Graded Generalized Algebraic Data Types

Harley Eades and Dominic Orchard

October 25, 2022

Abstract

Write abstract

Contents

1 Introduction

Write the intro.

2 Graded GADTs Syntactically

2.1 Graded GADTs

```
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)
    GCon3 :: f3 m3 (G n3 f3 h3) a -> G (m3 * n3) f3 h3 (h3 a)
    ...
    GConi :: fi mi (G ni fi hi) a -> G (mi * ni) fi hi (hi a)
```

2.2 Fold and Build

$$\frac{\begin{array}{l}\mathsf{X} : \mathcal{E} \to * \to * \\ h : \forall n, m : \mathcal{E}.\mathsf{K}(m, \mathsf{X}(n)) \to \mathsf{X}(m \otimes n)\end{array}}{(\mathsf{X}, h) : \mathsf{K}\text{-}\mathsf{GradeAlg}} \qquad \frac{(\mathsf{X}, h) : \mathsf{K}\text{-}\mathsf{GradeAlg} \qquad n : \mathcal{E}}{\mathsf{fold}^*_{\mathsf{K},\mathsf{X}}(h, n) : \mu^* \mathsf{K}(n) \to \mathsf{X}(n)}$$

$$(\mathsf{X}, h_1) : \mathsf{K}\text{-}\mathsf{GradeAlg} \qquad (\mathsf{Y}, h_2) : \mathsf{K}\text{-}\mathsf{GradeAlg}$$

3 The Fundamental Theory

This is all based on the initial algebra semantics for GADTs [2].

3.1 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 $\mathsf{f} : [|\mathcal{C}|, \mathcal{C}] \longrightarrow [|\mathcal{C}|, \mathcal{C}]$.
- `h` maps to a functor $\mathsf{h} : |\mathcal{C}| \longrightarrow |\mathcal{C}|$.

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

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

At this point, we can see a problem, we want $(\mathsf{G_{f,h}}, \mathsf{in})$ to be an initial f-algebra, but in has a target that does not fit the proper form, because it is currently $\mathsf{G_{f,h}}(\mathsf{h}(-))$, 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 $\mathsf{G_{f,h}}(-)$.

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

Definition 3.1 (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* $\mathsf{Lan_P F} : \mathcal{C}' \longrightarrow \mathsf{D}$ *equipped with a natural isomorphism:*

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

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

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

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:

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

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

Definition 3.2 (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 3.3 (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 3.4 (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 $\mathsf{id}_b : \mathsf{h}(\mathsf{b}) \longrightarrow c$ 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 $(b, \mathsf{h}(b) = c)$ yields an object $\mathsf{f}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(b))$. Now since we are dealing with a discrete category we can take the colimit of this functor to obtain the coproduct we denote as

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

with injections:

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

which can be written:

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

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

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

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.

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 $\mathsf{f}(\mathsf{G}_{\mathsf{h},\mathsf{f}}(-))$ using the coend and coproducts. Now we can obtain an isomorphic version of `GCon` using the following fact about Left Kan extensions and coends:

$$\begin{array}{ll} & \mathsf{Hom}_?(\exists(b:|\mathcal{C}|).\mathsf{Hom}_{|\mathcal{C}|}(h(b),a)\times\mathsf{f}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(b)),\mathsf{G}_{\mathsf{h},\mathsf{f}}(a)) \\ \cong & \mathsf{Hom}_?(\mathsf{Lan}_{\mathsf{h}}\mathsf{f}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(a)),\mathsf{G}_{\mathsf{h},\mathsf{f}}(a)) \\ \cong & \mathsf{Hom}_?(\mathsf{f}(\mathsf{G}_{\mathsf{h},\mathsf{f}}(a)),\mathsf{G}_{\mathsf{h},\mathsf{f}}(\mathsf{h}(a))) \end{array}$$

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

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

This can be fixed by defining the following functor:

$$\begin{array}{l} \mathsf{K}_{\mathsf{f},\mathsf{h}} : [|\mathcal{C}|,\mathcal{C}]\longrightarrow[|\mathcal{C}|,\mathcal{C}] \\ \mathsf{K}_{\mathsf{f},\mathsf{h}}(\mathsf{g},c) = \exists(b:|\mathcal{C}|).\mathsf{Hom}_{|\mathcal{C}|}(h(b),c)\times\mathsf{f}(c,\mathsf{g}(b)) \end{array}$$

Using this functor we can see that in is equivalent to:

$$\mathsf{in}_c : \mathsf{K}_{\mathsf{f},\mathsf{h}}(\mathsf{G}_{\mathsf{f},\mathsf{h}}(c))\longrightarrow\mathsf{G}_{\mathsf{f},\mathsf{h}}(c)$$

Thus, $(\mathsf{G}_{\mathsf{f},\mathsf{h}})$ is an initial $\mathsf{K}_{\mathsf{f},\mathsf{h}}$-algebra.

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

```
data G f h a where
    GCon :: exists b.(Eql (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 $\mathsf{h} : |\mathcal{C}|\longrightarrow|\mathcal{C}|$.
- `f` maps to the functor $\mathsf{K}_{\mathsf{f},\mathsf{h}} : [|\mathcal{C}|,\mathcal{C}]\longrightarrow[|\mathcal{C}|,\mathcal{C}]$.

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

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

  making $(\mathsf{G}_{\mathsf{h},\mathsf{f}}, \mathsf{in})$ an initial $\mathsf{K}_{\mathsf{f},\mathsf{h}}$-algebra.

We can generalize this semantics to the graded case.

3.2 The Graded Case

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

Definition 3.5 (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:

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

Lemma 3.6 (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. □

Definition 3.7. *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 3.8 (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 3.6). 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. □

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

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

  $$\mathsf{in}_{m,n} : \mathsf{f}(m, \mathsf{G}_{\mathsf{f},\mathsf{h}}(n, b)) \longrightarrow \mathsf{G}_{\mathsf{f},\mathsf{h}}(m \otimes n, \mathsf{h}(b))$$

  making $(\mathsf{G}_{\mathsf{h},\mathsf{f}}, \mathsf{in})$ a graded initial f-algebra.

But, we have the same problem as before where the target of $\mathsf{in}_{m,n}$ 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:

$$\mathsf{f}(m, \mathsf{G}_{\mathsf{f},\mathsf{h}}(n,-)) : |\mathcal{C}| \longrightarrow \mathcal{C}$$

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

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

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

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

which are again injections into the coproduct.

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

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

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:

$$\begin{array}{l}\mathsf{K}_{\mathsf{f},\mathsf{h}} : \mathcal{E} \times [|\mathcal{C}|, \mathcal{C}] \longrightarrow [|\mathcal{C}|, \mathcal{C}] \\ \mathsf{K}_{\mathsf{f},\mathsf{h}}(m, \mathsf{g}) = \exists(b : |\mathcal{C}|).\mathsf{Hom}_{|\mathcal{C}|}(h(b), -) \times \mathsf{f}(m, \mathsf{g}(b))\end{array}$$

Using this functor we can see that in is equivalent to:

$$\mathsf{in}_{m,n} : \mathsf{K}_{\mathsf{f},\mathsf{h}}(m, \mathsf{G}_{\mathsf{f},\mathsf{h}}(n)) \longrightarrow \mathsf{G}_{\mathsf{f},\mathsf{h}}(m \otimes n)$$

Thus, we now have the following interpretation:

- `h` maps to a functor $\mathsf{h} : |\mathcal{C}| \longrightarrow |\mathcal{C}|$.
- `f` maps to the functor $\mathsf{K}_{\mathsf{f},\mathsf{h}} : \mathcal{E} \times [|\mathcal{C}|, \mathcal{C}] \longrightarrow [|\mathcal{C}|, \mathcal{C}]$.

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

  $$\mathsf{in}_{m,n} : \mathsf{K}_{\mathsf{f,h}}(m, \mathsf{G}_{\mathsf{f,h}}(n)) \longrightarrow \mathsf{G}_{\mathsf{f,h}}(m \otimes n)$$

  making $(\mathsf{G}_{\mathsf{h,f}}, \mathsf{in})$ a graded initial $\mathsf{K}_{\mathsf{h,f}}$-algebra.

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

3.3 Initial Algebra Packages for Basic Graded GADTs

Throughout this section we assume $(\mu\mathsf{K}, \mathsf{in})$ is the initial graded K-algebra.

3.3.1 Graded Folds and Fold Fusion

Every graded GADT has an assocated graded fold combinator associated with it.

Definition 3.9 (Graded Folds)**.** *The unquie map between* $\mu\mathsf{K}$ *and any other graded* K*-algebra* (G, h) *is the* ***fold*** *for* $\mu\mathsf{K}$ *and is denoted by*

$$\mathsf{fold} : [\mathsf{K}(m, \mathsf{G}(n)), \mathsf{G}(m \otimes n)] \longrightarrow [\mu\mathsf{K}, \mathsf{G}]$$

Furthermore, we know that the following must hold:

$$\begin{tikzcd}
\mathsf{K}(m, \mu\mathsf{K}(n)) \arrow[r, "\mathsf{K}(\mathsf{id}_m, \mathsf{fold}(h,n))"] \arrow[d, "\mathsf{in}_{m,n}"'] & \mathsf{K}(m, \mathsf{G}(n)) \arrow[d, "h"] \\
\mu\mathsf{K}(m \otimes n) \arrow[r, "\mathsf{fold}(h, m\otimes n)"] & \mathsf{G}(m \otimes n)
\end{tikzcd}$$

Thus, $\mathsf{fold}(h) : \mu\mathsf{K} \longrightarrow \mathsf{G}$ *is a graded* K*-algebra homomorphism.*

Since $\mathsf{fold}(h) : \mu\mathsf{K} \longrightarrow \mathsf{G}$ is a map from the initial graded K-algebra to (G, h) we know it must be unique by initiality. This is the basis for fold fusion which says that given a graded K-algebra homomorphism $\psi : (\mathsf{G}, h) \longrightarrow (\mathsf{G}', h')$ the following diagram commutes:

$$\begin{tikzcd}
 & \mathsf{G} \arrow[dr, "\psi"] & \\
\mu\mathsf{K} \arrow[ur, "\mathsf{fold}(h)"] \arrow[rr, "\mathsf{fold}(h')"] & & \mathsf{G}'
\end{tikzcd}$$

But, this easily follows from the uniqueness of $\mathsf{fold}(h)$ and $\mathsf{fold}(h')$.

3.3.2 Graded Fold/Build Fusion Rules

The fold/build fusion rules for GADTs allow for the optimization of programs over data structures. In this section we explain how graded fold/build fusion can be modeled in graded initial algebra semantics. We first will need a new universal property for graded initial algebras in terms of a limit over $\mathsf{U_F}$ cones.

Definition 3.10 (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:*

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

Given an object of $[\mathcal{E},\mathcal{C}]$*, say* X*, then a* $\mathsf{U_F}$***-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:*

$$\begin{tikzcd} & A \arrow[dr, "\alpha"] & \\ X \arrow[ur, "v_{(\mathsf{A},h)}"] \arrow[rr, "v_{(\mathsf{B},g)}"] & & B \end{tikzcd}$$

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

A $\mathsf{U_F}$***-cone morphism*** $\gamma : (\mathsf{X},v_1) \longrightarrow (\mathsf{Y},v_2)$ *is a natural transformation* $\gamma : \mathsf{X} \longrightarrow \mathsf{Y}$ *such that for any graded* F*-algebra* (A,h)*, we have:*

$$\begin{tikzcd} & \mathsf{Y} \arrow[dr, "v_2"] & \\ \mathsf{X} \arrow[ur, "\gamma"] \arrow[rr, "v_1"] & & \mathsf{A} \end{tikzcd}$$

A $\mathsf{U_F}$***-limit*** *is a* $\mathsf{U_F}$*-cone to which there is a unique* $\mathsf{U_F}$*-cone morphism, call the* ***mediating morphism****, from any other* $\mathsf{U_F}$*-cone.*

It is also the case that graded $\mathsf{U_K}$-limit's are unique up to isomorphism.

Lemma 3.11 (Forgetful Limits are Unique)**.** *If* (X, v) *is a* $\mathsf{U_K}$*-limit, then it is unique up to isomorphism.*

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

We denote a chosen graded $\mathsf{U_K}$-limit by $(\mu^*\mathsf{K}, \mathsf{fold}^*_\mathsf{K})$ and $\mathsf{build}^*_\mathsf{K}$ for the mediating map from some other $\mathsf{U_K}$-cone (X, v).

Every graded initial K-algebra is also a graded $\mathsf{U_F}$-limit.

Lemma 3.12 (Initial Algebras are also Limits)**.**

i. If there is a graded initial K*-algebra* $(\mu\mathsf{K}, \mathsf{in})$ *for a functor* $\mathsf{K} : \mathcal{E} \times \mathcal{C} \longrightarrow \mathcal{C}$*, then* $\mu\mathsf{K}$ *is the vertex of a graded* $\mathsf{U_K}$*-limit.*

ii. If there is a graded $\mathsf{U_K}$*-limit* $(\mu^*\mathsf{K}, \mathsf{fold}^*_\mathsf{K})$*, then* $\mu^*\mathsf{K}$ *is the carrier of a graded initial* K*-algebra.*

Proof. This proof follows from the proof of the same result for the non-graded case originally given in Proposition 4 and Proposition 5 of [1]. □

If there is time, add the dinaturality from [1]

References

[1] Neil Ghani, Tarmo Uustalu, and Varmo Vene. Build, augment and destroy, universally. In Wei-Ngan Chin, editor, *Programming Languages and Systems*, pages 327–347, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[2] Patricia Johann and Neil Ghani. Initial algebra semantics is enough! In Simona Ronchi Della Rocca, editor, *Typed Lambda Calculi and Applications*, pages 207–222, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.