Graded Generalized Algebraic Data Types

Harley Eades

October 16, 2022

Abstract

Write abstract

1 Introduction

Write the intro.

2 The Fundamental Theory

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

Definition 2.1 (Graded F-Algebra)**.** *For a functor* $\mathsf{F} : \mathcal{E} \times \mathcal{C} \longrightarrow \mathcal{C}$*, a graded* F*-algebra is a pair* (A, h) *that consists of a functor* $\mathsf{A} : \mathcal{E} \longrightarrow \mathcal{C}$ *and a family* h *of morphisms:*

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

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

$$\alpha : (\mathsf{A}, h) \longrightarrow (\mathsf{B}, h')$$

is defined as a natural transformation $\alpha : \mathsf{A}_1 \longrightarrow \mathsf{A}_2$ *such that:*

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

Definition 2.2. *If the category of graded* F*-algebras has an initial object, then we call this a* ***graded initial*** F*-algebra denoted by* $(\mu\mathsf{F}, \mathsf{in})$*. That is, for any other* F*-algebra* (A, h) *there must be a unique morphism* $\alpha : (\mu\mathsf{F}, \mathsf{in}) \longrightarrow (\mathsf{A}, h)$*, but this implies that for any object* n*,* $\alpha_n : \mu\mathsf{F}(n) \longrightarrow \mathsf{A}(n)$ *is unique and* $\mu\mathsf{F}(n)$ *is an initial object in* $\mathcal{C}$*.*

Lemma 2.3 (From Structures to Homomorphisms)**.** *Given any graded* F*-algebra* (A, h)*, the structure map* h *is also a homomorphism between* F*-algebras* $(\mathsf{F}(m, A(-)), \mathsf{F}(-, h_{m,-}))$ *and* $(\mathsf{A}(m \otimes -)), h_{-,m\otimes -})$*.*

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

Lemma 2.4 (Graded Lambek's Lemma)**.** *If* (A, h) *is a graded initial algebra of* F*, then for any object* m*,* $\mathsf{A}(m \otimes -) : \mathcal{E} \longrightarrow \mathcal{C}$ *is isomorphic to* $\mathsf{F}(m, A(-)) : \mathcal{E} \longrightarrow \mathcal{C}$ *via* h_m*.*

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

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

Then by initiality there exists an F-algebra homomorphism

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

such that:

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

We also know that $h_m : \mathsf{F}(m, A(-)) \longrightarrow \mathsf{A}(m \otimes -)$ is itself a graded F-algebra homomorphism (Lemma 2.3). Thus, since we know that $\mathsf{A}(m \otimes n)$ is an initial object by definition and assumption that A is a graded initial object, and hence, $i_{m,n}; h_{m,n} = \mathsf{id}_{m\otimes n}$.

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

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

but again by initiality we know that

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

Therefore, i is the inverse of h and we obtain our result. □

Definition 2.5 (Graded Folds)**.** *Suppose* $(\mu\mathsf{F}, \mathsf{in})$ *is a graded initial* F*-algebra. Then the unquie map between* $\mu\mathsf{F}$ *and any other graded* F*-algebra* (A, h) *is the* ***fold*** *for* $\mu\mathsf{F}$ *and is denoted by*

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

Furthermore, we know that the following must hold:

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

Definition 2.6 (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:*

$$\begin{aligned} \mathsf{U}_\mathsf{F}(\mathsf{A}, h) &= \mathsf{A} \\ \mathsf{U}_\mathsf{F}(\alpha) &= \alpha \end{aligned}$$

Given an object of $[\mathcal{E}, \mathcal{C}]$*, say* X*, then a* U_F***-cone for*** X *comprises, for every graded* F*-albebra* (A, h)*, a natural transformation* $v_{(\mathsf{A},h)} : \mathsf{X} \longrightarrow \mathsf{A}$ *in* $[\mathcal{E}, \mathcal{C}]$ *such that, for every graded* F*-algebra homomorphism* $\alpha : \mathsf{A} \longrightarrow \mathsf{B}$*, we have* $v_{(\mathsf{B},g)} = \alpha; v_{(\mathsf{A},h)}$*. We denote these cones by* (X, v) *and call* X *its* ***vertex*** *and* $v_{(A,h)}$ *the* ***projection*** *from* X *to* A*.*

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

Lemma 2.7 (Forgetful Limits are Unique)**.** *If* (X, v) *is a* U_F*-limit, then it is unique up to isomorphism.*

Proof. Suppose (Y, v') is another U_F-limit. Thus, there is a unique U_F-cone morphism $i : \mathsf{Y} \longrightarrow \mathsf{X}$ such that $i; v = v'$. But, there must also be a unique U_F-cone morphism $j : \mathsf{X} \longrightarrow \mathsf{Y}$ 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 $i; j = \mathsf{id}_\mathsf{Y}$ and $j; i = \mathsf{id}_\mathsf{X}$, and thus i and j are inverses of each other. □

2.1 Interpretation

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

$$\mathsf{in} : \mathsf{f}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(-)) \longrightarrow \mathsf{G}_{\mathsf{f},\mathsf{h}}(\mathsf{h}(-))$$

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

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

Definition 2.8 (Left Kan Extension)**.** *The left Kan extension of a functor* $\mathsf{F} : \mathcal{C} \longrightarrow \mathcal{D}$ *along a functor* $\mathsf{P} : \mathcal{C} \longrightarrow \mathcal{C}'$ *is, if it exists, a functor* $\mathsf{Lan}_{\mathsf{P}}\mathsf{F} : \mathcal{C}' \longrightarrow \mathsf{D}$ *equipped with a natural isomorphism:*

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

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

If we can define $\mathsf{Lan}_{\mathsf{h}}\mathsf{f}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(-))$ and its associated natural isomorphism then we can simply apply the latter to 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:

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

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

Definition 2.9 (Cowedge). *Suppose* $\mathsf{F} : \mathcal{C}^{op} \times \mathcal{C} \longrightarrow \mathcal{D}$ *is a functor. A* ***cowedge*** $e : \mathcal{F} \longrightarrow w$ *is an object* w *and a family of maps* $e_c : \mathsf{F}(c,c) \longrightarrow w$ *for each* c*, such that given any other morphism* $f : c' \longrightarrow c$*, the following holds:*

$$\mathsf{F}(f,f); e_{c'} = \mathsf{F}(\mathsf{id}_{c'}, f); e_c$$

Cowedges are also perserved by composition, that is given a cowedge e : $F \longrightarrow w$ *and a map* $f : w \longrightarrow v$*, then* $e;f : v \longrightarrow F$ *is a cowedge.*

Definition 2.10 (Coend). *Suppose* $\mathsf{F} : \mathcal{C}^{op} \times \mathcal{C} \longrightarrow \mathcal{D}$ *is a functor. A* ***coend*** *of* F *denoted* $\exists(c : \mathcal{C}).\mathsf{F}(c,c)$ *is a universal cowedge* $e : \mathsf{F} \longrightarrow w$ *where every other cowedge* $e' : \mathsf{F} \longrightarrow w'$ *factors through* e *via a unique map* $w \longrightarrow w'$.

The functor we wish to take the coend of is:

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

for any object $c \in |\mathcal{C}|$. Thus, we need to find an object $w \in |\mathcal{C}|$ and an universal cowedge $e : \mathsf{Hom}_{|\mathcal{C}|}(h(-)), c) \times \mathsf{f}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(-)) \longrightarrow w$. As it turns out, what we essentially want w to be is the disjoint union of $\mathsf{Hom}_{|\mathcal{C}|}(h(b)), c) \times \mathsf{f}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(b))$ 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 2.11 (Comma Category). *Suppose we have functors* $\mathsf{F} : \mathcal{D} \longrightarrow \mathcal{C}$ *and* $\mathsf{G} : \mathcal{E} \longrightarrow \mathcal{C}$*. Then the arrow category* $\mathsf{F} \downarrow \mathsf{G}$ *consists of:*

- *Objects are triples* $(d \in \mathcal{D}, e \in \mathcal{E}, f : \mathsf{F}d \longrightarrow \mathsf{G}e \in \mathcal{C})$.
- *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:*

$$f_1; \mathsf{G}k = \mathsf{F}h; f_2$$

There is are projection functor:

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

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

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

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

Now if we take the first projection:

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

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

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

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

$$\coprod_{(b, c = \mathsf{h}(b)) \in \mathsf{h} \downarrow c} (\mathsf{f}(\mathsf{G}_{\mathsf{f,h}}(b)))$$

with injections:

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

which can be written:

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

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

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

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

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

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

References