{B}_{i-1} & \xrightarrow{\perp} & \mathcal{C}_{i-1} \\ & \searrow F_{i-1} & \\ & & \vdots \\ & & \mathcal{C}_0 \\ & \swarrow G_0 & \\ \mathcal{B}_0 & \xrightarrow{\perp} & \mathcal{C}_0 \\ & \searrow F_0 & \end{array}$$

R_0^i (curved arrow from \mathcal{B}_i to \mathcal{B}_0)

3.1 Preliminaries

Grothendieck construction For a category \mathcal{C} and a functor $F : \mathcal{C} \rightarrow \text{Cat}$, the Grothendieck construction of F is a category whose objects are pairs of

- X an object of \mathcal{C}
- an object of $F(X)$

A morphism $(X, Y) \rightarrow (X', Y')$ is a pair of a morphism $f : X \rightarrow X'$ in \mathcal{C} and a morphism $g : F(f)(Y) \rightarrow Y'$ in $H(X')$.

We will denote this category $(X : \mathcal{C}) \times F(X)$ as its objects are pairs.

Slice category For a category \mathcal{C} and X an object in that category, the slice category (or over category) \mathcal{C}/X is a category whose objects are pairs of

- Y an object of \mathcal{C}
- an arrow $Y \rightarrow X$ of \mathcal{C}

A morphism $(Y, f) \rightarrow (Y', f')$ is a morphism $g : Y \rightarrow Y'$ such that $f' \circ g = f$.

We can define a functor $(\mathcal{C}/-) : \mathcal{C} \rightarrow \mathbf{Cat}$ from this construction.

We will often concatenate the two methods above to create from a category \mathcal{C} and a functor $H : \mathcal{C} \rightarrow \mathbf{Set}$ a new category $(X : \mathcal{C}) \times (\mathbf{Set}/H(X))$.

Slice category over a set When the category \mathcal{C} is \mathbf{Set} , we have the equivalence

$$\mathbf{Set}/X \simeq \mathbf{Set}^X$$

This equivalence is described by the following correspondence of objects. An object (Y, f) of \mathbf{Set}/X is transformed into the family of sets $A : X \rightarrow \mathbf{Set}$ defined by

$$A(x) := f^{-1}(\{x\}) = \text{the preimage by } f \text{ of the singleton } \{x\}$$

Conversely, a family of sets $A : X \rightarrow \mathbf{Set}$ is transformed into the following object of \mathbf{Set}/X

$$\coprod_{x \in X} A(x) \xrightarrow{\pi_1} X$$

where \coprod is the coproduct of \mathbf{Set} and π_1 is the first projection of the coproduct.

Morphisms are translated using the same logic.

This equivalence justifies that we used $[\mathcal{T}, \mathbf{Set}]$, the category of presheaves over \mathcal{T} instead of \mathcal{FamSet} to describe the category of models of the sort specification $(\mathcal{O}, \mathcal{E}1)$ in [subsection 2.1](#).

We recall that the category \mathcal{T} is as follows

$$\begin{array}{c} \mathcal{E}1 \\ \downarrow p \\ \mathcal{U} \end{array}$$

This category is equivalent to the category \mathcal{FamSet} , as $(X : \mathbf{Set}) \times \mathbf{Set}^X$ is equivalent to $(X : \mathbf{Set}) \times (\mathbf{Set}/X)$ (as seen above), which is isomorphic to $[\mathcal{T}, \mathbf{Set}]$ (both categories consists of two sets and one arrow between them).

The categories of models of the transformed GATs will be built atop this category $\mathcal{B}_0 = [\mathcal{T}, \mathbf{Set}]$.

K functor We define a helper functor $K_A : (X : \mathcal{C}) \times (\mathbf{Set}/A(X)) \rightarrow [\mathcal{T}, \mathbf{Set}]$ defined as follows

$$K_A(X, (Y, f)) = \left[\begin{array}{ccc} \mathcal{E}1 & \longmapsto & Y \\ p \downarrow & \longmapsto & \downarrow f \\ \mathcal{U} & \longmapsto & A(X) \end{array} \right]$$

The morphisms are translated as-is.

Remark 3 *This functor can be constructed using the formal construction of the Grothendieck construction of $[\mathcal{T}, \mathbf{Set}] \simeq (X : \mathbf{Set}) \times (\mathbf{Set}/X)$ as a pullback in the category of categories \mathbf{Cat}*

3.2 Initialization

In this subsection, we will recall objects built in the example of [The empty sort specification](#), and build some other objects corresponding to the empty sort specification i.e. the first step of our induction.

- \mathcal{C}_0 is $\mathbf{1}$, the category with only one object \star and one trivial morphism (i.e. the terminal category of $\mathcal{C}at$)
- \mathcal{B}_0 is $[\mathcal{T}, \mathcal{S}et]$, the category of models of the $(\mathcal{O}, \mathcal{E}l)$ sort specification.
- The universal property of \triangleleft^0 is that of the coproduct. Therefore, we will define $\triangleleft^0 : \mathcal{B}_0 \times \mathcal{B}_0 \rightarrow \mathcal{B}_0$ to be the coproduct \oplus of $[\mathcal{T}, \mathcal{S}et]$, with $\text{inj}_1^0 : X \rightarrow X \triangleleft^0 Y$ being the first injector of the coproduct and $\text{inj}_2^0 : Y \rightarrow X \triangleleft^0 Y$ being the second injector.
- $F_0 : [\mathcal{T}, \mathcal{S}et] \rightarrow \mathbf{1}$ is the terminal functor of $\mathcal{C}at$
- $G_0 : \mathbf{1} \rightarrow [\mathcal{T}, \mathcal{S}et]$ is the functor that sends the only object of $\mathbf{1}$ to the initial object of $[\mathcal{T}, \mathcal{S}et]$: $0_{[\mathcal{T}, \mathcal{S}et]}$
- For X an object of $[\mathcal{T}, \mathcal{S}et]$, we have

$$\mathcal{H}om(G_0\star, X) = \mathcal{H}om(0_{[\mathcal{T}, \mathcal{S}et]}, X) \cong 1_{\mathcal{S}et} \cong \mathcal{H}om(\star, F_0X)$$

(reminder: \star is the only object of $\mathbf{1}$) Therefore, we have that $F_0 \vdash G_0$.

- $F_0G_0 : \mathbf{1} \rightarrow \mathbf{1}$ so $F_0G_0 = \mathcal{I}d_{\mathbf{1}}$ as $\mathbf{1}$ is terminal in $\mathcal{C}at$
- $F_i \text{inj}_1^0 = \text{id}_{\star}$ which is an isomorphism

3.3 Inductive Step: Constructing the categories

In this part, we will show how we construct recursively both categories \mathcal{C}_i and \mathcal{B}_i , along with the functor $R_{i-1}^i : \mathcal{B}_i \rightarrow \mathcal{B}_{i-1}$. We will use the loop invariants that we expressed in the introduction of this section.

To construct the categories, we need some object describing the sorts we are adding to the categories. This object is a given functor $H_i : \mathcal{C}_{i-1} \rightarrow \mathcal{S}et$. Those functors are the same that Altenkirch and al. [[Altenkirch2018](#)] used in their paper to describe a specification of sorts. They are such that, for $X : \mathcal{C}_{i-1}$ a model, they describe the set H_iX of parameter range for the added sort. We gave an example of those functors in section [Examples](#) if you want to understand how they work. There is also a way of obtaining them directly from the syntax, described in section [Creating semantics object from the syntax](#). We suppose that those H_i functors are given.

3.3.1 Constructing \mathcal{C}_i

We construct the category \mathcal{C}_i as the following Grothendieck pair:

$$\mathcal{C}_i := (X : \mathcal{C}_{i-1}) \times (\mathcal{S}et/H_i(X))$$

and where H_i is the specific functor described above.

This way of constructing the category has formerly been described by Altenkirch and al. [[Altenkirch2018](#)], with the only difference of the equivalence $\mathcal{S}et/H_i(X) \simeq \mathcal{S}et^{H_i(X)}$.

3.3.2 Constructing \mathcal{B}_i

We construct the category \mathcal{B}_i as follows.

An object of \mathcal{B}_i is a pair of:

- an object X of \mathcal{B}_{i-1}
- a “sort constructor” $\mathcal{C}str^X$ as a function $H_iF_{i-1}X \rightarrow (R_0^{i-1}X)_{\mathcal{U}}$ where H_i is the functor $\mathcal{C}_{i-1} \rightarrow \mathcal{S}et$ described above and F_{i-1} is the right part of the adjunction $\mathcal{B}_{i-1} \rightarrow \mathcal{C}_{i-1}$ that we have defined at the previous induction step.

If we have an object X of \mathcal{B}_i , the first component is denoted as $R_{i-1}^i X$, which is an object of \mathcal{B}_{i-1} , and the second component is denoted as $\mathcal{C}\text{str}^X : R_{i-1}^i X \rightarrow (R_0^i X)\mathcal{U}$.

A morphism $X \rightarrow X'$ of \mathcal{B}_i is a morphism $f : R_{i-1}^i X \rightarrow R_{i-1}^i X'$ in \mathcal{B}_{i-1} such that the following diagram commutes.

$$\begin{array}{ccc} H_i F_{i-1} R_{i-1}^i X & \xrightarrow{\mathcal{C}\text{str}^X} & (R_0^i X)\mathcal{U} \\ H_i F_{i-1} f \downarrow & & \downarrow (R_0^{i-1} f)\mathcal{U} \\ H_i F_{i-1} R_{i-1}^i X' & \xrightarrow{\mathcal{C}\text{str}^{X'}} & (R_0^i X')\mathcal{U} \end{array}$$

For a morphism $f : X \rightarrow X'$ of \mathcal{B}_i , we denote as $R_{i-1}^i f$ the underlying morphism $R_{i-1}^i X \rightarrow R_{i-1}^i X'$.

Identities and compositions are that of the category \mathcal{B}_{i-1} , and categorical equalities are trivially derived from the diagram above.

R_{i-1}^i acts on objects and morphisms and induces a functor $\mathcal{B}_i \rightarrow \mathcal{B}_{i-1}$. The induced functor is monadic and therefore preserves limits.

3.4 Inductive step: Constructing the functors F_i and G_i

The adjunction $F_i \vdash G_i$ is built using the two functors from the adjunction $F_{i-1} \vdash G_{i-1}$ defined in the previous induction step. We use them to define the first part of the adjunction, and we compose them with two adjoint functors W and E that we will define in this section. The overall construction for this induction step is as follows:

$$\begin{array}{ccc} \mathcal{C}_i & \xlongequal{\quad} & (X : \mathcal{C}_{i-1}) \times \mathcal{S}\text{et}/H_i X \\ & \nearrow^{G_{i-1} \times (H_i \eta_{i-1} \circ -)} \left(\downarrow \right)^{F_{i-1} \times \text{id}} & \\ & & (X : \mathcal{B}_{i-1}) \times \mathcal{S}\text{et}/H_i F_{i-1} X \\ & \searrow_{G_i} \left(\downarrow \right)^{W} \left(\downarrow \right)^{E} & \\ & & \mathcal{B}_i \end{array}$$

Where

$$\begin{aligned} (F_{i-1} \times \text{id})(X, \underset{y:Y \rightarrow F_{i-1}(X)}{(Y, y)}) &= (F_{i-1} X, \underset{y:Y \rightarrow F_{i-1} X}{(Y, y)}) \\ (F_{i-1} \times \text{id})(f, g) &= (F_{i-1} f, g) \end{aligned}$$

and

$$\begin{aligned} (G_{i-1} \times (H_i \eta_{i-1} \circ -))(X, \underset{y:Y \rightarrow H_i X}{(Y, y)}) &= (G_{i-1} X, \underset{Y:Y \rightarrow H_i X \rightarrow H_i F_{i-1}(G_{i-1} X)}{(Y, H_i \eta_{i-1} \circ y)}) \\ (G_{i-1} \times (H_i \eta_{i-1} \circ -))(f, g) &= (G_{i-1} f, g) \end{aligned}$$

with $\eta_{i-1} : X \rightarrow F_{i-1} G_{i-1} X$ being the unit of the adjunction $F_{i-1} \vdash G_{i-1}$.

These equations define two functors, that create the following adjunction:

$$F_{i-1} \times \text{id} \vdash G_{i-1} \times (H_i \eta_{i-1} \circ -)$$

The unit of this adjunction is $\eta_{i-1} \times (H_i \eta_{i-1} \circ -)$ and its counit is $\varepsilon_{i-1} \times \text{id}$, where ε_{i-1} is the counit of the adjunction $F_{i-1} \vdash G_{i-1}$

3.4.1 W definition

We define a functor $W : ((X : \mathcal{B}_{i-1}) \times \mathcal{S}et/H_i F_{i-1} X) \rightarrow \mathcal{B}_i$

The action on objects is as follows:

$$W(X, Y) := \left(X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y), \widetilde{\text{inj}}_2 \right)$$

With $\widetilde{\text{inj}}_2$ being defined by the composition

$$\begin{array}{c} H_i F_{i-1}(X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y)) \\ \downarrow \simeq \\ H_i(F_{i-1} \text{inj}_2^{i-1})^{-1} \\ \downarrow \\ H_i F_{i-1} X \\ \parallel \\ (K_{H_i F_{i-1}}(X, Y))_{\mathcal{U}} \\ \downarrow \text{inj}_2^{i-1} \\ R_0^{i-1}(X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y))_{\mathcal{U}} \end{array}$$

The action on a morphism (g, h) from (X, Y) to (X', Y') is defined such that:

$$R_{i-1}^i W(g, h) := \left(g \triangleleft^{i-1} K_{H_i F_{i-1}}(g, h) \right)$$

We verify in Appendix A.1 that this morphism verifies the property of morphisms of \mathcal{B}_i so that it is indeed a morphism of \mathcal{B}_i .

3.4.2 E definition

We define $E : \mathcal{B}_i \rightarrow (X : \mathcal{B}_{i-1}) \times (\mathcal{S}et/H_i F_{i-1} X)$

The action on objects is

$$E(X) = (R_{i-1}^i X, (A, h))$$

Where (A, h) is defined as the following pullback:

$$\begin{array}{ccc} A & \xrightarrow{\quad} & R_0^i(X)_{\mathcal{E}1} \\ \downarrow \lrcorner \scriptstyle h & & \downarrow R_0^i(X)_p \\ H_i F_{i-1} R_{i-1}^i X & \xrightarrow{\text{Cstr}^X} & R_0^i(X)_{\mathcal{U}} \end{array}$$

The action on a morphism f from X to X' is defined as $E(f) = (R_{i-1}^i f, !)$, with $!$ being the only morphism making the following diagram commute (thanks to the pullback property):

$$\begin{array}{ccccc} A & \xrightarrow{\quad} & R_0^i(X)_{\mathcal{E}1} & & \\ \downarrow \lrcorner \scriptstyle h & \searrow \scriptstyle ! & \downarrow & \searrow \scriptstyle R_0^i(f)_{\mathcal{E}1} & \\ & & A' & \xrightarrow{\quad} & R_0^i(X')_{\mathcal{E}1} \\ & & \downarrow \lrcorner \scriptstyle h' & & \downarrow R_0^i(X)_p \\ & & H_i F_{i-1} R_{i-1}^i X & \xrightarrow{\text{Cstr}^X} & R_0^i(X)_{\mathcal{U}} \\ & \searrow \scriptstyle H_i F_{i-1} R_{i-1}^i f & & \searrow \scriptstyle R_0^i(f)_{\mathcal{U}} & \\ & & H_i F_{i-1} R_{i-1}^i X' & \xrightarrow{\text{Cstr}^{X'}} & R_0^i(X')_{\mathcal{U}} \\ & & & & \downarrow R_0^i(X')_p \end{array}$$

3.5 Inductive Step: Proof of the coreflection

In this subsection, we prove that (W, E) make up a coreflection i.e. an adjunction where the left adjoint is full and faithful.

Adjunction We will prove the adjunction by showing that there is a natural isomorphism between $\mathcal{H}om$ sets in both categories.

We want to construct for each (X, Y) in $(X : \mathcal{B}_{i-1}) \times (\text{Set}/H_i F_{i-1} X)$ and Z in \mathcal{B}_i , a natural isomorphism ϕ_{XYZ} .

$$\phi_{XYZ} : \mathcal{H}om(W(X, Y), Z) \rightarrow \mathcal{H}om((X, Y), E(Z))$$

You can find the proof of the adjunction in [Appendix B](#), made of the following parts:

- The construction of ϕ_{XYZ}
- The construction of ϕ_{XYZ}^{-1}
- A proof that $\phi_{XYZ}^{-1} \circ \phi_{XYZ}(f) = f$
- A proof that $\phi_{XYZ} \circ \phi_{XYZ}^{-1}(g, h) = (g, h)$
- A proof that ϕ_{XYZ} is natural.

All these steps give us that F_i and G_i are in an adjunction $F_i \vdash G_i$.

Coreflection Next, we have proven that this newly created adjunction $F_i \vdash G_i$ creates a coreflection. It means that $F_i G_i \cong \mathcal{I}d_{\mathcal{C}_i}$, or equivalently that G_i is full and faithful.

The proof of that second statement is given in [Appendix C](#).

3.6 Inductive step: Constructing \triangleleft^i

Constructing the objects We will define the \triangleleft^i operator of two objects X from \mathcal{B}_i and Y from \mathcal{B}_0 as follows:

$$X \triangleleft^i Y := \left((R_{i-1}^i X) \triangleleft^{i-1} Y, (R_0^{i-1} \text{inj}_1^{i-1})_{\mathcal{U}} \circ \mathcal{C}\text{str}_i^X \circ (H_i F_{i-1} \text{inj}_1^{i-1})^{-1} \right)$$

The constructor goes as follows:

$$\begin{array}{c} H_i F_{i-1} (R_{i-1}^i X \triangleleft^{i-1} Y) \\ \downarrow \simeq H_i (F_{i-1} \text{inj}_1^{i-1})^{-1} \\ H_i F_{i-1} (R_{i-1}^i X) \\ \downarrow \mathcal{C}\text{str}^X \\ (R_0^i X)_{\mathcal{U}} \\ \downarrow (R_0^{i-1} \text{inj}_1^{i-1})_{\mathcal{U}} \\ (R_0^{i-1} (R_{i-1}^i X \triangleleft^{i-1} Y))_{\mathcal{U}} \end{array}$$

The first injector $\text{inj}_1^i : X \rightarrow X \triangleleft^i Y$ is defined such that:

$$R_{i-1}^i \text{inj}_1^i := \text{inj}_1^{i-1} : R_{i-1}^i X \rightarrow R_{i-1}^i (X \triangleleft^i Y) = R_{i-1}^i X \triangleleft^{i-1} Y$$

We check that inj_1^i is a morphism of \mathcal{B}_i in [Appendix A.2](#).

The second injector is defined as

$$\text{inj}_2^i := \text{inj}_2^{i-1} : Y \rightarrow R_0^{i-1} (R_{i-1}^i X \triangleleft^{i-1} Y) = R_0^i (X \triangleleft^i Y)$$

Universal Property We will now show that the universal property stated in the introduction of this section holds. To that end, we take two objects X and Z in \mathcal{B}_i , Y in \mathcal{B}_0 , a morphism $g : X \rightarrow Z$ in \mathcal{B}_i and a morphism $h : Y \rightarrow R_0^i Z$ in \mathcal{B}_0 . We want to build a morphism $\{g, h\}$ of \mathcal{B}_i such that the following diagrams of \mathcal{B}_i and \mathcal{B}_0 commute.

$$\begin{array}{ccc}
 X & \xrightarrow{\text{inj}_1^i} & X \triangleleft^i Y \\
 & \searrow g & \downarrow \{g, h\} \\
 & & Z
 \end{array}
 \qquad
 \begin{array}{ccc}
 R_0^i X & \xrightarrow{R_0^i \text{inj}_1^i} & R_0^i(X \triangleleft^i Y) \xleftarrow{\text{inj}_2^i} Y \\
 & \searrow R_0^i g & \downarrow R_0^i \{g, h\} \\
 & & R_0^i Z
 \end{array}$$

We check that $\{R_{i-1}^i g; h\}_{i-1}$ verifies the property of morphisms of \mathcal{B}_i in Appendix A.3. Therefore, we can define $\{g; h\}_i$ such that $R_{i-1}^i \{g; h\}_i = \{R_{i-1}^i g; h\}_{i-1}$.

This morphism $\{g; h\}_i$ is such that the universal property for it is the same as the one of $\{R_{i-1}^i g, h\}$.

With this definition, the isomorphism $\mathfrak{H}_{i-1}^i : (R_{i-1}^i X) \triangleleft^{i-1} Y \rightarrow R_{i-1}^i(X \triangleleft^i Y)$ is simply the identity morphism.

Building morphisms For any two morphisms $g : X \rightarrow X'$ of \mathcal{B}_i and $h : Y \rightarrow Y'$ of \mathcal{B}_0 , we check that the morphism $R_{i-1}^i g \triangleleft^{i-1} h$ verifies the morphism property of \mathcal{B}_i in Appendix A.4, so that we can define a morphism $g \triangleleft^i h : X \triangleleft^i Y \rightarrow X' \triangleleft^i Y'$ such that

$$R_{i-1}^i (g \triangleleft^i h) = (R_{i-1}^i g) \triangleleft^{i-1} h$$

Composition with F_i We finally need to prove, for any objects X in \mathcal{B}_i and Y in $[\mathcal{T}, \text{Set}]$, that the morphism $F_i(\text{inj}_1^i) : F_i X \rightarrow F_i(X \triangleleft^i Y)$ is an isomorphism.

Morphisms \mathcal{B}_i are morphisms of \mathcal{B}_{i-1} that follow some equations, and composition is the same. Moreover, we know from the definition above that $R_{i-1}^i \text{inj}_1^i = \text{inj}_1^{i-1}$.

So, we just need to prove that $R_{i-1}^i(F_{i-1} \text{inj}_1^i)$ is an isomorphism.

$$R_{i-1}^i F_i \text{inj}_1^i = \left[(F_{i-1} \times \text{id})(E(\text{inj}_1^i)) \right]_1 = \left[(F_{i-1} \times \text{id})(R_{i-1}^i \text{inj}_1^i, !) \right]_1 = F_{i-1} \text{inj}_1^{i-1}$$

And we know from the previous induction step that $F_{i-1} \text{inj}_1^{i-1}$ is an isomorphism. Therefore, $F_i \text{inj}_1^i$ is an isomorphism.

4 Other Properties

4.1 Sort specifications as simple direct categories

In the formal proof, we defined \mathcal{C}_i as an inductive Grothendieck construction as in Altenkirch and al. paper [Altenkirch2018].

But there is another way [Fiore2008] of building the category \mathcal{C}_i which is equivalent to the one presented above.

With this other method, a sort specification is described by a finite simple direct category S (i.e. a finite category where all the arrows follow a unique direction, and Hom -sets are finite). The models of the sort specification are therefore the presheaves over the category S called $[S, \text{Set}]$.

Constructing S as a sequence To fall back on the definition we gave in the formal proof, we will build the final countable simple direct category S as a sequence $\emptyset = S_0, S_1, \dots, S_i$ where S_i is the category describing the sort specification limited to the i first sort declarations.

The input data describing the GAT will be a sequence of finite functors $\Gamma_i : S_{i-1} \rightarrow \text{Set}$.

We will then construct the category S_i by adding a single object α_i to the category S_{i-1} , along with morphisms $f : \alpha_j \rightarrow \alpha_{i+1}$ for $f \in \Gamma_{i+1}(\alpha_j)$ and $j \leq i$. The morphisms follow the composition condition, describing that every pair of morphisms $f : \alpha_j \rightarrow \alpha_{i+1}$ and $g : \alpha_k \rightarrow \alpha_{i+1}$ (i.e. $f \in \Gamma_{i+1}(\alpha_k)$ and $g \in \Gamma_{i+1}(\alpha_j)$) and for every morphism $h : \alpha_j \rightarrow \alpha_k$ of S_i , we have $\Gamma_{i+1}(h)(f) \circ f = g$.

The final category is simply $S = \bigcup S_i$.

In our type theory example, the category S is constructed as follows:

$$\begin{array}{c|c|c}
 \boxed{\text{Con} : \text{Set}} & \boxed{\text{Ty} : (\Gamma : \text{Con}) \rightarrow \text{Set}} & \boxed{\text{Tm} : (\Delta : \text{Con}) \rightarrow (A : \text{Ty } \Delta) \rightarrow \text{Set}} \\
 S_1 = \text{Con} & S_2 = \text{Con} \leftarrow \text{Ty} & S_3 = \text{Con} \leftarrow \text{Ty} \leftarrow \text{Tm} \\
 [S_1, \text{Set}] \simeq \mathcal{C}_1 & [S_2, \text{Set}] \simeq \mathcal{C}_2 & [S_3, \text{Set}] \simeq \mathcal{C}_3
 \end{array}$$

Remark 4 *This definition is reversible (up to the order of the sorts), and therefore, one can define a sequence of Γ_i functors from any countable simple direct category.*

Link with the formal proof In the formal proof, we used categories \mathcal{C}_i and functors $H_i : \mathcal{C}_i \rightarrow \text{Set}$. We now understand that the categories \mathcal{C}_i are defined on top of the categories S_i with $\mathcal{C}_i := [S_i, \text{Set}]$.

In the same way, the functors $H_i : \mathcal{C}_{i-1} \rightarrow \text{Set}$ are defined on top of the functors Γ_i with the following relation:

$$H_i(X) := \mathcal{H}om_{[S_{i-1}, \text{Set}]}(\Gamma_i, X)$$

4.2 Creating semantics object from the syntax

In this part, we will see how the functors Γ_i given above can be constructed from the syntax of a sort specification. The previous subsection gives us a way of constructing the H_i functors from the Γ_i functors, and thus one can make the complete proof with the definition below.

Verbose sort specification We will take a verbose version of a sort specification, where when “implementing” a sort, we give the name of the parameter and the name of the term. For example, our type theory example becomes the following verbose version:

$$\begin{array}{l}
 \text{Con} : \text{Set} \\
 \text{Ty} : (\Gamma : \text{Con}) \rightarrow \text{Set} \\
 \text{Tm} : (\Delta : \text{Con}) \rightarrow (A : \text{Ty } [\Gamma \mapsto \Delta]) \rightarrow \text{Set}
 \end{array}$$

The only sort declaration that changes is that of Tm as it is the only one for which one of its parameters is a sort with parameters.

With that verbose definition, a generic sort specification is a list of sort declarations indexed by \mathbb{T} , the fully ordered set of all sorts. The sort declarations are as follows, for a sort $T \in \mathbb{T}$

$$T : \left[x_a^T : U_T(a) \left[x_b^{U_T(a)} \mapsto x_{v_T(a,b)}^T \right]_{b \leq I_{U_T(a)}} \right]_{a \leq I_T} \rightarrow \text{Set}$$

Where

- I_T is the number of parameters of the sort T
- x_a^T is the a th parameter of the sort T
- $U_T(a)$ is the sort of the a th parameter of the sort T (we have $U_T(a) < T$)
- $I_{U_T(a)}$ is the number of parameters of the sort $U_T(a)$ i.e. the number of arguments we have to give to make an object of sort $U_T(a)$
- $x_b^{U_T(a)}$ is the b th parameter of the sort $U_T(a)$
- $v_T(a,b)$ is the index of the parameter of the sort T given as the b th parameter of the a th parameter of the sort T (we have $v_T(a,b) < a$)
- The types of parameters have to be respected, therefore we must have the equality

$$U_T(v_T(a,b)) = U_{U_T(a)}(b)$$

For example, for the above sort declaration, we would have

- $\mathbb{T} = \{\text{Con} < \text{Ty} < \text{Tm}\}$
- $I_{\text{Con}} = 0, I_{\text{Ty}} = 1, I_{\text{Tm}} = 2$
- $x_1^{\text{Ty}} = \text{"}\Gamma\text{"}, x_1^{\text{Tm}} = \text{"}\Delta\text{"}, x_2^{\text{Tm}} = \text{"}A\text{"}$
- $U_{\text{Ty}}(1) = \text{Con}, U_{\text{Tm}}(1) = \text{Con}, U_{\text{Tm}}(2) = \text{Ty}$
- $v_{\text{Tm}}(2, 1) = 1$ such that $x_{v_{\text{Tm}}(2,1)}^{\text{Tm}} = \text{"}\Delta\text{"}$ (associated to $x_{b=1}^{\text{Ty}} = \text{"}\Gamma\text{"}$)

Describing the category S We will then describe what is the category S corresponding to a given sort specification (\mathbb{T}, I, U, v)

Its objects are simply the sorts, i.e. the elements of \mathbb{T} .

Morphisms from a sort T_x to a different sort T_y are described by U and correspond to the parameters of the sort T_x that are of sort T_y . In other words:

$$\text{Hom}_S(T_x, T_y) = \left\{ x_a^{T_x} \mid a \leq I_{T_x} \text{ and } U_{T_x}(a) = T_y \right\}$$

We can see that the category is direct, because of the rule $U_T(a) < T$ i.e. if $T_x < T_y$ then $\text{Hom}_S(T_x, T_y) = \emptyset$. For every sort T , we define $\text{Hom}_S(T, T)$ to contain only the identity morphism id_T , for which composition rules are already defined.

Composition is described by v , corresponding to “parameters of parameters”. Let’s take two morphisms $x_b^{T_x} : T_x \rightarrow T_y$ and $x_a^{T_y} : T_y \rightarrow T_z$. Their composition is defined by

$$x_a^{T_y} \circ x_b^{T_x} = x_{v_{T_x}(a,b)}^{T_x}$$

This composition is well-defined as

$$U_{T_x}(v_{T_x}(a, b)) = U_{U_{T_x}(a)}(b) = U_{T_y}(b) = T_z$$

For our type theory example, the category S is as follows

$$\begin{array}{ccccc} & & \text{"}\Delta\text{"} = \Gamma \circ A & & \\ & \swarrow & \text{---} & \searrow & \\ \text{Con} & \xleftarrow{\text{"}\Gamma\text{"}} & \text{Ty} & \xleftarrow{\text{"}A\text{"}} & \text{Tm} \end{array}$$

Creating the functor Γ_i We know we can create the functors Γ_i directly from a complete category S , but we will look more in detail at what those functors mean related to the syntax.

If we are given the category S_{i-1} with objects corresponding to the $i - 1$ sorts already defined.

We want to build a finite functor $\Gamma_i : S_{i-1} \rightarrow \text{Set}$ corresponding to the following sort declaration of a new sort T_i

We will build Γ_i as follows:

$$\Gamma_i(T_x) = \left\{ x_a^{T_i} \mid a \leq I_{T_i} \text{ and } U_{T_i}(a) = T_x \right\}$$

$$\Gamma_i(x_b^{T_x} : T_x \rightarrow T_y)(x_a^{T_i} \in \Gamma_i(T_x)) = x_{v_{T_x}(a,b)}^{T_i}$$

This definition is redundant with that of the last paragraph, but it has the advantage of directly creating the functors Γ_i that are used to create the functors H_i and all the rest of the proof.

5 Summary

The initial goal of this internship was to study the transformation of GATs to GATs with only two sorts. We have only studied the transformation of sort specifications. We managed to give a formal construction of the semantics of any sort specification and its transformation.

We have described a relation between both categories, which is a coreflection, i.e. an adjunction where the left adjoint is full and faithful. This adjunction allows us to build the initial model of the initial GAT from the initial model of the transformed GAT, so one can study the properties of GATs with only two sorts and transpose them to any GATs.

There is still work to be done to make the contents of this document apply to any GAT instead of only sort specifications. There are also ways of generalizing our formal construction to description more general than GATs like GATs with sorts mutually included one into the other.

6 Bibliography

Appendices

A Soundness of definitions

A.1 $W(g, h)$ morphism of \mathcal{B}_i

The morphism $R_{i-1}^i W(g, h) = g \triangleleft_{i-1} K_{H_i F_{i-1}}(g, h)$ is a morphism of \mathcal{B}_i as it makes the following diagram commute:

$$\begin{array}{ccc}
 H_i F_{i-1}(X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y)) & \xrightarrow{C_{\text{str}}^{W(X, Y)}} & R_0^{i-1}(X \triangleleft^{i-1} K_{\bullet}(X, Y))_{\mathcal{U}} \\
 \downarrow H_i F_{i-1}(g \triangleleft^{i-1} K_{\bullet}(g, h)) & \searrow^{H_i(F_{i-1} \text{inj}_1^{i-1})^{-1}} & \uparrow \text{inj}_2^{i-1} \\
 & H_i F_{i-1} X = [K_{H_i F_{i-1}}(X, Y)]_{\mathcal{U}} & \\
 & \downarrow H_i F_{i-1} g \quad [K_{H_i F_{i-1}}(g, h)]_{\mathcal{U}} & \\
 & H_i F_{i-1} X' = [K_{H_i F_{i-1}}(X', Y')]_{\mathcal{U}} & \\
 & \downarrow H_i(F_{i-1} \text{inj}_1^{i-1})^{-1} & \\
 H_i F_{i-1}(X' \triangleleft^{i-1} K_{H_i F_{i-1}}(X', Y')) & \xrightarrow{C_{\text{str}}^{W(X', Y')}} & R_0^{i-1}(X' \triangleleft^{i-1} K_{\bullet}(X', Y'))_{\mathcal{U}} \\
 & \swarrow_{H_i(F_{i-1} \text{inj}_1^{i-1})^{-1}} & \downarrow R_0^i(g \triangleleft^{i-1} K_{\bullet}(g, h))_{\mathcal{U}}
 \end{array}$$

A.2 inj_1^i morphism of \mathcal{B}_i

The morphism $R_{i-1}^i \text{inj}_1^i = \text{inj}_1^{i-1}$ is a morphism of \mathcal{B}_i as it makes the following diagram commute:

$$\begin{array}{ccc}
 H_i F_{i-1} R_{i-1}^i X & \xrightarrow{C_{\text{str}}^X} & (R_0^i X)_{\mathcal{U}} \\
 \downarrow H_i F_{i-1} \text{inj}_1^{i-1} \simeq & \searrow & \downarrow R_0^{i-1}(\text{inj}_1^{i-1})_{\mathcal{U}} \\
 H_i F_{i-1}(R_{i-1}^i X \triangleleft^{i-1} Y) & \xrightarrow{H_i(F_{i-1} \text{inj}_1^{i-1})^{-1}} & H_i F_{i-1} R_{i-1}^i X \xrightarrow{C_{\text{str}}^X} (R_0^i X)_{\mathcal{U}} \\
 & \simeq & \downarrow R_0^{i-1}(\text{inj}_1^{i-1})_{\mathcal{U}} \\
 & & (R_0^{i-1}(R_{i-1}^i X \triangleleft^{i-1} Y))_{\mathcal{U}} \\
 & \xrightarrow{C_{\text{str}}^{X \triangleleft^i Y}} &
 \end{array}$$

A.3 $\{g, h\}_i$ morphism of \mathcal{B}_i

$R_{i-1}^i \{g, h\}_i := \{R_{i-1}^i g, h\}_{i-1}$ is a morphism of \mathcal{B}_i as it makes the following diagram commute:

$$\begin{array}{ccc}
 & H_i(F_{i-1} \text{inj}_1^{i-1})^{-1} & H_i F_{i-1} R_{i-1}^i X \xrightarrow{C_{\text{str}}^X} (R_0^i X)_{\mathcal{U}} \xrightarrow{(R_0^{i-1} \text{inj}_1^{i-1})_{\mathcal{U}}} \\
 & \simeq & \downarrow \\
 H_i F_{i-1}(R_{i-1}^i X \triangleleft^{i-1} Y) & & (R_0^{i-1}(R_{i-1}^i X \triangleleft^{i-1} Y))_{\mathcal{U}} \\
 \downarrow H_i F_{i-1} \{g, h\} & \nearrow H_i F_{i-1} R_{i-1}^i g & \downarrow (R_0^i g)_{\mathcal{U}} \\
 & & (R_0^i \{g, h\})_{\mathcal{U}} \\
 H_i F_{i-1} R_{i-1}^i Z & \xrightarrow{C_{\text{str}}^Z} & (R_0^i Z)_{\mathcal{U}}
 \end{array}$$

A.4 $g \triangleleft_i h$ morphism of \mathcal{B}_i

$(R_{i-1}^i g) \triangleleft_{i-1} h$ is a morphism of \mathcal{B}_i as it makes the following diagram commute:

$$\begin{array}{ccccc}
H_i F_{i-1}(R_{i-1}^i X \triangleleft^{i-1} Y) & \xrightarrow[\simeq]{H_i(F_{i-1} \text{inj}_4^{i-1})^{-1}} & H_i F_{i-1} R_{i-1}^i X & \xrightarrow{\mathcal{C}\text{str}^X} & (R_0^i X)_\mathcal{U} & \xrightarrow{(R_0^{i-1} \text{inj}_1^{i-1})_\mathcal{U}} & R_0^{i-1}(R_{i-1}^i X \triangleleft^{i-1} Y)_\mathcal{U} \\
\downarrow H_i F_{i-1}(R_{i-1}^i g \triangleleft^{i-1} h) & & \downarrow H_i F_{i-1} R_{i-1}^i g & & \downarrow (R_0^i g)_\mathcal{U} & & \downarrow R_0^{i-1}(R_{i-1}^i g \triangleleft^{i-1} h)_\mathcal{U} \\
H_i F_{i-1}(R_{i-1}^i X' \triangleleft^{i-1} Y') & \xrightarrow[\simeq]{H_i(F_{i-1} \text{inj}_4^{i-1})^{-1}} & H_i F_{i-1} R_{i-1}^i X' & \xrightarrow{\mathcal{C}\text{str}^{X'}} & (R_0^i X')_\mathcal{U} & \xrightarrow{(R_0^{i-1} \text{inj}_1^{i-1})_\mathcal{U}} & R_0^{i-1}(R_{i-1}^i X' \triangleleft^{i-1} Y')_\mathcal{U}
\end{array}$$

B $W \dashv E$ adjunction

B.1 Constructing ϕ_{XYZ}

Let f be in $\mathcal{H}\text{om}(W(X, Y), Z)$. We want to construct $\phi_{XYZ}(f) : (X, Y) \rightarrow E(Z)$.

The first component of $\phi_{XYZ}(f)$ is defined as the following composition:

$$X \xrightarrow{\text{inj}_1^{i-1}} X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y) = R_{i-1}^i(W(X, Y)) \xrightarrow{R_{i-1}^i f} R_{i-1}^i Z$$

The second component is defined through the universal property of the pullback defined by $E(Z)$ according to the following diagram:

$$\begin{array}{ccccc}
(K_{H_i F_{i-1}}(X, Y))_{\mathcal{E}1} & \xrightarrow{\text{inj}_2^{i-1}} & & & R_0^{i-1}(X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y))_{\mathcal{E}1} \\
\parallel & & & & \downarrow (R_0^{i-1} f)_{\mathcal{E}1} \\
Y & \xrightarrow{\phi_{XYZ}(f)_2} & E(Z) & \longrightarrow & (R_0^i Z)_{\mathcal{E}1} \\
\downarrow y & & \downarrow & & \downarrow (R_0^i Z)_p \\
H_i F_{i-1} X & & H_i F_{i-1} R_{i-1}^i Z & \xrightarrow{\mathcal{C}\text{str}^Z} & (R_0^i Z)_\mathcal{U} \\
\downarrow H_i F_{i-1} \text{inj}_1^{i-1} & & \downarrow H_i F_{i-1} R_{i-1}^i f & & \\
H_i F_{i-1} R_{i-1}^i(W(X, Y)) & \xrightarrow{H_i F_{i-1} R_{i-1}^i f} & H_i F_{i-1} R_{i-1}^i Z & \xrightarrow{\mathcal{C}\text{str}^Z} & (R_0^i Z)_\mathcal{U}
\end{array}$$

B.2 Constructing ϕ_{XYZ}^{-1}

Now, we take (g, h) a morphism from (X, Y) to $E(Z)$.

The morphism $\phi_{XYZ}^{-1}(g, h)$ of \mathcal{B}_i from $W(X, Y)$ to Z is a morphism of \mathcal{B}_{i-1} from $X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y)$ to $R_{i-1}^i(Z)$ that make a specific diagram commute. We build this morphism using the universal property of \triangleleft^{i-1} stated in the introduction of this section on the morphisms g and \square , where \square is a morphism $K_{H_i F_{i-1}}(X, Y) \rightarrow R_0^i Z$ in $[\mathcal{T}, \text{Set}]$ defined by the following diagram:

$$\begin{array}{ccccc}
& & \square_{\mathcal{E}1} & & \\
& & \curvearrowright & & \\
Y & \xrightarrow{h} & E(Z) & \xrightarrow{\pi_1} & (R_0^i Z)_{\mathcal{E}1} \\
\downarrow y & & \downarrow & & \downarrow (R_0^i Z)_p \\
H_i F_{i-1} X & \xrightarrow{H_i F_{i-1} g} & H_i F_{i-1} R_{i-1}^i Z & \xrightarrow{\mathcal{C}\text{str}^Z} & (R_0^i Z)_\mathcal{U} \\
& & \square_{\mathcal{U}} & &
\end{array}$$

We check that the morphism $\{g; \square\}$ verifies the property of morphisms of \mathcal{B}_i i.e. it makes the following diagram commute:

$$\begin{array}{ccc}
H_i(F_{i-1} \text{inj}_0^{i-1})^{-1} & \xrightarrow{\quad} & H_i F_{i-1} X = K_{H_i F_{i-1}}(X, Y)_{\mathcal{U}} & \xrightarrow{(\text{inj}_2^{i-1})_{\mathcal{U}}} & R_0^{i-1}(X \triangleleft^{i-1} K_{\bullet}(X, Y))_{\mathcal{U}} \\
\downarrow H_i F_{i-1} \{g; \square\} & \nearrow \simeq & \downarrow H_i F_{i-1} g & \searrow \square_{\mathcal{U}} & \downarrow (R_0^{i-1} \{g; \square\})_{\mathcal{U}} \\
H_i F_{i-1} (X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y)) & & & & \\
\downarrow H_i F_{i-1} \{g; \square\} & & \downarrow H_i F_{i-1} g & & \downarrow (R_0^{i-1} \{g; \square\})_{\mathcal{U}} \\
H_i F_{i-1} R_{i-1}^i Z & \xrightarrow{C_{\text{str}}^Z} & & & (R_0^i Z)_{\mathcal{U}}
\end{array}$$

Therefore, we can define $\phi_{XYZ}^{-1}(g, h)$ such that

$$R_{i-1}^i \phi_{XYZ}^{-1}(g, h) := \{g; \square\}_{i-1}$$

B.3 Composition $\phi_{XYZ} \circ \phi_{XYZ}^{-1}$

The first component of $\phi_{XYZ}(\phi_{XYZ}^{-1}(g, h))$ is

$$R_{i-1}^i \left(\phi_{XYZ}^{-1}(g, h) \right) \circ \text{inj}_1^{i-1} = \{g; \square\} \circ \text{inj}_1^{i-1} = g$$

The second component of $\phi_{XYZ}(\phi_{XYZ}^{-1}(g, h))$ follows the following diagram

$$\begin{array}{ccccc}
(K_{H_i F_{i-1}}(X, Y))_{\mathcal{E}1} & \xrightarrow{(\text{inj}_2^{i-1})_{\mathcal{E}1}} & & & R_0^{i-1}(X \triangleleft^{i-1} K_{H_i F_{i-1}}(X, Y))_{\mathcal{E}1} \\
\parallel & & & & \downarrow (R_0^{i-1} \{g; \square\})_{\mathcal{E}1} \\
Y & \xrightarrow{\square_{\mathcal{E}1}} & & & (R_0^i Z)_{\mathcal{E}1} \\
\downarrow y & \searrow \phi_{XYZ}(f)_2 & \dashrightarrow h & \searrow & \downarrow (R_0^i Z)_p \\
H_i F_{i-1} X & \xrightarrow{H_i F_{i-1} g} & E(Z) & \xrightarrow{\quad} & (R_0^i Z)_{\mathcal{U}} \\
\downarrow H_i F_{i-1} \text{inj}_1^{i-1} & & \downarrow & & \downarrow C_{\text{str}}^Z \\
H_i F_{i-1} R_{i-1}^i(W(X, Y)) & \xrightarrow{H_i F_{i-1} R_{i-1}^i \{g; \square\}} & H_i F_{i-1} R_{i-1}^i Z & \xrightarrow{C_{\text{str}}^Z} & (R_0^i Z)_{\mathcal{U}}
\end{array}$$

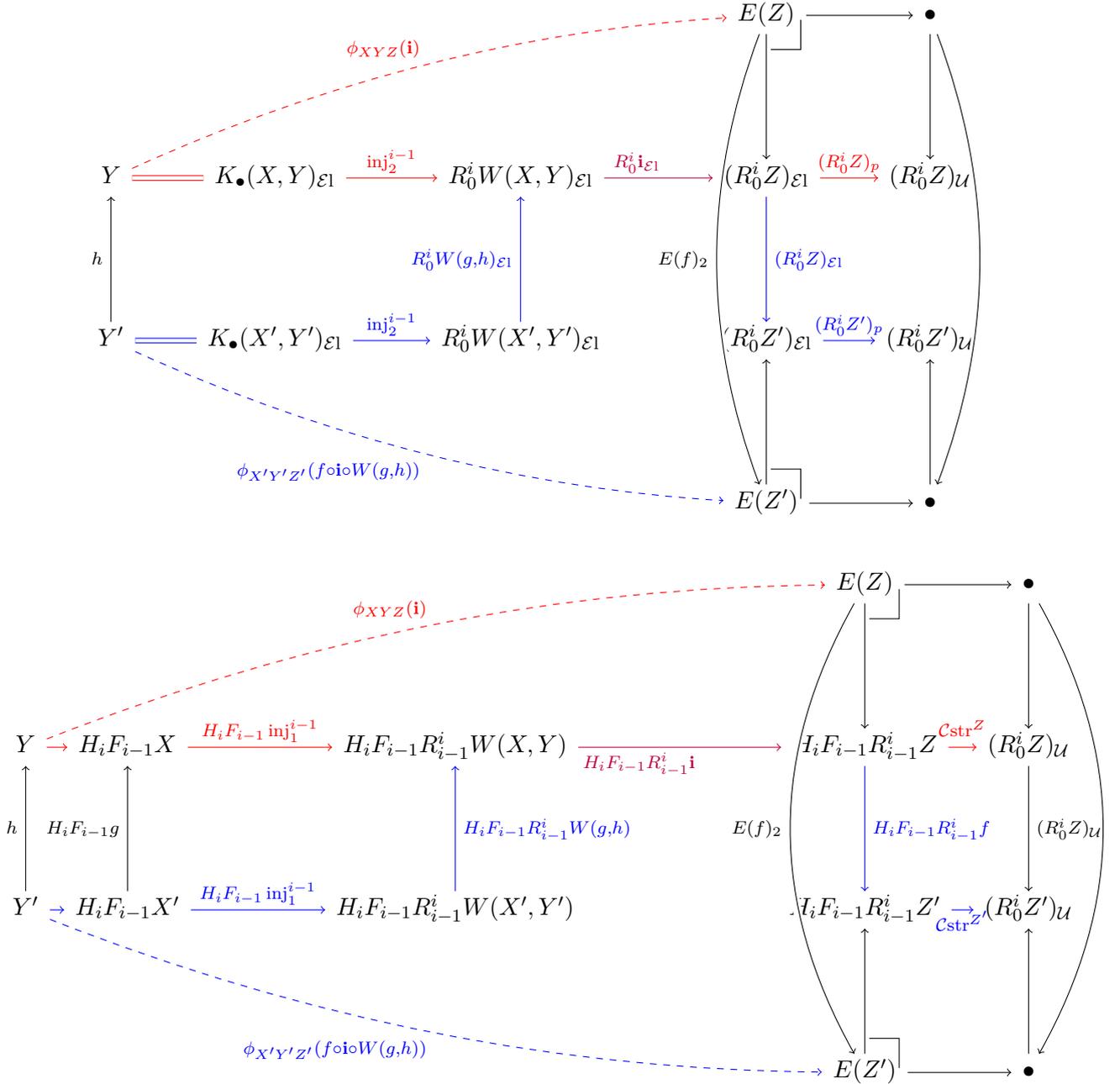
The diagram commutes, and so we can deduce that the second component of $\phi_{XYZ}(f)$ is h , by property of the pullback $E(Z)$

B.4 Composition $\phi_{XYZ}^{-1} \circ \phi_{XYZ}$

Now, the converse composition is

$$\phi_{XYZ}^{-1}(\phi_{XYZ}(f)) = \left\{ R_{i-1}^i f \circ \text{inj}_1^{i-1}; \square \right\}_{i-1}$$

where \square follows the following diagram



The second projection of $\phi_{XYZ}(f \circ \mathbf{i} \circ W(g, h))$ is defined by the pullback of $E(Z')$ commuting with the blue bottom path. And that of $\phi_{X'Y'Z'}(\mathbf{i})$ is defined by the red top path. Both diagrams commute, and therefore, by the pullback property, we get that

$$\phi_{X'Y'Z'}(f \circ \mathbf{i} \circ W(g, h))_2 = [E(f) \circ \phi_{XYZ}(\mathbf{i}) \circ (g, h)]_2$$

And thus we have naturality.

C $F_i \vdash G_i$ reflection

We want to find, for each object $(X, (B, g))$ of $\mathcal{C}_i = (X : \mathcal{C}_{i-1}) \times (\text{Set}/H_i(X))$, an isomorphism $(X, (B, g)) \rightarrow F_i G_i(X, (B, g))$. g is a morphism from B to $H_i(X)$

$$\begin{aligned}
F_i G_i(X, (B, g)) &= F_i W_i(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)) \\
&= F_i \left(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)), \widetilde{\text{inj}}_2 \right) \\
&= (F_{i-1} \times \text{id}) (G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)), (A, h)) \\
&= (F_{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g))), (A, h))
\end{aligned}$$

Where (A, h) is the pullback defined as

$$\begin{array}{ccc}
A & \xrightarrow{\quad} & R_0^{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)))_{\mathcal{E}1} \\
\downarrow h & \lrcorner & \downarrow R_0^{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)))_p \\
H_i F_{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g))) & \xrightarrow[\widetilde{\text{inj}}_2]{} & R_0^{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)))_{\mathcal{U}}
\end{array}$$

We can extend this pullback with the isomorphisms $H_i F_{i-1}(\text{inj}_{\triangleleft}^{i-1})$ into another pullback. This new pullback is over the injection morphism inj_2 of the coproduct of $[\mathcal{T}, \text{Set}]$, so the new pullback object is the second component of the $\bullet \oplus B$ i.e. B , by property of the coproduct.

$$\begin{array}{ccc}
& & \bullet \oplus B \\
& & \parallel \\
B & \xrightarrow{\text{inj}_2} & (R_0^{i-1}G_{i-1}X)_{\mathcal{E}1} \oplus K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g))_{\mathcal{E}1} \\
\downarrow H_i \eta_{i-1} \circ g & \searrow \text{inj}_2^{i-1} & \downarrow h_0^i \simeq \\
& A & \xrightarrow{\quad} & R_0^{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)))_{\mathcal{E}1} \\
& \downarrow h & & \downarrow \\
H_i F_{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g))) & \xrightarrow[\widetilde{\text{inj}}_2]{} & R_0^{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)))_{\mathcal{U}} \\
\downarrow H_i F_{i-1}(\text{inj}_1^i) & \nearrow \text{inj}_2^{i-1} & \downarrow h_0^i \simeq \\
H_i F_{i-1}G_{i-1}X & \xrightarrow{\text{inj}_2} & (R_0^{i-1}G_{i-1}X)_{\mathcal{U}} \oplus K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g))_{\mathcal{U}} \\
& & \parallel \\
& & \bullet \oplus H_i F_{i-1}G_{i-1}X
\end{array}$$

The first component of the isomorphism is the following isomorphism, where η_{i-1} is the counit of the adjunction $F_{i-1} \vdash G_{i-1}$, which we know to be an isomorphism from the induction hypothesis.

$$X \xrightarrow[\simeq]{\eta_{i-1}} F_{i-1}G_{i-1}X \xrightarrow[\simeq]{F_{i-1}(\text{inj}_1^{i-1})} F_{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)))$$

The second component is made using the isomorphism constructed by the pullback, which makes the diagram commute.

$$\begin{array}{ccccc}
B & \xrightarrow[\simeq]{\text{id}_B} & B & \xrightarrow[\simeq]{} & A \\
\downarrow g & & \downarrow H_i \eta_{i-1} \circ g & & \downarrow h \\
H_i X & \xrightarrow[\simeq]{H_i \eta_{i-1}} & H_i F_{i-1}G_{i-1}X & \xrightarrow[\simeq]{H_i F_{i-1} \text{inj}_1^{i-1}} & H_i F_{i-1}(G_{i-1}X \triangleleft^{i-1} K_{\bullet}(G_{i-1}X, (B, H_i \eta_{i-1} \circ g)))
\end{array}$$