Graded Generalized Algebraic Data Types

Harley Eades

October 15, 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 $(\mu\mathsf{F}, \mathrm{in})$. That is, for any other F-algebra (A, h) there must be a unique morphism $\alpha : (\mu\mathsf{F}, \mathrm{in}) \longrightarrow (\mathsf{A}, h)$, 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. $\qquad\square$

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:

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

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, $i_{m,n}; h_{m,n} = \mathrm{id}_{m \otimes n}$.

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

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

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

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

$$U_F(A, h) = A$$
$$U_F(\alpha) = \alpha$$

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* $v_{(A,h)} : X \longrightarrow A$ *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 U_F-***cone morphism** $g : (X, v_1) \longrightarrow (Y, v_2)$ *is a natural transformation* $g : X \longrightarrow Y$ *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 $i : Y \longrightarrow X$ such that $i; v = v'$. But, there must also be a unique U_F-cone morphism $j : X \longrightarrow Y$ 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. $\square$

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 $f : [|\mathcal{C}|, \mathcal{C}] \longrightarrow [|\mathcal{C}|, \mathcal{C}]$.

- `h` maps to a functor $h : |\mathcal{C}| \longrightarrow |\mathcal{C}|$.

- `G f h` maps to a functor $G_{f,h} : |\mathcal{C}| \longrightarrow \mathcal{C}$.

- `GCon` maps to a natural transformation:

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

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

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* $F : \mathcal{C} \longrightarrow \mathcal{D}$ *along a functor* $P : \mathcal{C} \longrightarrow \mathcal{C}'$ *is, if it exists, a functor* $\mathsf{Lan}_P F : \mathcal{C}' \longrightarrow D$ *equipped with a natural isomorphism:*

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

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

If we can define $\mathsf{Lan}_h f(G_{f,h}(-))$ 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.

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

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

4

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* $e' : w' \longrightarrow \mathsf{F}$ *factors through e via a unique map* $w' \longrightarrow w$.

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 $(b \in |\mathcal{C}|, \mathrm{id}_b : \mathsf{h}(b) \longrightarrow b \in |\mathcal{C}|)$.

- Morphisms $(b_1, \mathrm{id}_{b_1}) \longrightarrow (b_2, \mathrm{id}_{b_2})$ are morphisms $f : b_1 \longrightarrow b_2 \in |\mathcal{C}|$, but this implies that $f = \mathrm{id}_{b_1}$.

In the objects above we write the identities as $\mathrm{id}_b : \mathsf{h}(\mathsf{b}) \longrightarrow c$ 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 $\mathsf{f}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(-))$ as follows:

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

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 $\coprod_{b \in |\mathcal{C}|}(h(b) = c, f(G_{f,h}(b)))$.

Suppose we have an object c. Then define the functor:

$$F : |\mathcal{C}|^{\mathrm{op}} \times |\mathcal{C}| \longrightarrow \mathsf{Set} \times \mathcal{C}$$
$$F(b,b) = \mathsf{Hom}_{|\mathcal{C}|}(c, h(b)) \times f(G_{f,h}(b))$$

$$\mathsf{Lan}_h f(G_{f,h}(c)) \quad = \mathrm{colim}(h \downarrow c \xrightarrow{\Pi^c} |\mathcal{C}| \xrightarrow{f(G_{f,h}(-))} \mathcal{C})$$
$$= \exists(b : |\mathcal{C}|).\mathsf{Hom}_{|\mathcal{C}|}(c, h(b))) \times f(G_{f,h}(b))$$

$$f(G_{f,h}(-)) : |\mathcal{C}|^{\mathrm{op}} \longrightarrow \mathcal{C}$$
$$f(G_{f,h}(b)) = \mathsf{Hom}_{[|\mathcal{C}|^{\mathrm{op}}, \mathcal{C}]}(\mathsf{Point}_b, f(G_{f,h}(-)))$$

References