

Polymorphic Algebraic Theories

Marcelo Fiore

COMPUTER LABORATORY
UNIVERSITY OF CAMBRIDGE

GlynnFest
1.VI.2013

Joint work with Makoto Hamana

Method and Technique

- ▶ Interplay between:
mathematical structure and formal language.
- ▶ Categorical modelling tools:
categorical algebra, presheaf categories,
Grothendieck construction, Kan extensions,
discrete generalised polynomial functors.

This Talk

One more step on *Algebraic Foundations for Type Theories*.

- Second-order algebraic theories. [CSL'10, MFCS'10]
(e.g. untyped and simply-typed λ -calculus)
- ▶ Polymorphic algebraic theories. [LICS'13]
(e.g. System F)
- Dependent algebraic theories.
(e.g. MLTT)

Polymorphic Algebraic Theories

- ▶ Theory:

Mathematical foundations.

- ▶ Application:

Mechanised formalisation.

Algebraic Desiderata

- ▶ Signatures = syntax specification.

Algebraic Desiderata

- ▶ Signatures = syntax specification.
- ▶ Pre-models = signature algebras.

Algebraic Desiderata

- ▶ Signatures = syntax specification.
- ▶ Pre-models = signature algebras.
- ▶ Free pre-models = abstract syntax.

Algebraic Desiderata

- ▶ Signatures = syntax specification.
- ▶ Pre-models = signature algebras.
- ▶ Free pre-models = abstract syntax.
- ▶ **Equational presentations = sets of equations.**

Algebraic Desiderata

- ▶ Signatures = syntax specification.
- ▶ Pre-models = signature algebras.
- ▶ Free pre-models = abstract syntax.
- ▶ Equational presentations = sets of equations.
- ▶ **Models = pre-models satisfying equational presentations.**

Algebraic Desiderata

- ▶ Signatures = syntax specification.
- ▶ Pre-models = signature algebras.
- ▶ Free pre-models = abstract syntax.
- ▶ Equational presentations = sets of equations.
- ▶ Models = pre-models satisfying equational presentations.
- ▶ Logical framework = sound and complete equational logic.

Algebraic Desiderata

- ▶ Signatures = syntax specification.
- ▶ Pre-models = signature algebras.
- ▶ Free pre-models = abstract syntax.
- ▶ Equational presentations = sets of equations.
- ▶ Models = pre-models satisfying equational presentations.
- ▶ Logical framework = sound and complete equational logic.
- ▶ **Translations = algebraic homomorphisms.**

Algebraic Desiderata

- ▶ Signatures = syntax specification.
- ▶ Pre-models = signature algebras.
- ▶ Free pre-models = abstract syntax.
- ▶ Equational presentations = sets of equations.
- ▶ Models = pre-models satisfying equational presentations.
- ▶ Logical framework = sound and complete equational logic.
- ▶ Translations = algebraic homomorphisms.
- ▶ **Equational theories = invariant presentations.**

Polymorphic Type Theories

Example: Polymorphic FPC

Types:

$$\tau ::= \alpha \mid \tau_1 + \tau_2 \mid \tau_1 \times \tau_2 \mid \tau_1 \Rightarrow \tau_2 \mid \mu(\alpha.\tau) \mid \forall(\alpha.\tau)$$

Terms:

$$t ::= \dots \mid \lambda(x:\tau.t) \mid t_1(t_2) \mid \Lambda(\alpha.t) \mid t(\tau)$$

Polymorphic Type Theories

Example: Polymorphic FPC

Types:

LEVEL 1

$\tau ::= \alpha \mid \tau_1 + \tau_2 \mid \tau_1 \times \tau_2 \mid \tau_1 \Rightarrow \tau_2 \mid \mu(\alpha.\tau) \mid \forall(\alpha.\tau)$

Terms:

LEVEL 2

$t ::= \dots \mid \lambda(x:\tau.t) \mid t_1(t_2) \mid \Lambda(\alpha.t) \mid t(\tau)$

Polymorphic Type Theories

Example: Polymorphic FPC with equi-recursive types

Types:

LEVEL 1

$\tau ::= \alpha \mid \tau_1 + \tau_2 \mid \tau_1 \times \tau_2 \mid \tau_1 \Rightarrow \tau_2 \mid \mu(\alpha.\tau) \mid \forall(\alpha.\tau)$

$\mu(\alpha.\tau) = \tau[\mu(\alpha.\tau)/\alpha]$

Terms:

LEVEL 2

$t ::= \dots \mid \lambda(x:\tau.t) \mid t_1(t_2) \mid \Lambda(\alpha.t) \mid t(\tau)$

Second-Order Algebraic Theories

- ▶ Binding signatures.

Example:

operator	arity	specification
λ	$: (*)* \rightarrow *$	one binding argument
@	$: *, * \rightarrow *$	two non-binding arguments

Second-Order Algebraic Theories

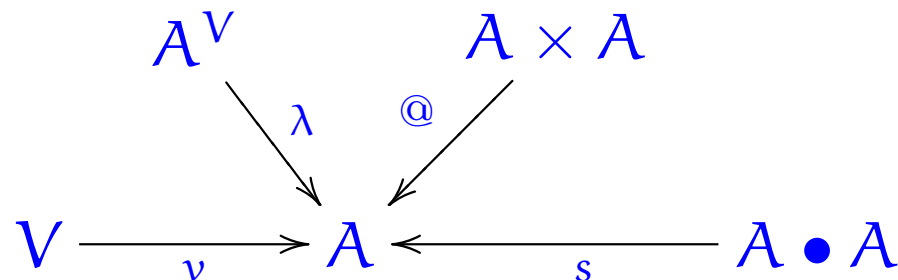
- ▶ Binding signatures.

Example:

operator	arity	specification
λ	$(*)^* \rightarrow *$	one binding argument
@	$*, * \rightarrow *$	two non-binding arguments

- ▶ Pre-models = Σ -monoids in $Set^{\mathbb{F}}$.

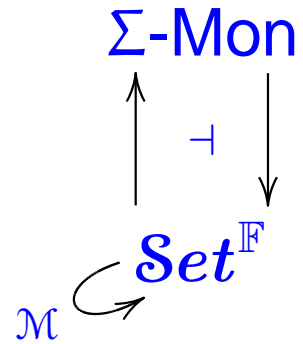
Example:



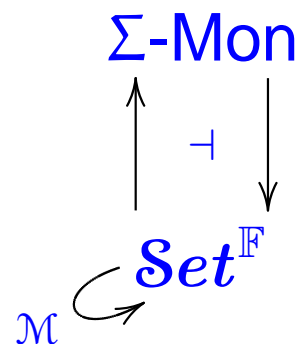
algebra
 \Updownarrow compatible structures
 Lawvere theory

► Free pre-models

= abstract syntax with variable binding and metavariables



- ▶ Free pre-models
 - = abstract syntax with variable binding and metavariables



Example: $t \in \mathcal{M}(\underline{X})(\Gamma)$ iff $X \triangleright \Gamma \vdash t$

$$\frac{X \triangleright \Gamma, x \vdash t}{X \triangleright \Gamma \vdash \lambda(x.t)}$$

$$\frac{X \triangleright \Gamma \vdash t_1 \quad \Gamma \vdash t_2}{X \triangleright \Gamma \vdash t_1 @ t_2}$$

$$\frac{}{X \triangleright \Gamma \vdash x} \quad (x \in \Gamma)$$

$$\frac{X \triangleright \Gamma \vdash t_i (1 \leq i \leq n)}{X \triangleright \Gamma \vdash M[t_1, \dots, t_n]} \quad (M \in X(n))$$

► Second-order equational presentations

Example:

$$(\beta) \quad M : [*]* , N : * \vdash \lambda(x.M[x]) @ N = M[N]$$

► Second-order equational presentations

Example:

$$(\beta) \quad M : [*]* , N : * \vdash \lambda(x.M[x]) @ N = M[N]$$

► Second-order models

Example: premodels A satisfying

$$\llbracket \lambda(x.M[x]) @ N[] \rrbracket = \llbracket M[N[]] \rrbracket : A^V \times A \rightarrow A$$

Coincides with Martin Hyland's notion of *semiclosed algebraic theory* as a model of the λ -calculus

► Second-Order Equational Logic

Judgements

$$X \triangleright \Gamma \vdash t_1 = t_2$$

subject to congruence rules of meta substitution and extension.

Polymorphic Algebraic Theories

Types:

LEVEL 1

Second-order equational presentations

Example:

$$+, \times, \Rightarrow : *, * \rightarrow *$$

$$\forall, \exists, \mu : (*)* \rightarrow *$$

$$\top : [*]* \vdash \mu(x.T[x]) = \top[\mu(x.T[x])]$$

Terms:

LEVEL 2

Example: System F

Vernacular rules.

$$\frac{\alpha_i \mid x_j : \tau_j, x : \sigma \vdash t : \tau}{\alpha_i \mid x_j : \tau_j \vdash \lambda(x : \sigma.t) : \sigma \Rightarrow \tau}$$

$$\frac{\alpha_i, \alpha \mid x_i : \tau_i \vdash t : \tau}{\alpha_i \mid x_i : \tau_i \vdash \Lambda(\alpha.t) : \forall(\alpha.\tau)}$$

Terms:

LEVEL 2

Example: System F

Vernacular rules.

$$\frac{\alpha_i \mid x_j : \tau_j, x : \sigma \vdash t : \tau}{\alpha_i \mid x_j : \tau_j \vdash \lambda(x : \sigma.t) : \sigma \Rightarrow \tau}$$

$$\frac{\alpha_i, \alpha \mid x_i : \tau_i \vdash t : \tau}{\alpha_i \mid x_i : \tau_i \vdash \Lambda(\alpha.t) : \forall(\alpha.\tau)}$$

Polymorphic signature operators.

$$\lambda \quad :: \quad S, T : * \quad \triangleright \quad (S)T \rightarrow S \Rightarrow T$$

$$\Lambda \quad :: \quad T : [*]* \quad \triangleright \quad \langle \alpha \rangle T[\alpha] \rightarrow \forall(\alpha.T[\alpha])$$

Example: Existential λ -calculus

Vernacular rules.

$$\frac{\alpha_i \mid x_j : \tau_j \vdash s : \sigma[\tau/\alpha]}{\alpha_i \mid x_j : \tau_j \vdash \text{pack}(\tau, s) : \exists(\alpha.\sigma)}$$

$$\frac{\alpha_i \mid x_j : \tau_j \vdash s : \exists(\alpha.\sigma) \quad \alpha_i, \alpha \mid x_j : \tau_j, x : \sigma \vdash t : \tau}{\alpha_i \mid x_j : \tau_j \vdash \text{unpack } s \text{ as } (\alpha, x) \text{ in } t : \tau} \quad (\alpha \# \tau_j, \tau)$$

Example: Existential λ -calculus

Vernacular rules.

$$\frac{\alpha_i \mid x_j : \tau_j \vdash s : \sigma[\tau/\alpha]}{\alpha_i \mid x_j : \tau_j \vdash \text{pack}(\tau, s) : \exists(\alpha.\sigma)}$$

$$\frac{\alpha_i \mid x_j : \tau_j \vdash s : \exists(\alpha.\sigma) \quad \alpha_i, \alpha \mid x_j : \tau_j, x : \sigma \vdash t : \tau}{\alpha_i \mid x_j : \tau_j \vdash \text{unpack } s \text{ as } (\alpha, x) \text{ in } t : \tau} \quad (\alpha \# \tau_j, \tau)$$

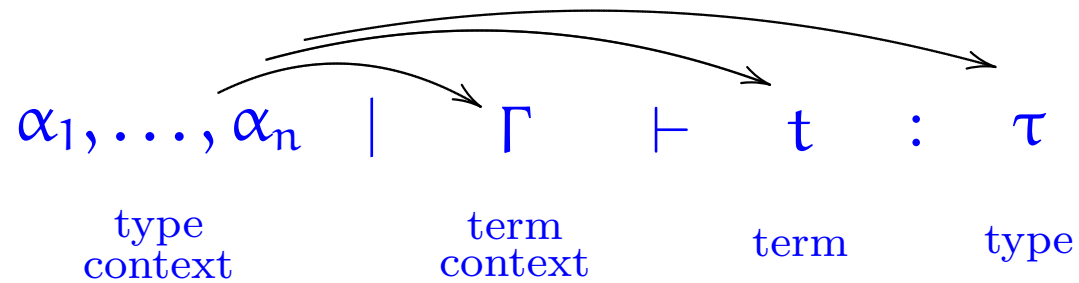
Polymorphic signature operators.

$$\text{pack} \quad :: \quad S : [*]*, T : * \quad \triangleright \quad S[T] \rightarrow \exists(\alpha.S[\alpha])$$

$$\text{unpack} \quad :: \quad S : [*]*, T : * \quad \triangleright \quad \exists(\alpha.S[\alpha]), \langle \alpha \rangle (S[\alpha])T \rightarrow T$$

Contexts for Polymorphism

Contexts



with types in a universe \mathbf{U} are modelled by the Grothendieck construction

[Hamana FOSSACS'11]

$$\mathbf{GU} = \int^{n \in \mathbf{F}} \mathbf{F} \downarrow (\mathbf{U}n) \times \mathbf{U}n$$

Polymorphic Signatures

- ▶ A *polymorphic signature* consists of
 1. a second-order signature Σ_1 and equational presentation E_1 for *type structure*, and
 2. a polymorphic signature Σ_2 for *term structure*.

Polymorphic Signatures and Structures

- ▶ A *polymorphic signature* consists of
 1. a second-order signature Σ_1 and equational presentation E_1 for *type structure*, and
 2. a polymorphic signature Σ_2 for *term structure*.
- ▶ A *polymorphic structure* consist of
 1. a type universe \mathcal{U} modelling (Σ_1, E_1) , and
 2. compatible algebraic structure in $\mathcal{S}et^{\mathcal{G}\mathcal{U}}$ as follows ...

$$\begin{array}{ccccc}
 & \Sigma_2 A & & \uparrow A & \\
 & \searrow & & \swarrow & \\
 V & \xrightarrow{v} & A & \xleftarrow{s} & A \bullet A
 \end{array}$$

where Σ_2 and \uparrow are endofunctors on $\mathcal{Set}^{\text{GU}}$ whose algebras respectively model the **polymorphic signature operators** and the operation of **type-in-term substitution**.

$$\begin{array}{ccccc}
 & & \Sigma_2 A & & \uparrow A \\
 & & \searrow & & \swarrow \\
 V & \xrightarrow{\quad v \quad} & A & \xleftarrow{\quad s \quad} & A \bullet A
 \end{array}$$

where Σ_2 and \uparrow are endofunctors on $\mathcal{Set}^{\text{GU}}$ whose algebras respectively model the **polymorphic signature operators** and the operation of **type-in-term substitution**.

NB: These constructions are performed within the theory of discrete generalised polynomial functors on presheaf categories^[ICALP'12] and hence support free constructions, which provide the **abstract syntax of polymorphic algebraic theories**.

Polymorphic Equational Presentations

Example: System F

Vernacular.

$$(\beta) \quad \Gamma \vdash \lambda(x : \sigma.M)@N = M[N/x] : \tau$$

$$(\beta') \quad \Gamma \vdash \Lambda(\alpha.M)(\sigma) = M[\sigma/\alpha] : \tau[\sigma/\alpha]$$

Formal.

$$(\beta) \quad S, T : * \triangleright M : [S]T, N : S$$

$$\vdash @_{S,T}(\lambda_{S,T}(x.M[x]), N) = M[N] : T$$

$$(\beta') \quad S : *, T : [*]* \triangleright M : \{\alpha\}T[\alpha]$$

$$\vdash @'_{S,T}(\Lambda_T(\alpha.M\{\alpha\})) = M\{S\} : T[S]$$

Example: Existential λ -calculus.

Vernacular.

$$(\exists\beta) \quad \Gamma \vdash \text{unpack } (\text{pack}(\iota, N)) \text{ as } (\alpha, x) \text{ in } M = M[\iota/\alpha, N/x] : \tau$$

$$(\exists\eta) \quad \Gamma \vdash \text{unpack } N \text{ as } (\alpha, x) \text{ in } M[\text{pack}(\alpha, x)/z] = M[N/z] : \tau$$

Formal.

$$(\exists\beta) \quad S : [*]*, T, U : * \triangleright M : \{\alpha\}[S[\alpha]]T, N : S[U] \\ \vdash \text{unpack}_{S,T}(\text{pack}_{S,U}(N), \alpha.x.M\{\alpha\}[x]) = M\{U\}[N] : T$$

$$(\exists\eta) \quad S : [*]*, T : * \triangleright M : [\exists(\alpha.S[\alpha])]T, N : \exists(\alpha.S[\alpha]) \\ \vdash \text{unpack}_{S,T}(N, \alpha.x.M[\text{pack}_{S,\alpha}(x)]) = M[N] : T$$

Algebraic Models

An *algebraic model* of a polymorphic equational presentation is a polymorphic structure satisfying the axioms.

Example: Seely's PL-category semantics of System F yields an algebraic model of the equational presentation of System F.

Algebraic Models

An *algebraic model* of a polymorphic equational presentation is a polymorphic structure satisfying the axioms.

Example: Seely's PL-category semantics of System F yields an algebraic model of the equational presentation of System F.

Polymorphic Translations

Polymorphic translations $(\mathcal{U}, \mathcal{A}) \rightarrow (\mathcal{U}', \mathcal{A}')$ are given by homomorphisms $\varphi : \mathcal{U} \rightarrow \mathcal{U}'$ and $\vartheta : \mathcal{A} \rightarrow \varphi^* \mathcal{A}'$.

Example: Fujita's CPS translation from System F to the Existential λ -calculus is a polymorphic translation.

Polymorphic Equational Logic

Judgements:

$$Z \triangleright \vec{\alpha} \mid \Gamma \vdash_{\mathbf{U}} s = t : \tau$$

where \mathbf{U} is a type universe, Z is a set of metavariable declarations, $\vec{\alpha} \mid \Gamma \vdash \tau$ is a context, and $Z \triangleright \vec{\alpha} \mid \Gamma \vdash s, t : \tau$ are meta-terms.

Polymorphic Equational Logic

Judgements:

$$Z \triangleright \vec{\alpha} \mid \Gamma \vdash_{\mathbf{U}} s = t : \tau$$

where \mathbf{U} is a type universe, Z is a set of metavariable declarations, $\vec{\alpha} \mid \Gamma \vdash \tau$ is a context, and $Z \triangleright \vec{\alpha} \mid \Gamma \vdash s, t : \tau$ are meta-terms.

Rules:

Congruence of all the algebraic structure plus *universe shift*:

$$\frac{Z \triangleright \vec{\alpha} \mid \Gamma \vdash_{\mathbf{U}} s = t : \tau}{(\mathbf{Z}) \triangleright \vec{\alpha} \mid (\mathbf{\Gamma}) \vdash_{\mathbf{V}} (\mathbf{s}) = (\mathbf{t}) : (\mathbf{\tau})} \quad (_ - _) : \mathbf{U} \rightarrow \mathbf{V}$$