Initiality for Typed Syntax and Semantics

Benedikt Ahrens

Université Nice Sophia–Antipolis

Sept 11, 2011

Motivating(?) Question: $PCF \to LC$

Mathematical structure of

$$PCF \to LC \qquad ?$$

Challenges:

- varying types
- capture compatibility with substitution + reduction

$PCF \to LC$ morphism in some category ?

Motivating(?) Question: $PCF \to LC$

Mathematical structure of

$$PCF \to LC \qquad ?$$

Challenges:

- varying types
- capture compatibility with substitution + reduction

$PCF \to LC$ initial morphism in some category ?

What is Initial Semantics?

Ingredients:

- Signature S
- $\rightsquigarrow$ Category of Semantics of S
- $\rightsquigarrow$ Initial Semantics $\Sigma(S)$ — the Syntax of S

Why?

- Definition of inductive type
- Categorical recursion operator (Ex. `fold`)

Features:

- Variable Binding
- Typing
- Reductions

Outline

Encodings of variable binding

- Nominal Approach – Named Abstraction

$$lam : A \times T \to T$$

- Higher-Order Abstract Syntax (HOAS)

$$lam : (T \to T) \to T$$

- Nested Datatype

$$lam : T(X + 1) \to T(X)$$

Example: Lambda Calculus

Signature: $\Lambda = \{abs : [1] \;,\quad app : [0,0]\}$

Category: ???

Syntax: Inductive LC (V : Set) : Set :=
| Var : V $\rightarrow$ LC (V)
| Abs : LC (V + 1) $\rightarrow$ LC (V)
| App : LC (V) $\times$ LC (V) $\rightarrow$ LC (V)

- $LC : Set \rightarrow Set$
- Constructors are Natural Transformations

Idea: Integrate Substitution...

Substitution $\equiv$ Monad

Monad $=$ Functor $+$ "variables–as–terms" $+$ Substitution:

- $P : \mathcal{C} \to \mathcal{C}$
- $\eta_X : \mathcal{C}(X, PX)$
- $\sigma_{X,Y} : \mathcal{C}(X, PY) \to \mathcal{C}(PX, PY) \quad + \quad$ substitution properties

Example ($LC : Set \to Set$, Altenkirch & Reus)

- $V \mapsto LC(V)$
- $v \mapsto Var_V(v) \in LC(V)$
- $(\gg=) : LC(V) \times (V \to LC(W)) \to LC(W)$

Substitution distributes over Constructors

- expressed by notion of

  Module over a Monad $\quad + \quad$ Morphism of Modules

- Domain and Codomain of Constructor = Module
- Constructor = Morphism of Modules

Examples

- $LC : V \mapsto LC(V)$
  $LC^* : V \mapsto LC(V + 1)$
  $LC \times LC$ — Modules over LC
- $Abs : LC^* \to LC$
  $App : LC \times LC \to LC$ — Morphisms of Modules over LC

Representations of a Signature

Definition (Representation of Signature S)

- Monad $P : Set \to Set$
- Morphism of Modules over P for each Arity $s \in S$

Example (Representation of Λ)

- Monad $P : Set \to Set$
- $app : P \times P \to P \quad , \qquad abs : P^* \to P$

Initial Semantics, untyped

Theorem (Hirschowitz & Maggesi)

For any signature S, the category of representations of S has an initial object.

Example

(LC, App, Abs) is the initial representation of Λ

Outline

Example: Simply–Typed LC

Types: $T \quad ::= \quad * \quad | \quad T \Rightarrow T$

Signature: $T\Lambda = \left\{ (abs_{s,t} : (s)t \to s \Rightarrow t)_{s,t \in T} \ , (app_{s,t} : \ldots)_{s,t} \right\}$

Syntax: Inductive TLC (V : T → Set) : T → Set :=
| Var : ∀ t, V(t) → TLC(V)(t)
| Abs : ∀ s t, TLC(V + s)(t) → TLC(V)(s⇒t)
| App : ∀ s t, TLC(V)(s⇒t) × TLC(V)(s) → . . .

- Monad $\quad TLC : Set^T \to Set^T$
- Fibre Module over monad *TLC*

$$V \mapsto TLC(V)(t) : Set^T \to Set$$

Initiality, typed

Definition (Representation of a Signature S over types T)

- Monad L on Set^T
- Morphism of Modules over L for each Arity $s \in S$

Theorem (Zsidó)

The Category of Representations of S has an initial object.

Implemented in Coq

B. A. and J. Zsidó, JFR, 2011

Initiality, typed

Definition (Representation of a Signature S over types T)

- Monad L on Set^T
- Morphism of Modules over L for each Arity $s \in S$

Problem: Set T of types hard–coded

Initiality, typed

Definition (Representation of a Signature S over types T)

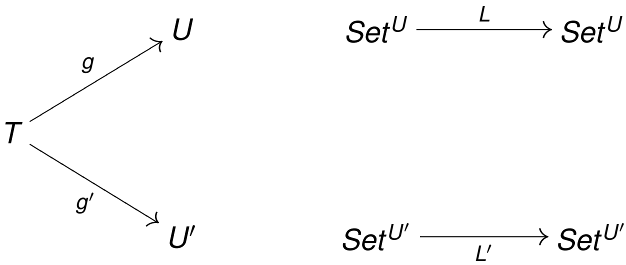
- Monad L on Set^T
- Morphism of Modules over L for each Arity $s \in S$

Problem: Set T of types hard–coded

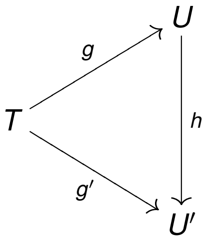
Definition II (Representation)

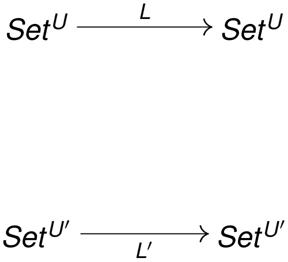
- Set U
- $g : T \to U$ morphism of types
- Monad L on Set^U
- representation of "S transported along g" in L

Changing Types, Changing Categories



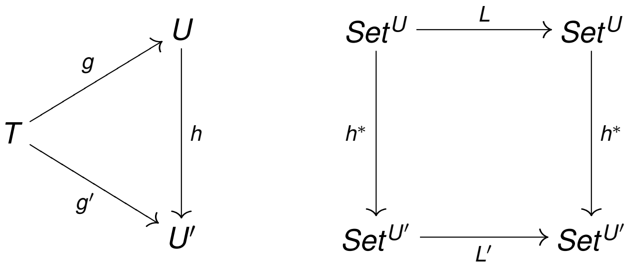
Changing Types, Changing Categories





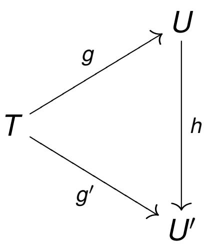
- $h : U \to U'$

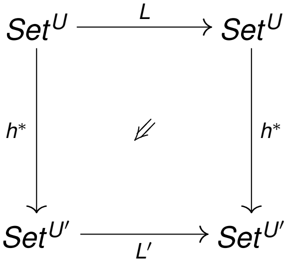
Changing Types, Changing Categories



- $h : U \to U'$
- induces $h^* : Set^U \to Set^{U'}$

Changing Types, Changing Categories





- $h : U \to U'$
- induces $h^* : Set^U \to Set^{U'}$
- Morphism of Representations:

$$h^* \circ L \Rightarrow L' \circ h^*$$

+ commutative diagrams

Initiality w. Type Change

Theorem (Generalized Initiality)

The category of representations of S has an initial object

Example (Representation of Sig of PCF in LC)

- map of types $T_{PCF} \to \{*\}$
-

Arity	**Rep in LC**
$app_{s,t}$	$App : LC \times LC \to LC$
$abs_{s,t}$	$Abs : LC^{*} \to LC$
$true : * \to Bool$	$True : * \to LC$
...	...

Yields a translation $PCF \to LC$

Outline

Integrating Semantics, untyped

$$\lambda x.M(N) \leadsto M[x := N]$$

- X Terms modulo Relations, Quotienting ?
- X Monads *Ord* $\to$ *Ord* ?
- ✓ *Relative* Monads *Set* $\to$ *Ord*

Relative Monad = Functor + Monad–like Data

- + $F : \mathcal{C} \to \mathcal{D}$
- ▸ $P : \mathcal{C} \to \mathcal{D}$
- ▸ $\eta_X : \mathcal{D}(FX, PX)$
- ▸ $\sigma : \mathcal{D}(FX, PY) \to \mathcal{D}(PX, PY)$ + substitution properties

Example : LCβ as Relative Monad on Δ

$$\Delta : Set \to Ord, \quad X \mapsto (X, diagonal)$$

LCβ as Relative Monad on Δ:

- $V \mapsto LC\beta(V) := (LC(V), \beta^*)$
- $Var_V : Ord(\Delta V, LC\beta(V))$
- $(\gg=) : Ord(\Delta V, LC\beta(W)) \to Ord\big(LC\beta(V), LC\beta(W)\big)$

Example : LCβ as Relative Monad on Δ

$$\Delta : Set \to Ord, \quad X \mapsto (X, diagonal)$$

LCβ as Relative Monad on Δ:

- $V \mapsto LC\beta(V) := (LC(V), \beta^*)$
- $Var_V : Ord(\Delta V, LC\beta(V))$
- $(\gg=) : Ord(\Delta V, LC\beta(W)) \to Ord\big(LC\beta(V), LC\beta(W)\big)$

Adjunction $\Delta \dashv U$ with forgetful $U : Ord \to Set$

Example : LCβ as Relative Monad on Δ

$$\Delta : Set \to Ord, \quad X \mapsto (X, diagonal)$$

LCβ as Relative Monad on Δ:

- $V \mapsto LC\beta(V) := (LC(V), \beta^*)$
- $Var_V : V \to LC(V)$
- $(\gg=) : \quad Set(V, LC(W)) \quad \to Ord\big(LC\beta(V), LC\beta(W)\big)$

Adjunction $\Delta \dashv U$ with forgetful $U : Ord \to Set$

2–Signatures

Definition (2–Signature)

- a (1–)signature S
- a set of S–inequations

Example ($\Lambda\beta$)

- signature Λ
- $app \circ (abs, id) \quad \leq \quad \_[* := \_]$

Representations of 2–Signatures

Definition (Representation of a 2–signature (S, A))

- Representation R of S in Relative Monad on Δ
- s.t. R verifies each inequation of A

Example (Representation of $\Lambda\beta$)

- Monad $\quad P : Set \stackrel{\Delta}{\to} Ord$
  $app : P \times P \to P \quad , \quad abs : P^* \to P$
- $app \circ (abs, id) \quad \leq \quad \_[* := \_]$

$$Rep(S, A) \quad \subseteq \quad Rep(S) \qquad \text{full subcategory}$$

Initiality for 2–Signatures

Theorem

The category of representations of (S, A) *has an initial object.*

- Proof in Coq
- Works with types, too

Example with Types: $PCF \to LC$

- in Coq
- compatible with reduction
- painful due to intrinsic typing

Thanks for your attention