# GATs and CwFs

Apiros3

First Version : September 25, 2025
Last Update : – –, 2025

## Contents

# 1 Preliminaries

## 1.1 Algebras

### 1.1.1 Ambient Structure

Fix a base category $\mathcal{C}$, and a split fibration $p : \mathcal{E} \to \mathcal{C}$. Over the type theory setting, the base category corresponds to `Set`, and the fibration corresponds to families $P : A \to$ `Type`, or just predicates.

**Definition 1.1.1.** *Given an endofunctor $F : \mathcal{C} \to \mathcal{C}$, a **truth-preserving fibred lift** is a lift $\hat{F} : \mathcal{E} \to \mathcal{E}$ such that the following commutes:*

$$\begin{tikzcd} \mathcal{C} \arrow[r, "F"] & \mathcal{C} \\ \mathcal{E} \arrow[u, "p"] \arrow[r, "\hat{F}"'] & \mathcal{E} \arrow[u, "p"'] \end{tikzcd}$$

This lifting corresponds to one inductive hypotheses per recursive position over the type theory setting.

### 1.1.2 $F$-algebra

**Definition 1.1.2.** *An $F$**-algebra** is a pair $(A, \alpha)$ with*

$$\alpha : F(A) \to A$$

Over the type theoretic setting, this corresponds to a handler for one constructor step. Over haskell, we can see something like:

```haskell
class Functor f ⇒ Algebra f a where alg :: f a → a
cata :: Algebra f a ⇒ Fix f → a
cata (In t) = alg (fmap cata t)
```

**Definition 1.1.3.** *A **morphism** of algebras*

$$f : (A, \alpha) \to (B, \beta)$$

*is a map $f : A \to B$ in $\mathcal{C}$ such that the following square commutes:*

$$\begin{tikzcd} F(A) \arrow[r, "F(f)"] \arrow[d, "\alpha"'] & F(B) \arrow[d, "\beta"] \\ A \arrow[r, "f"'] & B \end{tikzcd}$$

*As notation, we write $\mathrm{Alg}(F)$ for the category of such algebras/morphisms.*

**Definition 1.1.4.** *$(\mu F, \mathrm{in})$ is **initial** in $\mathrm{Alg}(F)$ if for every $(B, \beta)$ there is a unique algebra morphism*

$$\mathrm{fold}_\beta : \mu F \to B$$

At the type theory level, $(\mu F, in)$ corresponds to the inductive type and its constructors. Then, $\mathrm{fold}_\beta$ is the non-dependent eliminator / recursor. The square is exactly the $\beta$-rule, and the uniqueness of fold is the $\eta$-rule. The square is exactly the statement that map and fold commute.

### 1.1.3 Displayed (dependent) $F$-algebra

Fix a base algebra $(A, \alpha)$.

**Definition 1.1.5.** *A* ***displayed*** $F$***-algebra*** *over* $(A, \alpha)$ *is*

- *an object* $P \in \mathcal{E}_A$
- *a* ***dependent structure map*** *lying over* $\alpha$ *with*

$$\hat{\alpha} : \hat{F}_A(P) \to P$$

*with* $p(\hat{\alpha}) = \alpha$*, where* $p$ *is the projection from* $\hat{F}_A(P)$ *to* $P$*.*

*That is,* $\hat{\alpha}$ *is a morphism in the fibre* $\mathcal{E}_A$ *lying over* $\alpha$*.*

Intuitively, $P$ first states the goal to prove for each element of $A$. $\hat{F}_A(P)$ packages exactly the induction hypotheses needed at one constructor step, and $\hat{\alpha}$ is the method that consumes these hypotheses to produce a $P$-witness at the result of that constructor step.

From a type theory perspective $P : A \to \texttt{Type}$ is the *motive*, $\hat{\alpha}$ packages one *method* per constructor, consuming inductive hypotheses ($\hat{F}_A(P)$).

We consider some examples:

**Example 1.1.6.** The naturals with functor $F(X) = 1 + X$ comes with the baes algebra $(\mathbb{N}, \mathrm{in} = [0, \mathsf{s}])$.

The lift is given by

$$\hat{F}_{\mathbb{N}}(P)(\mathrm{inl}(*)) \cong 1 \qquad \hat{F}_{\mathbb{N}}(P)(\mathrm{inr}(n)) \cong P(n)$$

Then the displayed algebra for some $P : \mathbb{N} \to \texttt{Set}$ is given by

$$\hat{\alpha}(\mathrm{inl}(*), \_) : 1 \to P(0) := \text{pick } p_0 \qquad \hat{\alpha}(\mathrm{inr}(n), -) : P(n) \to P(\mathsf{s}\ n) := p_s(n, -)$$

### 1.1.4 Displayed Morphism over an Algebra

Fix an algebra $(A, \alpha), (B, \beta)$ and an algebra morphism $f : (A, \alpha) \to (B, \beta)$. Let $(P, \hat{\alpha})$ and $(Q, \hat{\beta})$ be the displayed algebras over $(A, \alpha)$ and $(B, \beta)$ respectively.

**Definition 1.1.7.** *A* ***displayed morphism over*** $f$ *is a fibre map*

$$\overline{f} : P \to f^*(Q)$$

*in* $\mathcal{E}_A$*, such that the following* ***displayed square*** *commutes:*

$$\begin{tikzcd}
\hat{F}_A(P) \arrow[r, "\hat{F}_A(\overline{f})"] \arrow[d, "\hat{\alpha}"'] & \hat{F}_A(f^*(Q)) \arrow[d, "f^*(\hat{\beta})"] \\
P \arrow[r, "\overline{f}"'] & f^*(Q)
\end{tikzcd}$$

We make sure this type checks by:

$$\begin{tikzcd}
\hat{F}_A(P) \arrow[r, "\hat{F}_A(\overline{f})"] \arrow[d, "\hat{\alpha}"'] & \hat{F}_A(f^*(Q)) \arrow[r, leftrightarrow, "\cong"] & (Ff)^*\hat{F}_B(Q) \arrow[d, "f^*(\hat{\beta})"] & \hat{F}_B(Q) \arrow[l, "(Ff)^*"'] \arrow[d, "\hat{\beta}"] \\
P \arrow[rr, "\overline{f}"'] & & f^*(Q) & Q \arrow[l, "f^*"]
\end{tikzcd}$$

Intuitively, $\overline{f}$ transports $P$-evidence on $A$ to $Q$-evidence on $B$ along $f$ and does so constructor-by-constructor compatibly with the methods. This is the naturality / fusion condition for logical predicates.

**Definition 1.1.8.** *Let* $\mathrm{DAlg}_A(F, \alpha)$ *denote the category whose*

- *objects are displayed algebras* $(P, \hat{\alpha})$
- *morphisms* $\theta : (P, \hat{\alpha}) \to (P', \hat{\alpha'})$ *are maps* $\theta : P \to P'$ *with*

$$\begin{tikzcd} \hat{F}_A(P) \arrow[r, "\hat{F}_A(\theta)"] \arrow[d, "\hat{\alpha}"'] & \hat{F}_A(P') \arrow[d, "\hat{\alpha}'"] \\ P \arrow[r, "\theta"'] & P' \end{tikzcd}$$

Displayed morphisms compose over composition of base morphisms and identities are given by the fibre identities. This gives the projection

$$p^F : \mathrm{DAlg}(F) \to \mathrm{Alg}(F)$$

with the structure of a fibration, where objects over $(A, \alpha)$ are displayed algebras and arrows over $f$ are displayed morphisms.

### 1.2 Substitution

Viewing **Con** as the metacategory whose objects are contexts and whose morphisms are context morphisms, substitutions correspond exactly to context morphisms. Intuitively, a substitution $\sigma : \Delta \to \Gamma$ gives information about how to interpret each variable/assumption in $\Gamma$ as a term built in $\Delta$.

## 2 Correspondence

### 2.1 Generalized Algebraic Theory

### 2.2 Category with Families

### 2.3 Quotient Inductive Inductive Types

### 2.4 Correspondence

Intuitively, a GAT's syntax is the internal language of some CwF, and every CwF is up to equivalence the classifying/category of contexts of some GAT.

Given a GAT $T$,