Initial-algebra semantics with structural operations

Tom Hirschowitz and Ambroise Lafont

November 15, 2021
Motivation

- We “categorists” are very pleased with our tools for syntax with binding: nominal sets, Fiore-Plotkin-Turi (FPT), etc.
- Truth: need a lot more work.
  - Complex type systems (Gratzer-Sterling).
  - Linearity: sparsely studied (Power, Tanaka).
  - Structural operations on syntax, beyond capture-avoiding substitution: never heard of.
- This talk is about structural operations.
  - Evaluation contexts + potentially capture-allowing context application $\mathcal{E}[e]$.
  - Partial differentiation in differential $\lambda$-calculus.
  - ... and more!

Warning: work in progress!!!
A simple language with evaluation contexts

Syntax:

\[ e, f ::=} x \mid e \ f \mid \lambda x. e \]
\[ E ::=} \Box \mid E \ e \]

Context application:

\[ \Box[e] = e \]
\[ (E \ e)[f] = E[f] e. \]

How would a categorist define this language?
Definition by equational systems

Rough idea:

• build context application into the syntax, and
• incorporate its definition as equations.

\[ e, f ::= x | e \cdot f | \lambda x. e | E[e] \]
\[ E ::= \square | E \cdot e \]

modulo

\[ \square[e] = e \]
\[ (E \cdot e)[f] = E[f] \cdot e. \]

Problem

We get an explicit construction with a \textit{quotient}, instead of the expected, inductive one.
FPT show that the syntax actually satisfies both

- its usual, lightweight induction principle, and
- a refined, heavier one which incorporates substitution.

Crucially, the simpler principle gives the desired construction of the initial algebra.

Let us sketch this viewpoint, and then abstract over it.
Our model pattern: FPT

(and others with them, as in Ambroise’s talk about De Bruijn monads)

- Syntax as initial algebra $\mu A.(I + \Sigma(A))$.
- Bonus property: also initial algebra with substitution, given
  \[
  \Sigma(A) \otimes B \to \Sigma(A \otimes B).
  \]
  Such algebras are called $\Sigma$-monoids: category $\Sigma$-Mon.
- Categorically: the forgetful functor
  \[
  (I + \Sigma) \text{-alg} \leftarrow \Sigma \text{-Mon} \rightarrow [F, \text{Set}]
  \]
  creates the initial object.
- Equivalently, for the induced monads, say $(I + \Sigma)^*$ and $\Sigma^\otimes$, and monad morphism
  \[
  \alpha : (I + \Sigma)^* \to \Sigma^\otimes,
  \]
  $\alpha_0 : (I + \Sigma)^*(0) \to \Sigma^\otimes(0)$ is an isomorphism.
### Definition

A monad morphism $\alpha : R \to S$ is **admissible** when $\alpha_0$ is an isomorphism. We also sometimes say that $S$ is an **admissible extension** of $R$.

### Goal in this talk

Design **signatures** for admissible monad morphisms.

### Remark

*In agdaian, a *universe* of admissible monad morphisms?*
## Distributive laws

### Definition

A (monad) distributive law of $R$ over $T$ is a natural transformation $\delta : TR \to RT$, compatible with the units and multiplications of $R$ and $T$.

### Proposition (Beck)

Any distributive law $\delta : TR \to RT$ makes $RT$ into a monad with

- **unit**
  $$C \xrightarrow{\eta^T} C \xleftarrow{T} C \xrightarrow{\eta^R} C$$

- **multiplication**
  $$RT \xrightarrow{R\delta T} RRT \xrightarrow{\mu^R \mu^T} RT.$$
Admissible extensions from distributive laws

**Proposition**

For any distributive law \( \delta : TR \to RT \), if \( T(0) \cong 0 \) then \( RT \) is an admissible extension of \( R \).

More precisely, if \( 0 \xrightarrow{\eta_0^T} T(0) \) is an isomorphism, then the monad morphism

\[
R \xrightarrow{R\eta^T} RT
\]

is admissible.

**Proof.**

Obvious.

- All of our applications arise in this way.
- \( \leadsto \) subgoal: design signatures for such distributive laws.
Recalling FPT

- Set up: $[\mathcal{F}, \text{Set}]$, or equivalently $[\text{Set}, \text{Set}]_f$.
  - $\mathcal{F}$ has finite ordinals as objects, with all maps between them as morphisms.
- For $X \in [\text{Set}, \text{Set}]_f$, think of $X(n)$ as “$X$-terms with free variables in $n$.”
- Monoidal structure $(\otimes, I)$ given by composition.
- Monoid structure $X \otimes X \rightarrow X = \text{monad structure} = \text{substitution}:
  \[ X(X(n)) \rightarrow X(n) \]
Recalling FPT

- Syntax specified by endofunctor $\Sigma: \mathcal{[Set, Set]}_f \rightarrow \mathcal{[Set, Set]}_f$, e.g.,

$$\Sigma(X)(n) = X(n)^2 + X(n + 1).$$

- Auxiliary operation, substitution, defined by its commutation with constructors, i.e., $\Sigma(X) \otimes X \rightarrow \Sigma(X \otimes X)$, or more generally

$$st_{X,Y}: \Sigma(X) \otimes Y \rightarrow \Sigma(X \otimes Y)$$

(for pointed $Y$).
Recalling FPT

**Augmented models** $= \Sigma$-monoids $= (I + \Sigma)$-algebras $X$ with $m : X \otimes X \rightarrow X$ such that

\[
\begin{align*}
\Sigma(X) \otimes X & \xrightarrow{st_{X,X}} \Sigma(X \otimes X) \\
\downarrow a & \quad & \downarrow \Sigma(m) \\
X \otimes X & \quad & \Sigma(X)
\end{align*}
\]

(satisfying monoid equations).
Adapting FPT

- Understand $st_{X,Y} : \Sigma(X) \otimes Y \to \Sigma(X \otimes Y)$ as a **structurally inductive** definition of substitution.

- Let $\Gamma_Y(X) = X \otimes Y$
  $\Theta(X) = I + \Sigma(X)$.

- Generalise $\Gamma_Y(\Theta(X)) \to \Theta(\Gamma_Y(X))$ (which requires pointed $Y$) to
  \[
  \Gamma_Y(\Theta(X)) \to \Theta^*(\Gamma_{\Theta^*(Y)}(X) + \Theta^*(Y) + X)
  \]
  (which doesn’t).
\[
\Gamma_Y(\Theta(X)) \rightarrow \Theta^*(\Gamma_{\Theta^*(Y)}(X) + \Theta^*(Y) + X)
\]

- Simple format for inductive definitions, structurally decreasing in \( X \).
- Right-hand side: basic term with free variables in
  - basic terms over \( Y \),
  - \( X \) (for technical reasons),
  - recursive calls, i.e., \( \Gamma_{\Theta^*(Y)}(X) \).
- In a recursive call, only the main argument \( X \) needs to decrease.
- \( \sim \rightarrow \) may use pointedness (and more) of \( \Theta^*(Y) \).
**Structural extensions**

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
</table>

A **structural extension** of an endofunctor $\Sigma: C \to C$ on any locally finitely presentable category $C$ is

- a finitary functor $\Gamma: C^2 \to C$,
- cocontinuous in its first (main) argument, equipped with
- a natural transformation

$$d_{X,Y}: \Gamma_Y(\Sigma(X)) \to S(\Gamma_{S(Y)}(X) + S(Y) + X),$$

where

$$S = \Sigma^*$$

$$\Gamma_Y(X) = \Gamma(X, Y).$$
Theorem

Any structural extension

\[ d_{X,Y} : \Gamma_Y(\Sigma(X)) \to S(\Gamma_{S(Y)}(X) + S(Y) + X) \]

(with again \( S = \Sigma^* \)) induces a monad distributive law

\[ TS \to ST, \]

where

\[ T(X) = (\Gamma_{S\Delta}^*)(X) \]

\[ (\Gamma_{S\Delta})(X) = \Gamma_{S(X)}(X), \]

hence

\[ T(X) = \mu_A . (X + \Gamma_{S\Delta}(A)). \]

Furthermore, \( T(0) \cong 0 \) by cocontinuity of \( \Gamma \), hence \( S \to ST \) is admissible.
Augmented algebras

As a bonus, we may characterise $ST$-algebras.
Augmented algebras

An algebra for a structural extension $E = (\Gamma, d)$ of $\Sigma : C \rightarrow C$ consists of an object $X \in C$, equipped with

- $\Sigma$-algebra structure $a : \Sigma(X) \rightarrow X$ and
- $\Gamma\Delta$-algebra structure $b : \Gamma_X(X) \rightarrow X$,

making the following diagram commute,

\[
\begin{array}{ccc}
\Gamma_X(\Sigma(X)) & \xrightarrow{d_X, X} & SO_XOSX\Gamma_SX(X) \\
\Gamma_X(a) & & \downarrow {SO_XO\bar{a}\Gamma_{\bar{a}}(X)} \\
\Gamma_X(X) & & \downarrow S[X, X, b] \\
& b & \xleftarrow{\bar{a}} X
\end{array}
\]

where $O_X(Z) := X + Z$ and $\bar{a} : S(X) \rightarrow X$ is freely induced by $a$. 
Category of augmented algebras

- A morphism $X \to Y$ of algebras for $E = (\Gamma, d)$ is a morphism between underlying objects which is both a morphism of $\Sigma$- and $\Gamma\Delta$-algebras.
- Let $E\text{-alg}$ denote the category of algebras for $E$, or $E$-algebras.
Characterisation of algebras

Let $E = (\Gamma, d)$ be any structural extension of $\Sigma: \mathcal{C} \to \mathcal{C}$, and let $T = (\Gamma_S \Delta)^*$. Then we have

$$E\text{-alg} \cong ST\text{-Alg}$$

over $\mathcal{C}$, where capital $\text{Alg}$ denotes monad algebras.
Application 1: evaluation contexts

Recall

\[ e, f \ ::= \ x \mid e \ f \mid \lambda x. e \]

\[ E \ ::= \ \Box \mid E \ e \]

with

\[ \Box[e] = e \]

\[ (E\ e)[f] = E[f\ e]. \]
A structural extension for context application

- Ambient category $[\text{Set}, \text{Set}^2]_f$:
  - $X(n)_p = \text{set of programs with } n \text{ free variables}$,
  - $X(n)_c = \text{set of contexts with } n \text{ free variables}$.

- Syntax:
  \[
  \Sigma(X)(n)_p = n + X(n)_p^2 + X(n + 1)_p
  \]
  \[
  (e, f) ::= x \mid e \ f \mid \lambda x. e
  \]
  \[
  \Sigma(X)(n)_c = 1 + X(n)_c \times X(n)_p
  \]
  \[
  (E ::= \Box \mid E \ e)
  \]

- Structural extension: take $\Gamma(X, Y)(n)_p = X(n)_c \times Y(n)_p$ (empty at $c$) with, at $p$:

  \[
  \Sigma(X)(n)_c \times Y(n)_p \to S(\Gamma(X, Y) + S(Y) + X)_p
  \]
  \[
  (\Box, e) \mapsto \eta^S(in_2(\eta^S(e)))
  \]
  \[
  (E \ f, e) \mapsto \eta^S(in_1(E, e)) \eta^S(in_3(f)).
  \]
Application 2: Etonnante modernité de l’addition

- Ambient category $\text{Set}$.  
- $\Sigma(X) = 1 + X$, hence $\Sigma^*(0) = \mathbb{N}$.  
- “Auxiliary” operation: addition.

$$s(x) + y = s(x + y) \quad \quad 0 + y = y$$

where

$$0 := \text{in}_1(\star) \quad \quad s(x) := \text{in}_2(x).$$

From a structural extension

Take $\Gamma(X, Y) = X \times Y$ and, letting $S(X) := \Sigma^*(X)$,

$$
\Gamma(\Sigma(X), Y) = (1 + X) \times Y \quad \rightarrow \quad S(\Gamma(X, SY) + SY + X)
$$

$$(0, y) \rightarrow 0$$

$$(s(x), y) \rightarrow s(\text{in}_1(x, y)).$$

\footnote{Je paie un coup au premier qui a la ref – Daniel est hors concours, bien sûr.}
Interlude: un peu de voyance

Now I see... someone has a question, right?
Need for a relative notion

How about multiplication?

\[ s(x) \times y = (x \times y) + y \]

Uses addition.
Slightly less academic example: differential $\lambda$-calculus

Simple terms $\exists e, f ::= x | e \ M | De \cdot f | \lambda x.e$
Multiterms $\exists M, N ::= 0 | e + M$

- Extending simple operations to multiterms:
  $$(e + M) \ N = (e \ N) + M \ N \quad \lambda x.(e + M) = \lambda x.e + \lambda x.M \quad \ldots$$
  and defining multiterm sum
  $$0 + M = M \quad (e + M) + N = e + (M + N).$$

- Partial differentiation:
  \[
  \frac{\partial (e \ M)}{\partial x} \cdot N = \left( \frac{\partial e}{\partial x} \cdot N \right) \ M + \left( De \cdot \left( \frac{\partial M}{\partial x} \cdot N \right) \right) \ M \\
  \frac{\partial \lambda y.e}{\partial x} \cdot M = \lambda y. \left( \frac{\partial e}{\partial x} \cdot M \right) \quad \ldots
  \]
  (uses multioperations).
Distributive law increments

Definition

Given a distributive law $\delta : TS \rightarrow ST$, an increment of $\delta$ to some monad $T'$ is a natural transformation

$$T'S \rightarrow (T \oplus T')S,$$

satisfying coherence conditions, where $\oplus$ denotes coproduct in the category of monads.
Interlude: monad coproducts

- **Warning:** not pointwise, i.e., not created by the forgetful functor $U^C : \text{Mnd}_f(C) \to [C, C]_f$.

- **Example:** free monads.
  - The “free monad” functor is left adjoint to $U^C$, so
    \[ F^* \oplus G^* \cong (F + G)^* \neq F^* + G^* \]
    in general.
  - **Intuitively:**
    - $(F + G)^*$ interleaves operations from $F$ and $G$, while
    - $F^* + G^*$ is the disjoint union of $F$-terms and $G$-terms.
Back to increments

**Theorem**

Any increment $\gamma : T'S \to S(T \oplus T')$ over $\delta : TS \to ST$ extends to a monad distributive law

$$(T \oplus T')S \to S(T \oplus T').$$

If furthermore $T'(0) \cong 0$, then of course

$$S \to S(T \oplus T')$$

is admissible.
Relative structural extensions

A structural extension relative to some given monad distributive law $\delta : TS \to ST$ on a locally finitely presentable category $\mathcal{C}$, with $S = \Sigma^*$, consists of

- a finitary functor $\Gamma : \mathcal{C}^2 \to \mathcal{C}$,
- cocontinuous in its first argument, equipped with
- a natural transformation

$$\Gamma_Y(\Sigma(X)) \to ST(\Gamma_{STY}(X) + STY + X),$$

where again $\Gamma_Y(X) := \Gamma(X, Y)$. 
Theorem

Any relative structural extension

\[ d_{X,Y} : \Gamma_Y(\Sigma(X)) \to ST(\Gamma_{ST(Y)}(X) + ST(Y) + X) \]

over \( \delta : TS \to ST \), with \( S := \Sigma^* \), induces a distributive law increment

\[ T'S \to S(T \oplus T') , \]

where

\[ T'(X) = (\Gamma_{ST\Delta})^*(X) \]

\[ (\Gamma_{ST\Delta})(X) = \Gamma_{ST}(X)(X) , \]

hence

\[ T'(X) = \mu A.(X + \Gamma_{ST\Delta}(A)). \]

Furthermore, \( T'(0) \cong 0 \) by cocontinuity.
Corollary

Any relative structural extension

\[ d_{X,Y} : \Gamma_Y(\Sigma(X)) \to ST(\Gamma_{ST(Y)}(X) + ST(Y) + X) \]

over \( \delta : TS \to ST \), with \( S := \Sigma^* \), induces a distributive law

\[ (T \oplus T')S \to S(T \oplus T') \]

with \( (T \oplus T')(0) \cong 0 \), hence an admissible morphism

\[ S \to S(T \oplus T') . \]
Application: multiplication

Take

\[ \Sigma(X) = 1 + X \quad SX = \mu A.(1 + X + A) \quad \Gamma_Y(X) = X \times Y \]

\[ TX = (\Gamma_S \Delta)^*(X) = \mu A.(X + X \times SA) \quad \delta: TS \to ST \]

as for addition, take

\[ \Theta_Y(X) = X \times Y, \]

and define

\[ \Theta_Y(\Sigma X) \to ST(\Theta_{STY}(X) + STY + X) \]

(0, y) ↦ 0

(s(x), y) ↦ s(in_1(x, y) + in_2(y))

\(^2\)Not doing differential \( \lambda \) yet...
C’est la honte

The current framework does not even handle FPT!

<table>
<thead>
<tr>
<th>Missing: equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e[\sigma][\sigma'] = e[\sigma[\sigma']]$</td>
</tr>
</tbody>
</table>

Need to incorporate equations, potentially involving auxiliary operations, that are provable by induction on the initial model.
In differential $\lambda$-calculus, terms are quotiented by

$$D(De \cdot f) \cdot g = D(De \cdot g) \cdot f \quad e + (f + M) = f + (e + M).$$

Need to incorporate equations on basic operations, with which auxiliary operations are compatible.
Work in progress.
Thanks a lot for your attention (and bravery)!