

# Graded Generalized Algebraic Data Types

Harley Eades

October 19, 2022

## Abstract

Write abstract

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Graded GADTs Syntactically</b>	<b>1</b>
<b>3</b>	<b>The Fundamental Theory</b>	<b>2</b>
3.1	Interpretation . . . . .	4
3.1.1	The Non-Graded Case . . . . .	4
3.1.2	The Graded Case . . . . .	9
3.2	Initial Algebra Packages for Basic Graded GADTs . . . . .	10

## 1 Introduction

Write the intro.

## 2 Graded GADTs Syntactically

```
data G f1 f2 f3 ... fi h1 h2 h3 ... hi a where
  GCon1 :: f1 m1 (G n1 f1 h1) a -> G (m1 * n1) f1 h1 (h1 a)
  GCon2 :: f2 m2 (G n2 f2 h2) a -> G (m2 * n2) f2 h2 (h2 a)
```

$\text{GCon3} :: \text{f3 m3 (G n3 f3 h3) a} \rightarrow \text{G (m3 * n3) f3 h3 (h3 a)}$   
 $\dots$   
 $\text{GConi} :: \text{fi mi (G ni fi hi) a} \rightarrow \text{G (mi * ni) fi hi (hi a)}$

### 3 The Fundamental Theory

Suppose  $\mathcal{C}$  is a category and  $(\mathcal{E}, \otimes, I)$  is a strict monoidal category.

**Definition 3.1** (Graded F-Algebra). *For a functor  $F : \mathcal{E} \times \mathcal{C} \rightarrow \mathcal{C}$ , a graded F-algebra is a pair  $(A, h)$  that consists of a functor  $A : \mathcal{E} \rightarrow \mathcal{C}$  and a family  $h$  of morphisms:*

$$h_{m,n} : F(m, A(n)) \rightarrow A(m \otimes n)$$

*A **homomorphism** between two graded F-algebras  $(A, h)$  and  $(B, h')$  consists of a morphism*

$$\alpha : (A, h) \rightarrow (B, h')$$

*is defined as a natural transformation  $\alpha : A_1 \rightarrow A_2$  such that:*

$$F(m, \alpha_n); h'_{m,n} = h_{m,n}; \alpha_{m \otimes n}$$

**Definition 3.2.** *If the category of graded F-algebras has an initial object, then we call this a **graded initial** F-algebra denoted by  $(\mu F, \text{in})$ . That is, for any other F-algebra  $(A, h)$  there must be a unique morphism  $\alpha : (\mu F, \text{in}) \rightarrow (A, h)$ , but this implies that for any object  $n$ ,  $\alpha_n : \mu F(n) \rightarrow A(n)$  is unique and  $\mu F(n)$  is an initial object in  $\mathcal{C}$ .*

**Lemma 3.3** (From Structures to Homomorphisms). *Given any graded F-algebra  $(A, h)$ , the structure map  $h$  is also a homomorphism between F-algebras  $(F(m, A(-)), F(-, h_{m,-}))$  and  $(A(m \otimes -), h_{-, m \otimes -})$ .*

*Proof.* This proof holds trivially by writing out the commutative square for the F-algebra homomorphism.  $\square$

**Lemma 3.4** (Graded Lambek's Lemma). *If  $(A, h)$  is a graded initial algebra of F, then for any object  $m$ ,  $A(m \otimes -) : \mathcal{E} \rightarrow \mathcal{C}$  is isomorphic to  $F(m, A(-)) : \mathcal{E} \rightarrow \mathcal{C}$  via  $h_m$ .*

*Proof.* Suppose  $h_{m,n} : F(m, A(n)) \rightarrow A(m \otimes n)$  is an initial algebra structure for any  $m$  and  $n$ . Now define an algebra structure:

$$F(m', h_{m,n}) : F(m', F(m, A(n))) \rightarrow F(m', A(m \otimes n))$$

Then by initiality there exists an F-algebra homomorphism

$$i_m : A(m \otimes -) \longrightarrow F(m, A(-))$$

such that:

$$F(m', i_{m,n}); F(m', h_{m,n}) = h_{m', (m \otimes n)}; i_{m', (m \otimes n)}$$

We also know that  $h_m : F(m, A(-)) \longrightarrow A(m \otimes -)$  is itself a graded F-algebra homomorphism (Lemma 3.3). Thus, since we know that  $A(m \otimes n)$  is an initial object by definition and assumption that  $A$  is a graded initial object, and hence,  $i_{m,n}; h_{m,n} = \text{id}_{m \otimes n}$ .

Next we know that  $i$  is a graded F-algebra homomorphism which implies

$$F(m, i_{n,I}); F(m, h_{n,I}) = h_{m,n}; i_{m,n}$$

but again by initiality we know that

$$F(m, i_{n,I}); F(m, h_{n,I}) = \text{id}_{F(m, A(n))}$$

Therefore,  $i$  is the inverse of  $h$  and we obtain our result.  $\square$

**Definition 3.5** (Graded Folds). *Suppose  $(\mu F, \text{in})$  is a graded initial F-algebra. Then the unique map between  $\mu F$  and any other graded F-algebra  $(A, h)$  is the **fold** for  $\mu F$  and is denoted by*

$$\text{fold}(h) : \mu F \longrightarrow A$$

Furthermore, we know that the following must hold:

$$\text{in}_{m,n}; \text{fold}(h)_{m*n} = F(m, \text{fold}(h)_n); h$$

**Definition 3.6** (Graded Forgetful Limits). *There is a forgetful functor from the category of graded F-algebras and the functor category  $[\mathcal{E}, \mathcal{C}]$  and their natural transformations. This functor is defined as follows:*

$$\begin{aligned} U_F(A, h) &= A \\ U_F(\alpha) &= \alpha \end{aligned}$$

Given an object of  $[\mathcal{E}, \mathcal{C}]$ , say  $X$ , then a  $U_F$ -**cone for**  $X$  comprises, for every graded F-algebra  $(A, h)$ , a natural transformation  $v_{(A,h)} : X \longrightarrow A$  in  $[\mathcal{E}, \mathcal{C}]$  such that, for every graded F-algebra homomorphism  $\alpha : A \longrightarrow B$ , we have

$v_{(B,g)} = \alpha; v_{(A,h)}$ . We denote these cones by  $(X, v)$  and call  $X$  its **vertex** and  $v_{(A,h)}$  the **projection** from  $X$  to  $A$ .

A  $\mathcal{U}_F$ -**cone morphism**  $g : (X, v_1) \rightarrow (Y, v_2)$  is a natural transformation  $g : X \rightarrow Y$  such that for any graded  $F$ -algebra  $(A, h)$ , we have  $g; v_2 = v_1$ . A  $\mathcal{U}_F$ -**limit** is a  $\mathcal{U}_F$ -cone to which there is a unique  $\mathcal{U}_F$ -cone morphism, call the **mediating morphism**, from any other  $\mathcal{U}_F$ -cone.

**Lemma 3.7** (Forgetful Limits are Unique). *If  $(X, v)$  is a  $\mathcal{U}_F$ -limit, then it is unique up to isomorphism.*

*Proof.* Suppose  $(Y, v')$  is another  $\mathcal{U}_F$ -limit. Thus, there is a unique  $\mathcal{U}_F$ -cone morphism  $i : Y \rightarrow X$  such that  $i; v = v'$ . But, there must also be a unique  $\mathcal{U}_F$ -cone morphism  $j : X \rightarrow Y$  such that  $j; v' = v$ . But, by substitution  $i; j; v' = v'$  and  $j; i; v = v$ , but these in addition to the assumption that both  $i$  and  $j$  are unique imply that  $i; j = \text{id}_Y$  and  $j; i = \text{id}_X$ , and thus  $i$  and  $j$  are inverses of each other.  $\square$

## 3.1 Interpretation

### 3.1.1 The Non-Graded Case

We begin this section with an overview of the interpretation of non-graded GADTs. Then show how to move to the graded case. The basic form of a GADT is the following:

```
data G f h a where
  GCon :: f (G f h) a -> G f h (h a)
```

Giving an initial algebra semantics requires that we interpret  $G \ f \ h$  as the carrier of the initial algebra in the category of  $f$ -algebras where the constructor  $GCon$  is the structure map. That is, we have the following mappings:

- $f$  maps to a functor  $f : [|\mathcal{C}|, \mathcal{C}] \rightarrow [|\mathcal{C}|, \mathcal{C}]$ .
- $h$  maps to a functor  $h : |\mathcal{C}| \rightarrow |\mathcal{C}|$ .
- $G \ f \ h$  maps to a functor  $G_{f,h} : |\mathcal{C}| \rightarrow \mathcal{C}$ .
- $GCon$  maps to a natural transformation:

$$\text{in} : f(G_{f,h}(-)) \rightarrow G_{f,h}(h(-))$$

At this point, we can see a problem, we want  $(G_{f,h}, \text{in})$  to be an initial  $f$ -algebra, but it has a target that does not fit the proper form, because it is currently  $G_{f,h}(h(-))$ , and does not match the parameter to  $f$  in the source, due to the application of  $h$ . Thus, in its current form,  $\text{in}$  does not match the structure map we need. Rather, we need it to have a target of  $G_{f,h}(-)$ .

We can overcome this problem using the notion of a left Kan extension.

**Definition 3.8** (Left Kan Extension). *The left Kan extension of a functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  along a functor  $P : \mathcal{C} \rightarrow \mathcal{C}'$  is, if it exists, a functor  $\text{Lan}_P F : \mathcal{C}' \rightarrow \mathcal{D}$  equipped with a natural isomorphism:*

$$\text{Hom}_{[\mathcal{C}, \mathcal{D}]}(F, P^*) \cong \text{Hom}_{[\mathcal{C}', \mathcal{D}]}(\text{Lan}_P F, \text{id})$$

where  $P^*(H : \mathcal{C}' \rightarrow \mathcal{D}) = P; H$ .

If we can define  $\text{Lan}_h f(G_{f,h}(-))$  and its associated natural isomorphism then we can simply apply the latter to  $\text{in}$  to obtain an isomorphic natural transformation that fits the form of the structure map we need. This is possible using the notion of a coend.

It is well-known left Kan extensions are equivalent to coends. Instantiated to our case, we know that our left Kan extension is equivalent to a coend:

$$\text{Lan}_h f(G_{f,h}(c)) \cong \exists(b : |\mathcal{C}|). \text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(G_{f,h}(b))$$

for any  $c \in |\mathcal{C}|$ . Thus, to define our left Kan extension is to define the above coend.

**Definition 3.9** (Cowedge). *Suppose  $F : \mathcal{C}^{op} \times \mathcal{C} \rightarrow \mathcal{D}$  is a functor. A **cowedge**  $e : \mathcal{F} \rightarrow w$  is an object  $w$  and a family of maps  $e_c : F(c, c) \rightarrow w$  for each  $c$ , such that given any other morphism  $f : \mathcal{C}' \rightarrow \mathcal{C}$ , the following holds:*

$$F(f, f); e_{\mathcal{C}'} = F(\text{id}_{\mathcal{C}'}, f); e_c$$

*Cowedges are also preserved by composition, that is given a cowedge  $e : \mathcal{F} \rightarrow w$  and a map  $f : w \rightarrow v$ , then  $e; f : v \rightarrow \mathcal{F}$  is a cowedge.*

**Definition 3.10** (Coend). *Suppose  $F : \mathcal{C}^{op} \times \mathcal{C} \rightarrow \mathcal{D}$  is a functor. A **coend** of  $F$  denoted  $\exists(c : \mathcal{C}). F(c, c)$  is a universal cowedge  $e : \mathcal{F} \rightarrow w$  where every other cowedge  $e' : \mathcal{F} \rightarrow w'$  factors through  $e$  via a unique map  $w \rightarrow w'$ .*

The functor we wish to take the coend of is:

$$\text{Hom}_{|\mathcal{C}|}(h(-), c) \times f(G_{f,h}(-)) : |\mathcal{C}| \longrightarrow \text{Set} \times \mathcal{C}$$

for any object  $c \in |\mathcal{C}|$ . Thus, we need to find an object  $w \in |\mathcal{C}|$  and an universal cowedge  $e : \text{Hom}_{|\mathcal{C}|}(h(-), c) \times f(G_{f,h}(-)) \longrightarrow w$ . As it turns out, what we essentially want  $w$  to be is the disjoint union of  $\text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(G_{f,h}(b))$  indexed by objects  $b \in |\mathcal{C}|$ . In pure categorical terms, we want the indexed coproduct, and we can define this coproduct by taking the colimit of a particular functor projecting from the comma category.

**Definition 3.11** (Comma Category). *Suppose we have functors  $F : \mathcal{D} \longrightarrow \mathcal{C}$  and  $G : \mathcal{E} \longrightarrow \mathcal{C}$ . Then the arrow category  $F \downarrow G$  consists of:*

- *Objects are triples  $(d \in \mathcal{D}, e \in \mathcal{E}, f : Fd \longrightarrow Ge \in \mathcal{C})$ .*
- *Morphisms  $(d_1, e_1, f_1) \longrightarrow (d_2, e_2, f_2)$  are pairs  $(h : d_1 \longrightarrow d_2, k : e_1 \longrightarrow e_2)$  such that the following holds:*

$$f_1; Gk = Fh; f_2$$

*There is a projection functor:*

$$\begin{array}{ccc} F \downarrow G & \xrightarrow{\Pi_{\mathcal{D}}} & \mathcal{D} \\ \Pi(d, e, f) & = & d \\ \Pi(h, k) & = & h \end{array} \quad \begin{array}{ccc} F \downarrow G & \xrightarrow{\Pi_{\mathcal{E}}} & \mathcal{E} \\ \Pi(d, e, f) & = & e \\ \Pi(h, k) & = & k \end{array}$$

We can instantiate the above with the functor  $h : |\mathcal{C}| \longrightarrow |\mathcal{C}|$  and an object  $c \in |\mathcal{C}|$  to obtain the category  $h \downarrow c$ :

- *Objects are pairs  $(b \in |\mathcal{C}|, \text{id}_b : h(b) \longrightarrow c \in |\mathcal{C}|)$ .*
- *Morphisms  $(b_1, \text{id}_{b_1}) \longrightarrow (b_2, \text{id}_{b_2})$  are morphisms  $f : b_1 \longrightarrow b_2 \in |\mathcal{C}|$ , but this implies that  $f = \text{id}_{b_1}$ .*

In the objects above we write the identities as  $\text{id}_b : h(b) \longrightarrow c$  which can be written as the equation  $h(b) = c$ . Thus, we have a discrete category of all objects  $b \in |\mathcal{C}|$  such that  $h(b) = c$  which is a full subcategory of  $|\mathcal{C}|$ .

Now if we take the first projection:

$$h \downarrow c \xrightarrow{\Pi_{|\mathcal{C}|}} |\mathcal{C}|$$

we are projecting out the the object  $b$  and forgetting the proof that  $h(b) = c$ . Finally, we can compose this with  $f(G_{f,h}(-))$  as follows:

$$h \downarrow c \xrightarrow{\Pi_{|\mathcal{C}|}} |\mathcal{C}| \xrightarrow{f(G_{f,h}(-))} \mathcal{C}$$

Applying this functor to an object  $(b, h(b) = c)$  yields an object  $f(G_{f,h}(b))$ . Now since we are dealing with a discrete category we can take the colimit of this functor to obtain the coproduct we denote as

$$\coprod_{(b, c=h(b)) \in h \downarrow c} (f(G_{f,h}(b)))$$

with injections:

$$\text{inj}_b : \text{Hom}_{|\mathcal{C}|}(c, h(b)) \times f(G_{f,h}(b)) \longrightarrow \coprod_{(b, h(b)=c) \in h \downarrow c} (f(G_{f,h}(b)))$$

which can be written:

$$\text{inj}_{(b, h(b)=c)} : f(G_{f,h}(b)) \longrightarrow \coprod_{(b, h(b)=c) \in h \downarrow c} (f(G_{f,h}(b)))$$

This coproduct is the object  $w$  we require to define our cowedge for the coend:

$$\exists(b : |\mathcal{C}|). \text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(G_{f,h}(b))$$

Then the universal cowedge is the object  $\coprod_{(b, h(b)=c) \in h \downarrow c} (f(G_{f,h}(b)))$  and the family of maps:

$$e_b : \text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(G_{f,h}(b)) \longrightarrow \coprod_{(b, h(b)=c) \in h \downarrow c} (f(G_{f,h}(b)))$$

But, these are simply the injections of the coproduct. So take,  $e_b = \text{inj}_b$ . The universal property of the cowedge then follows from the fact that  $|\mathcal{C}|$  is discrete.

At this point a summary of our current results is in order. Recall that we wish to model the following basic GADT using initial algebra semantics:

```
data G f h a where
  GCon :: f (G f h) a -> G f h (h a)
```

But, we ran into a problem because the form of the target type of  $GCon$  is not have the form  $G \ f \ h \ b$  because of the application of  $h$  and hence,  $GCon$  cannot be modeled as the structure map of the initial algebra of  $f$ . We then learned how to calculate that the left Kan extension of  $f(G_{h,f}(-))$  using the coend and coproducts. Now we can obtain an isomorphic version of  $GCon$  using the following fact about Left Kan extensions and coends:

$$\begin{aligned} & \text{Hom}_?(\exists(b : |\mathcal{C}|). \text{Hom}_{|\mathcal{C}|}(h(b), a) \times f(G_{f,h}(b)), G_{h,f}(a)) \\ & \cong \text{Hom}_?(\text{Lan}_h f(G_{f,h}(a)), G_{h,f}(a)) \\ & \cong \text{Hom}_?(f(G_{h,f}(a)), G_{h,f}(h(a))) \end{aligned}$$

However, we still have a problem here. The form of the source object in the definition of  $\text{in}$  does not fit the form of a structure map:

$$\text{in}_c : \exists(b : |\mathcal{C}|). \text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(G_{f,h}(b)) \longrightarrow f(G_{f,h}(c))$$

This can be fixed by defining the following functor:

$$\begin{aligned} K_{f,h} : [|\mathcal{C}|, \mathcal{C}] &\longrightarrow [|\mathcal{C}|, \mathcal{C}] \\ K_{f,h}(g, c) &= \exists(b : |\mathcal{C}|). \text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(c, g(b)) \end{aligned}$$

Using this functor we can see that  $\text{in}$  is equivalent to:

$$\text{in}_c : K_{f,h}(G_{f,h}(c)) \longrightarrow G_{f,h}(c)$$

Thus,  $(G_{f,h})$  is an initial  $K_{f,h}$ -algebra.

Since coends correspond to existential quantifiers we can redefine our GADT as follows:

```
data G f h a where
  GCon :: exists b. (Eq1 (h b) a, f (G f h) b) -> G f h a
```

This version of our basic GADT can now be interpreted using initial algebra semantics as follows:

- $h$  maps to a functor  $h : |\mathcal{C}| \longrightarrow |\mathcal{C}|$ .
- $f$  maps to the functor  $K_{f,h} : [|\mathcal{C}|, \mathcal{C}] \longrightarrow [|\mathcal{C}|, \mathcal{C}]$ .
- $G \ f \ h$  maps to a functor  $G_{f,h} : |\mathcal{C}| \longrightarrow \mathcal{C}$ .
- $GCon$  maps to the natural transformation:

$$\text{in} : K_{f,h}(G_{f,h}(c)) \longrightarrow G_{f,h}(-)$$

making  $(G_{h,f}, \text{in})$  an initial  $K_{f,h}$ -algebra.

We can generalize this semantics to the graded case.



### 3.1.2 The Graded Case

Consider the case of graded GADTs. Using graded initial algebras we want the semantics to correspond to:

- $f$  maps to a functor  $f : \mathcal{E} \times [|\mathcal{C}|, \mathcal{C}] \rightarrow [|\mathcal{C}|, \mathcal{C}]$ .
- $h$  maps to a functor  $h : |\mathcal{C}| \rightarrow |\mathcal{C}|$ .
- $G \circ f \circ h$  maps to a functor  $G_{f,h} : \mathcal{E} \rightarrow [|\mathcal{C}|, \mathcal{C}]$ .
- $GCon$  maps to a natural transformation:

$$in_{m,n} : f(m, G_{f,h}(n, b)) \rightarrow G_{f,h}(m \otimes n, h(b))$$

making  $(G_{f,h}, in)$  a graded initial  $f$ -algebra.

But, we have the same problem as before where the target of  $in_{m,n}$  would have an application of  $h$ , but this application does not appear in the source of  $in$ . Just as before, we make use of the left Kan extension to fix this problem, and hence, equivalently a coend that is computed by a coproduct.

In the graded setting, the functor we want to take the left Kan extension of is:

$$f(m, G_{f,h}(n, -)) : |\mathcal{C}| \rightarrow \mathcal{C}$$

for any grades  $m$  and  $n$ . Now the left Kan extension is equivalent to the following coend:

$$(Lan_h f(m) G_{f,h}(n, -))(c) \cong \exists (b : |\mathcal{C}|). \text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(m, G_{f,h}(n, b))$$

for every grade  $m$  and  $n$ . In fact, constructing this coend is the same as constructing the coend for the non-graded case. The universal cowedge is the object  $\coprod_{(b, h(b)=c) \in h \downarrow c} (f(m, G_{f,h}(n, b)))$  and the family of maps:

$$e_{m,n,b} : \text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(m, G_{f,h}(n, b)) \rightarrow \coprod_{(b, h(b)=c) \in h \downarrow c} (f(m, G_{f,h}(n, b)))$$

which are again injections into the coproduct.

However, we want our structure map in the  $f$ -algebra  $(G_{f,h}, in)$  to be the following:

$$in_{m,n} : \exists (b : |\mathcal{C}|). \text{Hom}_{|\mathcal{C}|}(h(b), c) \times f(m, G_{f,h}(n, b)) \rightarrow G_{f,h}(m \otimes n)$$

But, the coend in the structure of the source object does not match the structure required for the structure map of a graded  $f$ -algebra. We can fix this problem by defining the following functor:

$$\begin{aligned} K_{f,h} : \mathcal{E} \times [|\mathcal{C}|, \mathcal{C}] &\longrightarrow [|\mathcal{C}|, \mathcal{C}] \\ K_{f,h}(m, g) &= \exists(b : |\mathcal{C}|). \text{Hom}_{|\mathcal{C}|}(h(b), -) \times f(m, g(b)) \end{aligned}$$

Using this functor we can see that  $\text{in}$  is equivalent to:

$$\text{in}_{m,n} : K_{f,h}(m, G_{f,h}(n)) \longrightarrow G_{f,h}(m \otimes n)$$

Thus, we now have the following interpretation:

- $h$  maps to a functor  $h : |\mathcal{C}| \longrightarrow |\mathcal{C}|$ .
- $f$  maps to the functor  $K_{f,h} : \mathcal{E} \times [|\mathcal{C}|, \mathcal{C}] \longrightarrow [|\mathcal{C}|, \mathcal{C}]$ .
- $G \circ f \circ h$  maps to a functor  $G_{f,h} : \mathcal{E} \longrightarrow [|\mathcal{C}|, \mathcal{C}]$ .
- $G\text{Con}$  maps to the natural transformation:

$$\text{in}_{m,n} : K_{f,h}(m, G_{f,h}(n)) \longrightarrow G_{f,h}(m \otimes n)$$

making  $(G_{h,f}, \text{in})$  a graded initial  $K_{h,f}$ -algebra.

This shows that graded GADTs correspond to graded initial algebras that can be computed using coproducts and the existential quantifier.

## 3.2 Initial Algebra Packages for Basic Graded GADTs

## References