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 $E[e]$.
  - Partial differentiation in differential λ-calculus.
  - ... and more!

Warning: work in progress!!!

A simple language with evaluation contexts

Syntax:

$$\begin{array}{rcl} e, f & ::= & x \mid e\ f \mid \lambda x.e \\ E & ::= & \square \mid E\ e \end{array}$$

Context application:

$$\begin{array}{rcl} \square[e] & = & e \\ (E\ e)[f] & = & E[f]\ e. \end{array}$$

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.

$$\begin{array}{rcl} e, f & ::= & x \mid e\ f \mid \lambda x.e \mid E[e] \\ E & ::= & \square \mid E\ e \end{array}$$

modulo

$$\square[e] = e \qquad\qquad (E\ e)[f] = E[f]\ e.$$

Problem

We get an explicit construction with a quotient, instead of the expected, inductive one.

A well-understood case

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 Σ-monoids: category $\Sigma\text{-}\mathbf{Mon}$.
- Categorically: the forgetful functor

$$\begin{array}{ccc} (I+\Sigma)\text{-}\mathbf{alg} & \longleftarrow & \Sigma\text{-}\mathbf{Mon} \\ & \searrow \qquad \swarrow & \\ & [\mathbb{F}, \mathbf{Set}] & \end{array}$$

  creates the initial object.
- Equivalently, for the induced monads, say $(I + \Sigma)^*$ and $\Sigma^{\circledast}$, and monad morphism

$$\alpha \colon (I + \Sigma)^* \to \Sigma^{\circledast},$$

  $\alpha_0 \colon (I + \Sigma)^*(0) \to \Sigma^{\circledast}(0)$ is an isomorphism.

Admissible monad morphisms

Definition

A monad morphism $\alpha \colon R \to S$ is admissible when α_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 $\delta\colon TR \to RT$, compatible with the units and multiplications of R and T.

Proposition (Beck)

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

- unit $\mathbf{C} \underset{T}{\overset{\Downarrow \eta^T}{\longrightarrow}} \mathbf{C} \underset{R}{\overset{\Downarrow \eta^R}{\longrightarrow}} \mathbf{C}$
- multiplication $RTRT \xrightarrow{R\delta T} RRTT \xrightarrow{\mu^R \mu^T} RT$.

Probably Damien (with Henning and Jurriaan), and then Matteo and Ralph have already told you a lot about distributive laws, so I won't insist.

Admissible extensions from distributive laws

Proposition

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

More precisely, if $0 \xrightarrow{\eta^T_0} T(0)$ is an isomorphism, then the monad morphism

$$R \xrightarrow{R\eta^T} RT$$

is admissible.

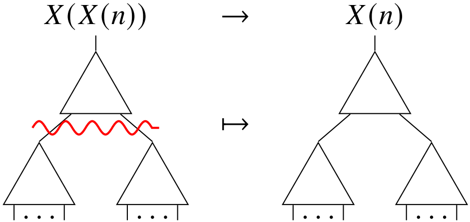
Proof.

Obvious. □

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

Recalling FPT

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



Recalling FPT

- Syntax specified by endofunctor $\Sigma\colon [\mathbf{Set},\mathbf{Set}]_f \to [\mathbf{Set},\mathbf{Set}]_f$, e.g.,

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

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

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

(for pointed Y).

Recalling FPT

Augmented models = Σ-monoids = $(I + \Sigma)$-algebras X with $\mathbf{m}\colon X \otimes X \longrightarrow X$ such that

$$
\begin{array}{ccccc}
\Sigma(X) \otimes X & & \xrightarrow{\;st_{X,X}\;} & & \Sigma(X \otimes X) \\
\Big\downarrow{\scriptstyle \mathbf{a} \otimes X} & & & & \Big\downarrow{\scriptstyle \Sigma(\mathbf{m})} \\
X \otimes X & & & & \Sigma(X) \\
& \searrow^{\mathbf{m}} & & \swarrow^{\mathbf{a}} & \\
& & X & &
\end{array}
$$

(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).

Moral

$$\Gamma_Y(\Theta(X)) \to \Theta^*(\Gamma_{\Theta^*(Y)}(X) + \Theta^*(Y) + X)$$

- 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.
- $\leadsto$ may use pointedness (and more) of $\Theta^*(Y)$.

Structural extensions

Definition

A structural extension of an endofunctor $\Sigma\colon \mathbf{C} \to \mathbf{C}$ on any locally finitely presentable category $\mathbf{C}$ is

- a finitary functor $\Gamma\colon \mathbf{C}^2 \to \mathbf{C}$,
- 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
$$\begin{aligned} S &= \Sigma^* \\ \Gamma_Y(X) &= \Gamma(X,Y). \end{aligned}$$

Theorem

Any structural extension

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

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

$$TS \longrightarrow ST,$$

where

$$\begin{array}{rcl} T(X) & = & (\Gamma_S \Delta)^*(X) \\ (\Gamma_S \Delta)(X) & = & \Gamma_{S(X)}(X), \end{array}$$

hence

$$T(X) = \mu A.(X + \Gamma_{SA}(A)).$$

Furthermore, $T(0) \cong 0$ by cocontinuity of Γ, hence $S \longrightarrow ST$ is admissible.

Augmented algebras

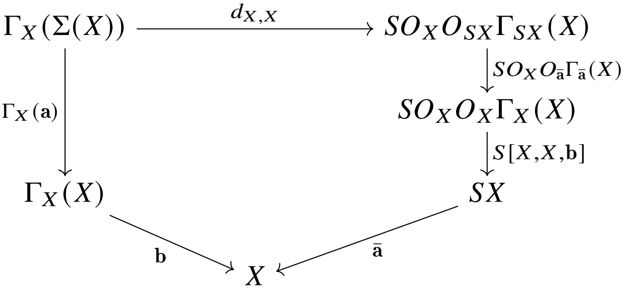
As a bonus, we may characterise ST-algebras.

Augmented algebras

An algebra for a structural extension $E = (\Gamma, d)$ of $\Sigma\colon \mathbf{C} \to \mathbf{C}$ consists of an object $X \in \mathbf{C}$, equipped with

- Σ-algebra structure $\mathbf{a}\colon \Sigma(X) \to X$ and
- $\Gamma\Delta$-algebra structure $\mathbf{b}\colon \Gamma_X(X) \to X$,

making the following diagram commute,



where $O_X(Z) := X + Z$ and $\bar{\mathbf{a}}\colon S(X) \to X$ is freely induced by $\mathbf{a}$.

Category of augmented algebras

- A morphism $X \longrightarrow Y$ of algebras for $E = (\Gamma, d)$ is a morphism between underlying objects which is both a morphism of Σ- and $\Gamma\Delta$-algebras.
- Let E **-alg** denote the category of algebras for E, or E-algebras.

Characterisation of algebras

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

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

over $\mathbf{C}$, where capital $\mathbf{Alg}$ denotes monad algebras.

Application 1: evaluation contexts

Recall

$$\begin{array}{rcl} e, f & ::= & x \mid e\ f \mid \lambda x.e \\ E & ::= & \square \mid E\ e \end{array}$$

with

$$\begin{array}{rcl} \square[e] & = & e \\ (E\ e)[f] & = & E[f]\ e. \end{array}$$

A structural extension for context application

- Ambient category $[\mathbf{Set}, \mathbf{Set}^2]_f$:
  - $X(n)_{\mathbf{p}}$ = set of programs with n free variables,
  - $X(n)_{\mathbf{c}}$ = set of contexts with n free variables.
- Syntax:

$$\begin{array}{rcl} \Sigma(X)(n)_{\mathbf{p}} & = & n + X(n)_{\mathbf{p}}^2 + X(n+1)_{\mathbf{p}} \\ (e, f & ::= & x \mid \ \ e\ f \ \ \mid \lambda x.e) \\ \Sigma(X)(n)_{\mathbf{c}} & = & 1 + X(n)_{\mathbf{c}} \times X(n)_{\mathbf{p}} \\ (E & ::= & \square \mid E\ \ e) \end{array}$$

- Structural extension: take $\Gamma(X, Y)(n)_{\mathbf{p}} = X(n)_{\mathbf{c}} \times Y(n)_{\mathbf{p}}$ (empty at $\mathbf{c}$) with, at $\mathbf{p}$:

$$\begin{array}{rcl} \Sigma(X)(n)_{\mathbf{c}} \times Y(n)_{\mathbf{p}} & \longrightarrow & S(\Gamma(X,Y) + S(Y) + X)_{\mathbf{p}} \\ (\square, e) & \mapsto & \eta^S(in_2(\eta^S(e))) \\ (E\ \ f, e) & \mapsto & \eta^S(in_1(E, e))\ \eta^S(in_3(f)). \end{array}$$

Application 2: Etonnante modernité de l'addition[1]

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

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

where

$$0 := in_1(\star) \qquad\qquad s(x) := in_2(x).$$

From a structural extension

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

$$\begin{array}{rcl} \Gamma(\Sigma(X), Y) = (1 + X) \times Y & \to & S(\Gamma(X, SY) + SY + X) \\ (0, y) & \mapsto & 0 \\ (s(x), y) & \mapsto & s(in_1(x, y)). \end{array}$$

[1] 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 λ-calculus

$$\begin{array}{rcl} \text{Simple terms} \ni e, f & ::= & x \mid e\ M \mid \mathbf{D}e \cdot f \mid \lambda x.e \\ \text{Multiterms} \ni M, N & ::= & 0 \mid e + M \end{array}$$

- Extending simple operations to multiterms:

$$(e + M)\ N = (e\ N) + M\ N \qquad \lambda x.(e + M) = \lambda x.e + \lambda x.M \qquad \ldots$$

  and defining multiterm sum

$$0 + M = M \qquad (e + M) + N = e + (M + N).$$

- Partial differentiation:

$$\frac{\partial (e\ M)}{\partial x} \cdot N = \left(\frac{\partial e}{\partial x} \cdot N\right)\ M + \left(\mathbf{D}e \cdot \left(\frac{\partial M}{\partial x} \cdot N\right)\right)\ M$$

$$\frac{\partial \lambda y.e}{\partial x} \cdot M = \lambda y.\left(\frac{\partial e}{\partial x} \cdot M\right) \qquad \ldots$$

  (uses multioperations).

Distributive law increments

Definition

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

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

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 $\mathbf{U^C} \colon \mathbf{Mnd}_f(\mathbf{C}) \to [\mathbf{C},\mathbf{C}]_f$.
- Example: free monads.
  - The "free monad" functor is left adjoint to $\mathbf{U^C}$, 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 $\gamma\colon T'S \to S(T \oplus T')$ over $\delta\colon TS \to ST$ 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 $\delta\colon TS \to ST$ on a locally finitely presentable category $\mathbf{C}$, with $S = \Sigma^*$, consists of

- a finitary functor $\Gamma\colon \mathbf{C}^2 \to \mathbf{C}$,
- 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
$$\begin{aligned} T'(X) &= (\Gamma_{ST}\Delta)^*(X) \\ (\Gamma_{ST}\Delta)(X) &= \Gamma_{ST(X)}(X), \end{aligned}$$

hence

$$T'(X) = \mu A.(X + \Gamma_{ST\,A}(A)).$$

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[2]

Take

$$\Sigma(X) = 1 + X \qquad SX = \mu A.(1 + X + A) \qquad \Gamma_Y(X) = X \times Y$$

$$TX = (\Gamma_S \Delta)^*(X) = \mu A.(X + X \times SA) \qquad \delta : TS \longrightarrow ST$$

as for addition, take

$$\Theta_Y(X) = X \times Y,$$

and define

$$\begin{array}{rcl} \Theta_Y(\Sigma X) & \longrightarrow & ST(\Theta_{STY}(X) + STY + X) \\ (0, y) & \mapsto & 0 \\ (s(x), y) & \mapsto & s(in_1(x, y) + in_2(y)) \end{array}$$

[2]Not doing differential λ yet...

C'est la honte

The current framework does not even handle FPT!

Missing: equations

$$e[\sigma][\sigma'] = e[\sigma[\sigma']] \qquad\qquad e[id] = e.$$

Need to incorporate equations, potentially involving auxiliary operations, that are provable by induction on the initial model.

Quotients 2

In differential λ-calculus, terms are quotiented by

$$\mathbf{D}(\mathbf{D}e \cdot f) \cdot g = \mathbf{D}(\mathbf{D}e \cdot g) \cdot f \qquad e + (f + M) = f + (e + M).$$

Need to incorporate equations on basic operations, with which auxiliary operations are compatible.

Work in progress.

Thanks a lot for your attention (and bravery)!