High-level signatures and initial semantics

Ambroise Lafont

joint work with Benedikt Ahrens, André Hirschowitz, Marco Maggesi

CSL 2018

Overview

Topic: specification and construction of untyped syntaxes with variables and a well-behaved substitution (e.g. lambda calculus).

Our work:

1. general notion of **signature** based on **monads** and **modules**.

   - *Caveat:* Not all of them do **generate a syntax**
   - special case: classical **binding signatures**

2. our main result: any **quotient** of algebraic signatures generates a syntax.

This talk: explain the words in bold
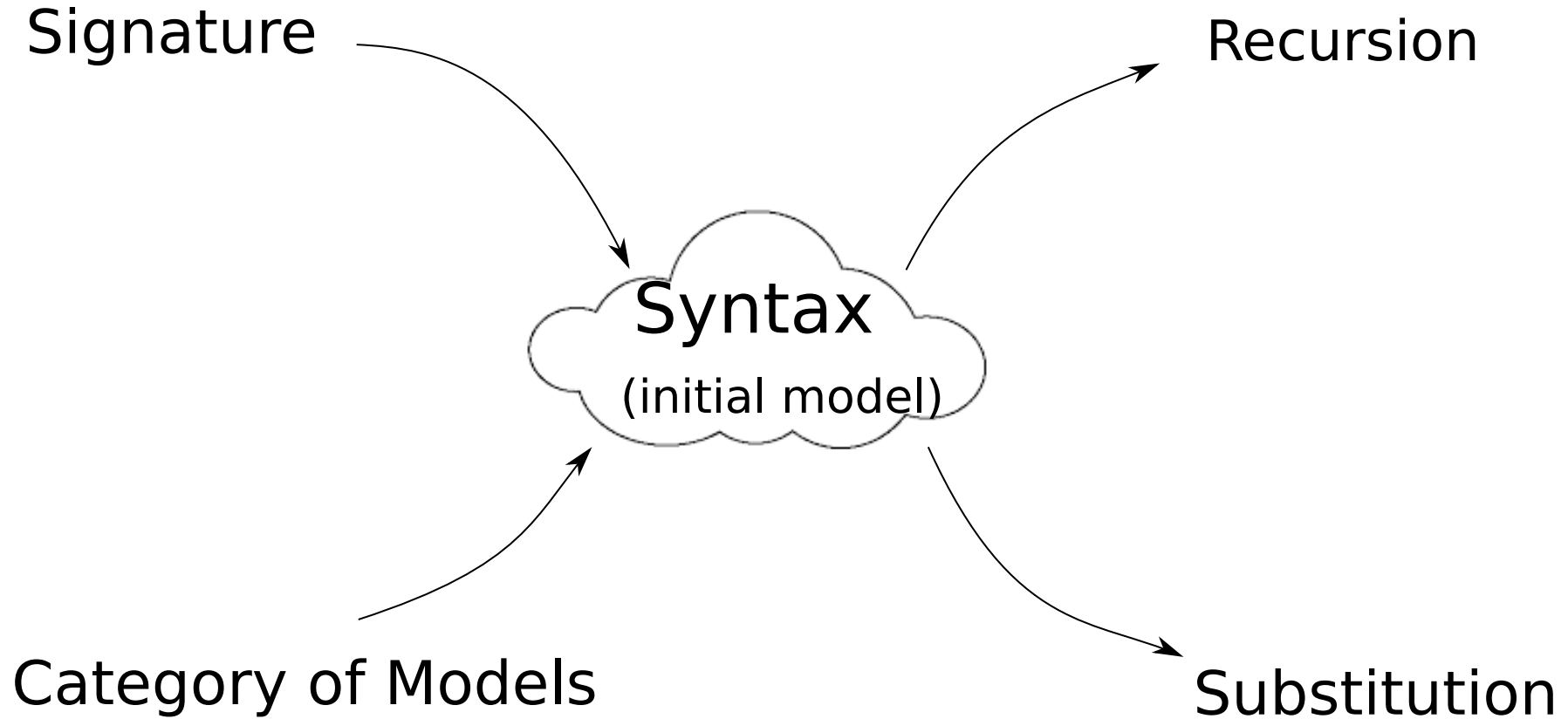
Operations covered by our result

Some examples:

- Symmetric operations

$$m : T \times T \to T \qquad \text{s.t.} \qquad m(t, u) = m(u, t)$$

- Explicit substitution

- Coherent fixed point operation

- Syntactic closure operator

What is a syntax?

Signature

Recursion

Syntax
(initial model)

Category of Models

Substitution

generates a syntax = existence of the initial model

Table of contents

1. Binding signatures and their models

- Categorical formulation of term languages

- Initial semantics for binding signatures

- Categorical formulation of substitution

2. Signatures and models based on monads and modules

3. Our main result

Categorical formulation of a term language

Example: syntax with a binary operation, a constant, and variables

$$\text{expr} ::= x \qquad \textit{(variable)}$$
$$| \ t_1 \star t_2 \qquad \textit{(binary operation)}$$
$$| \ 0 \qquad \textit{(constant)}$$

The syntax can be considered as the endofunctor B (on Set):

$$B : X \mapsto \{\text{expressions over } X\}$$

For example:

$$B(\emptyset) = \{0, 0 \star 0, \dots\}$$
$$B(\{x, y\}) = \{0, 0 \star 0, \dots, x, y, x \star y, \dots\}$$

Categorical formulation of a term language

Then we have:

$$\bigstar : B \times B \dot{\rightarrow} B$$

$$0 : \quad 1 \quad \dot{\rightarrow} B$$

$$\mathrm{var} : \quad \mathrm{Id}_{\mathrm{Set}} \dot{\rightarrow} B$$

Putting all together:

$$B \times B + 1 + \mathrm{Id}_{\mathrm{Set}} \dot{\rightarrow} B$$

i.e. B is an algebra for the endofunctor $F \mapsto F \times F + 1 + \mathrm{Id}_{\mathrm{Set}}$ on the category $\mathrm{End}_{\mathrm{Set}}$.

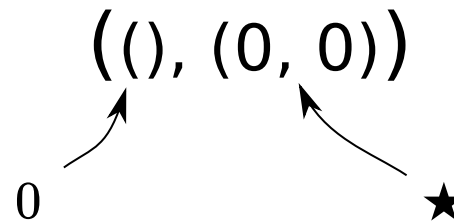Actually, B can be **defined** to be the initial algebra.

Binding Signatures

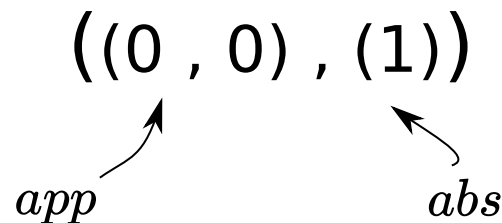**Binding signature** = a family of lists of natural numbers.

Each list specifies one operation in the syntax:

- length of the list = number of arguments of the operation

- natural number in the list = number of bound variables in the corresponding argument

Syntax with 0, ★:

$$\big((), (0, 0)\big)$$

0 ↗   ↖ ★

Lambda calculus:

$$\big((0, 0), (1)\big)$$

↗ *app*   ↖ *abs*

Initial semantics for binding signatures

Reminder

The syntax $(0,\star)$ is the initial algebra for the endofunctor:

$$F \mapsto F \times F + 1 + \mathrm{Id}_{\mathrm{Set}}$$

More generally, any binding signature gives rise to an endofunctor Σ.

Definition

Model = $(\Sigma + \mathrm{Id}_{\mathrm{Set}})$-algebra

Classical Theorem

The initial $(\Sigma + \mathrm{Id}_{\mathrm{Set}})$-algebra of a binding signature Σ always exists.

Does this initial algebra come with a well-behaved substitution?

Classical results on initial semantics

The endofunctor Σ induced by a binding signature comes with a *strength* which allows [FPT] to refine the notion of model:

Σ-monoid:

    $\Sigma+\mathrm{Id}_{\mathrm{Set}}$-algebra **equipped with a well-behaved substitution**.

Σ-monoid morphisms:

    algebra morphisms commuting with substitution.

Theorem [FPT]:

The initial $\Sigma + \mathrm{Id}_{\mathrm{Set}}$-algebra of a binding signature comes with a well-behaved substitution that makes it initial in the category of **Σ-monoids**.

This suggests defining signatures to be endofunctors on $\mathrm{End}_{\mathrm{Set}}$ *with strength* (as in [Matthes-Uustalu 2004]).

Table of contents

The Big Picture of signatures and models

Binding signatures $\hookrightarrow$ Endofunctors with strength $\hookrightarrow$ Our signatures

A **signature** Σ is a functorial assignment:
$$R \mapsto \Sigma(R)$$

A **model of** Σ is a pair:
$$(R, \quad \rho : \Sigma(R) \to R)$$

$$
\begin{aligned}
\text{monad} \ &:= \ \text{endofunctor with substitution} \\
\text{module over a monad} \ &:= \ \text{endofunctor with substitution} \\
\text{module morphism} \ &:= \ \text{natural transformation preserving substitution}
\end{aligned}
$$

The Big Picture of signatures and models

Binding signatures $\hookrightarrow$ Endofunctors with strength $\hookrightarrow$ Our signatures

A **signature** Σ is a functorial assignment:

$$R \mapsto \Sigma(R)$$

monad

A **model of** Σ is a pair:

$$(R, \quad \rho : \Sigma(R) \to R)$$

$$
\begin{array}{rcl}
\text{monad} & := & \text{endofunctor with substitution} \\
\text{module over a monad} & := & \text{endofunctor with substitution} \\
\text{module morphism} & := & \text{natural transformation preserving substitution}
\end{array}
$$

The Big Picture of signatures and models

Binding signatures $\hookrightarrow$ Endofunctors with strength $\hookrightarrow$ Our signatures

A **signature** Σ is a functorial assignment:

$$R \mapsto \Sigma(R)$$

monad

module over R

A **model of** Σ is a pair:

$$(R, \quad \rho : \Sigma(R) \to R)$$

$$
\begin{aligned}
\text{monad} \;&:= \; \text{endofunctor with substitution} \\
\text{module over a monad} \;&:= \; \text{endofunctor with substitution} \\
\text{module morphism} \;&:= \; \text{natural transformation preserving substitution}
\end{aligned}
$$

The Big Picture of signatures and models

Binding signatures $\hookrightarrow$ Endofunctors with strength $\hookrightarrow$ Our signatures

A **signature** Σ is a functorial assignment:

$$R \mapsto \Sigma(R)$$

monad

module over R

A **model of** Σ is a pair:

$$(R, \quad \rho : \Sigma(R) \to R)$$

monad

$$
\begin{array}{rcl}
\text{monad} & := & \text{endofunctor with substitution} \\
\text{module over a monad} & := & \text{endofunctor with substitution} \\
\text{module morphism} & := & \text{natural transformation preserving substitution}
\end{array}
$$

The Big Picture of signatures and models

Binding signatures $\hookrightarrow$ Endofunctors with strength $\hookrightarrow$ Our signatures

A **signature** Σ is a functorial assignment:

$$R \mapsto \Sigma(R)$$

monad

module over R

A **model of** Σ is a pair:

$$(R, \quad \rho : \Sigma(R) \to R)$$

monad

module morphism

$$
\begin{aligned}
\text{monad} \ &:= \ \text{endofunctor with substitution} \\
\text{module over a monad} \ &:= \ \text{endofunctor with substitution} \\
\text{module morphism} \ &:= \ \text{natural transformation preserving substitution}
\end{aligned}
$$

Substitution and monads

Reminder:

- $B(X)$ = expressions built out of 0, ★ and variables taken in X
- Variables induce a natural transformation $\eta : \mathrm{Id}_{\mathrm{Set}} \to B$

substitution $\mathrm{bind} : B(X) \to (X \to B(Y)) \to B(Y)$ subject to satisfy some equations.

A triple (B, η, bind) is called a **monad**.

A **monad morphism** between two monads R and S is a family of maps $(f_X : R(X) \to S(X))_X$ preserving variables and substitution.

Preview: Operations are module morphisms

★ **commutes with substitution**

$$(t \star u)[x \mapsto v_x] = t[x \mapsto v_x] \star u[x \mapsto v_x]$$

In the right hand side, substitution acts on a pair of expressions.

Categorical formulation

$B \times B$ supports B-substitution $\rightsquigarrow$ $B \times B$ is a **module over** B

★ commutes with substitution $\rightsquigarrow$ $\star : B \times B \to B$ is a **module morphism**

Building blocks for binding signatures

Essential constructions of **modules over a monad** R:

- R itself

- M x N for any modules M and N (in particular, R x R)

- The **derivative of a module** M is the module M' defined by
  $M'(X) = M(X + \{\bullet\})$.

  The derivative is used to model an operation binding a variable
  (Cf next slide).

Syntactic operations are module morphisms

A **module morphism** between two modules M and N on the same monad R is a family of maps $(f_X\!:\!M(X) \to N(X))_X$ commuting with substitution.

$id_M : M \to M$

the family of identity maps $(id_{M(X)}\!:\!M(X) \to M(X))_X$ for any module M

$\star : B \times B \to B$

$app : L \times L \to L$

the application operation of the lambda calculus monad L.

$abs : L'\ \to L$

Indeed, in $\lambda x.t$, the term t depends on an additional free variable x:
If $t \in L(Y + \{x\})$ **$= L'(Y)$**, then $abs(t) = \lambda x.t \in L(Y)$

The Big Picture again

A **signature** Σ is a functorial assignment:

monad module over R

$$R \mapsto \Sigma(R)$$

A **model of** Σ is a pair: $(R, \quad \rho : \Sigma(R) \to R)$

monad module morphism

A **model morphism** $m : (R,\rho) \to (S,\sigma)$ is a monad morphism commuting with the module morphism:

$$
\begin{array}{ccc}
\Sigma(R) & \xrightarrow{\rho} & R \\
\Sigma(m) \downarrow & & \downarrow m \\
\Sigma(S) & \xrightarrow{\sigma} & S
\end{array}
$$

Syntax

Given a signature Σ, its **syntax** is an initial object in its category of models.

Question: Does the syntax exist for every signature?

Answer: No.

Counter-example: the signature $R \mapsto \mathscr{P} \circ R$

powerset endofunctor on Set

Examples of signatures generating syntax

- $R \mapsto 1 + R \times R$

  Model: (R equipped with module morphisms $1 \to R$

  and $R \times R \to R$.

  The syntax is our previous $(0, \bigstar)$ language.

- $R \mapsto R \times R + R'$

  Models are monads R equipped with two modules morphisms:

  $R \times R \to R$ and $R' \to R$.

  The syntax is lambda calculus.

Algebraic signatures

More generally, the syntax exists for any signature induced by a disjoint sum of products of finite derivatives of the monad ($R \mapsto R' \times R'' \times R''' + R \times R'' \times R''' \times R + \dots$).

We call such a signature an **algebraic signature**. They correspond to binding signatures through the inclusion:

$$\text{Binding signatures} \hookrightarrow \text{Endofunctors with strength} \hookrightarrow \text{Our signatures}$$

Our main result: Quotients of algebraic signatures generate a syntax.

Table of contents

Quotient of a signature

Quotient of a set:

A quotient of a set X is a set Y together with a surjection $p : X \to Y$.

$$x \sim x' \qquad \Longleftrightarrow \qquad p(x) = p(x')$$

Quotient of a signature:

A quotient of a signature Σ consists of:

- a signature Ψ

- a (natural) family of surjective module

  morphisms $(f_R : \Sigma(R) \to \Psi(R))_R$

$$R \mapsto \begin{array}{c} \Sigma(R) \\ \downarrow f_R \\ \Psi(R) \end{array}$$

Syntax for presentable signatures

Definition
A **presentable signature** is a quotient of an algebraic signature.

Theorem
Any presentable signature generates a syntax.

Question: Are there interesting examples of presentable signatures?

Answer:
- Symmetric operations
- Explicit substitution
- Coherent fixed point operation
- ...

Example 1: Symmetric operations

Binary commutative operation $+$:

$$t + u = u + t$$

As a quotient of an algebraic signature:

$$R \times R$$

$$R \mapsto$$

$$\Downarrow$$

$$R \times R \,/\, \{(x,y) \sim (y,x)\}$$

This generalizes to **n-ary permutation invariant operations**.

Example 2: Explicit substitution

An operation $\_\langle x_i \mapsto t_i \rangle$ satisfying coherence equations:

- invariance under **permutation**

$$F(x, y)\langle x \mapsto t, y \mapsto u \rangle = F(y, x)\langle x \mapsto u, y \mapsto t \rangle$$

- invariance under **weakening**

$$F(x)\langle x \mapsto t, y \mapsto u \rangle = F(x)\langle x \mapsto u \rangle$$

- invariance under **contraction**

$$F(x, y)\langle x, y \mapsto t \rangle = F(x, x)\langle x \mapsto t \rangle$$

Example 2: Explicit substitution

Signature of explicit substitution as a quotient of the algebraic signature Σ:

$$\Sigma(R) := R' \times R \;+\; R'' \times R \times R \;+\; R''' \times R \times R \times R \;+\; ...$$

$$t\langle x \mapsto u\rangle \qquad\qquad t\langle x \mapsto u, y \mapsto v\rangle \qquad\qquad t\langle x \mapsto u, y \mapsto v, z \mapsto w\rangle$$

$$\Sigma(R)$$

$$R \mapsto$$

$$\Sigma(R)\,/\sim$$

- **permutation**: $\qquad t\langle x \mapsto u, y \mapsto v\rangle \;\sim\; t[x \rightleftarrows y]\langle x \mapsto v, y \mapsto u\rangle$

- **weakening**: $\qquad\qquad t\langle x \mapsto u\rangle \;\sim\; t\langle x \mapsto u, y \mapsto v\rangle$

- **contraction**: $\qquad t\langle x \mapsto u, y \mapsto u\rangle \;\sim\; t[y := x]\langle x \mapsto u\rangle$

Conclusion

Summary of the talk:

- presented a notion of signature and models

- identified a class of signatures that generate a syntax

  - encompasses the classical binding signatures

  - encompasses operations satisfying some equations

Future work:

- add equations (e.g. lambda calculus modulo beta/eta equivalence);

- extend our framework to simply typed syntaxes.

Conclusion

Summary of the talk:

- presented a notion of signature and models

- identified a class of signatures that generate a syntax

  - encompasses the classical binding signatures

  - encompasses operations satisfying some equations

Future work:

- add equations (e.g. lambda calculus modulo beta/eta equivalence);

- extend our framework to simply typed syntaxes.

Thank you!