Second-Order and Dependently-Sorted Abstract Syntax

Marcelo Fiore

Computer Laboratory
University of Cambridge

LICS 2008
24.VI.2008
Motivation I

Research Programme

MATHEMATICAL THEORY OF SYNTAX

see Fiore, Plotkin & Turi [4], Fiore [5, 7, 8]

◮ Algebraic:
  ♦ initial algebra semantics
    (⇒ compositionality)
  ♦ structural recursion
  ♦ induction principle

◮ Comprehensive:
  ♦ variable binding, α-equivalence
  ♦ capture-avoiding simultaneous and single-variable substitution
  ♦ term meta-variables, meta-substitution
  ♦ mono and multi sorting
  ♦ sort dependency
  ♦ linear, cartesian, mixed contexts
Motivation II

Research Programme

MATHEMATICAL FRAMEWORK FOR EQUATIONAL AND REWRITING LOGICAL FRAMEWORKS

see Fiore & Hur [9]
Motivation II

Research Programme

MATHEMATICAL FRAMEWORK FOR
EQUATIONAL AND REWRITING
LOGICAL FRAMEWORKS

see Fiore & Hur [9]

Idea (I)

Model

abstract syntax
(strong monad)

(enriched)
universe of
discourse
Idea (II)

Model

universe of discourse

Theory

abstract syntax
Idea (II)

Model

\[ S \]

universe of discourse

abstract syntax

Theory

Deductive System

\[ \mathcal{A}, \mathcal{R} \vdash f \equiv f' : C \rightarrow TA \]

\[ \mathcal{A}, \mathcal{R} \vdash f > f' : C \rightarrow TA \]

axioms

arities

rewrites

co-arities

sound for a canonical algebraic model theory

+ framework for completeness
Idea (III)

Deductive System

\[ \mathcal{A}, \mathcal{R} \models f \equiv f' : C \to TA \]
\[ \mathcal{A}, \mathcal{R} \models f > f' : C \to TA \]

sound for a canonical algebraic model theory
+ framework for completeness

concretion
/ / interpretation
\v
\v
syntactic structure
Idea (III)

Deductive System

\[ \mathcal{A}, \mathcal{R} \triangleright f \equiv f' : C \rightarrow TA \]
\[ \mathcal{A}, \mathcal{R} \triangleright f > f' : C \rightarrow TA \]

sound for a canonical algebraic model theory + framework for completeness

syntactic structure

Equational and Rewriting Logical Framework

\[ \mathcal{A}, \mathcal{R} \triangleright \Gamma \vdash t \equiv t' \]
\[ \mathcal{A}, \mathcal{R} \triangleright \Gamma \vdash t > t' \]
Example I

syntactic structure

= signature: \( \Sigma = \{ \Sigma_n \in Set \}_{n \in \mathbb{N}} \)
Example I

Equational and Rewriting Logic

\[ \Sigma \text{-Alg} \]

\[ \downarrow \quad \downarrow \]

\[ \text{Set} \]

\[ \Uparrow \quad \Uparrow \]

\[ T_\Sigma \]

\[ T_\Sigma(X) \cong X + \bigsqcup_{n \in \mathbb{N}} \Sigma_n \times (T_\Sigma X)^n \]

interpretation

syntactic structure

= signature: \( \Sigma = \{ \Sigma_n \in \text{Set} \}_{n \in \mathbb{N}} \)
syntactic structure
  = signature: $\Sigma = \{ \Sigma_\sigma \in \text{Set}^S \}_{\sigma \in S^*}$

**NB:** General method for the extension from the mono-sorted to the multi-sorted case.
Example II

Multi-Sorted
Equational and Rewriting Logic

\[
\Sigma \text{-Alg} \downarrow \downarrow \\
\downarrow \downarrow \\
\text{Set}^S \\
\text{Σ}_\Sigma \circ \circ \\
\]

\[
(T_\Sigma X)_s \cong X_s + \bigsqcup_{\sigma = (s_1 \ldots s_n) \in S^*} \Sigma_{\sigma,s} \times \prod_{i=1}^n (T_\Sigma X)_{s_i}
\]

interpretation

syntactic structure

= signature: \( \Sigma = \{ \Sigma_\sigma \in \text{Set}^S \}_{\sigma \in S^*} \)

**NB:** General method for the extension from the mono-sorted to the multi-sorted case.
Example III

see Fiore & Hur [9]

\[ \text{Nom} \]

syntactic structure see Clouston & Pitts [6]

= signature: \( \Sigma = \{ \Sigma_n \in \text{Nom} \}_{n \in \mathbb{N}} \)
Example III

\[ \Sigma \text{-Alg} \]

\[ \Downarrow \quad \Downarrow \]

\[ \text{Nom} \]

\[ T_\Sigma(X) \cong X + \bigsqcup_{n \in \mathbb{N}} \Sigma_n \times (T_\Sigma X)^n \]

interpretation

syntactic structure

= signature: \( \Sigma = \{ \Sigma_n \in \text{Nom} \}_{n \in \mathbb{N}} \)
Example III

Synthetic Nominal
Equational and Rewriting Logic

see Fiore & Hur [9]

\[ \Sigma \text{-Alg} \]
\[ \downarrow \downarrow \]
\[ \text{Nom} \]
\[ \Uparrow \Uparrow \]
\[ \Uparrow \text{T}_\Sigma \]

\[ T_\Sigma(X) \cong X + \bigsqcup_{n \in \mathbb{N}} \Sigma_n \times (T_\Sigma X)^n \]

interpretation \[ \downarrow \downarrow \]
concretion \[ \downarrow \downarrow \]
syntactic structure \[ \downarrow \downarrow \]

= signature: \[ \Sigma = \{ \Sigma_n \in \text{Nom} \}_{n \in \mathbb{N}} \]

\[ \begin{array}{c}
\text{arity} \\
\overbrace{a_1, \ldots, a_m} \quad \text{co-arity} \\
\overbrace{x_1[l_1], \ldots, x_n[l_n]} \end{array} \]

\[ a_1, \ldots, a_m ; x_1[l_1], \ldots, x_n[l_n] \vdash t \equiv t' \]
\[ a_1, \ldots, a_m ; x_1[l_1], \ldots, x_n[l_n] \vdash t > t' \]

e.g. \[ (\eta) \quad a, b ; x[1] \vdash \lambda_a x[a] \equiv \lambda_b x[b] \]
Example IV

Second-Order
Equational and Rewriting Theories

From the mathematical theory of second-order abstract syntax developed in Part I of the paper
The paradigmatic second-order theory:

\[ \Sigma_{\lambda} = 0 @ 0 \quad \text{(application)} \]

\[ \lambda(1) \quad \text{(abstraction)} \]

(\(\beta\)) \(M[1], N[0] \)

\[ \vdash \lambda((x)M[[x]]) @ N[] = M[N[]] \]

(\(\eta\)) \(M[] \vdash \lambda((x) M[] @ [x]) = M[] \)

compare Klop [1], Pigozzi & Salibra [2]
The paradigmatic second-order theory:

\[ \Sigma_\lambda = 0 \circ 0 \quad (\text{application}) \]

\[ \lambda(1) \quad (\text{abstraction}) \]

\[(\beta) \ M[1], N[0] \]

\[ \vdash \lambda((x)M[\lfloor x \rfloor]) \circ N[] = M[N[]] \]

\[(\eta) \ M[] \vdash \lambda((x) M[] @ \lfloor x \rfloor) = M[] \]

compare Klop [1], Pigozzi & Salibra [2]

The syntactic theory should account for:

- variables and meta-variables
- variable binding and \(\alpha\)-equivalence
- capture-avoiding and meta substitution
- mono and multi sorting
Second-Order Abstract Syntax

Model

finite sets and functions

see Fiore, Plotkin & Turi [4]

\[ \chi \in \mathbf{Set}^F \] is a functor

\[
\begin{align*}
\chi \Gamma (\Gamma \in F) \\
F(\Gamma, \Delta) &\rightarrow \mathbf{Set}(\chi \Gamma, \chi \Delta)
\end{align*}
\]

E.g. the object of variables is \( V\Gamma = \Gamma \)

syntactic structure =

- signature: \( \Sigma = \{ \sum_n \in \mathbf{Set}^F \}_{n \in \mathbb{N}^*} \)
Second-Order Abstract Syntax

Model

finite sets and functions

\[ \Sigma \subseteq \text{Set}^F \]

see Fiore, Plotkin & Turi [4]

\[ \Sigma(X) = \bigsqcup_{n=(n_1 \ldots n_k) \in \mathbb{N}^*} \Sigma_n \times \prod_{i=1}^k X^{V_{n_i}} \]

\[ X \in \text{Set}^F \] is a functor

\[ \begin{cases} 
X \Gamma (\Gamma \in F) \\
F(\Gamma, \Delta) \rightarrow \text{Set}(X\Gamma, X\Delta)
\end{cases} \]

E.g. the object of variables is \( \forall \Gamma = \Gamma \)

interpretation

syntactic structure =

\[ \text{signature: } \Sigma = \{ \Sigma_n \in \text{Set}^F \}_{n \in \mathbb{N}^*} \]
Second-Order Abstract Syntax

Model

finite sets and functions

\[
\Sigma(X) = \bigsqcup_{n=(n_1 \ldots n_k) \in \mathbb{N}^*} \Sigma_n \times \prod_{i=1}^{k} X^{V^{n_i}}
\]

\(\chi \in \text{Set}^F\) is a functor

\[
\begin{cases} 
\chi(\Gamma) (\Gamma \in F) \\
F(\Gamma, \Delta) \rightarrow \text{Set}(\chi\Gamma, \chi\Delta)
\end{cases}
\]

E.g. the object of variables is \(\forall \Gamma = \Gamma\)

interpretation

syntactic structure =

- signature: \(\Sigma = \{ \Sigma_n \in \text{Set}^F \}_{n \in \mathbb{N}^*}\)
- substitution
Algebras with substitution
(\(\Sigma\)-monoids)

see Fiore, Plotkin & Turi [4]

- algebra structure:
  \[
  \Sigma X \xrightarrow{\xi} X
  \]

- substitution structure:
  \[
  \text{monoid} \quad V \xrightarrow{e} X \xleftarrow{m} X \bullet X
  \]

  \[
  \begin{pmatrix}
  \Gamma & \xrightarrow{} & X\Gamma & \xleftarrow{} & X\Delta \times (X\Gamma)^\Delta \\
  \equiv & & & & \\
  \text{subject to the laws of substitution}
  \end{pmatrix}
  \]

subject to the compatibility condition:

\[
\begin{array}{c}
\Sigma (X) \bullet X \xrightarrow{\xi \bullet X} \Sigma (X \bullet X) \xrightarrow{\Sigma m} \Sigma X \\
\end{array}
\]

\[
\begin{array}{c}
\xi, \\
\end{array}
\]

\[
\begin{array}{c}
X \bullet X \xrightarrow{m} X
\end{array}
\]
Thm:

1. General result:

\[ \mathcal{M}_\Sigma(X) \cong V + X \bullet \mathcal{M}_\Sigma(X) + \Sigma(\mathcal{M}_\Sigma X) \]
Thm:

1. General result:
   \[ M_\Sigma(X) \cong V + X \cdot M_\Sigma(X) + \Sigma(M_\Sigma X) \]

2. For \( \Sigma \) induced by a binding signature, 
   \( M_\Sigma \) is a strong monad.

Rem: Need to develop a theory of strengths.
Thm:

1. General result:
\[ \mathcal{M}_\Sigma(X) \equiv V + X \cdot \mathcal{M}_\Sigma(X) + \Sigma(\mathcal{M}_\Sigma X) \]

2. For \( \Sigma \) induced by a binding signature, \( \mathcal{M}_\Sigma \) is a strong monad.

Rem: Need to develop a theory of strengths.

Syntactic theory for variables, meta-variables, variable binding, \( \alpha \)-equivalence, capture-avoiding substitution, meta-substitution.
Syntax:

For \( X \) an object of meta-variables,

\[
t \in M_{\Sigma}(X) \Gamma
\]

\[
::= [x] \quad (x \in \Gamma)
\]

\[
\mid M[t_1, \ldots, t_\ell]
\]

\[
\mid f((\vec{x}_1)t_1, \ldots, (\vec{x}_k)t_k))
\]

\[
\left( \begin{array}{c}
M \in X(\ell) \\
t_i \in (M_{\Sigma X}) \Gamma \\
\end{array} \right)
\]

\[
\left( \begin{array}{c}
f \in \Sigma(|\vec{x}_1|, \ldots, |\vec{x}_k|) \\
t_i \in (M_{\Sigma X}) \Gamma, \vec{x}_i \\
\end{array} \right)
\]
Syntactic Theory (I)

Syntax:

For $X$ an object of meta-variables,

$$t \in \mathcal{M}_\Sigma(X) \Gamma$$

$$::= \llbracket x \rrbracket \quad (x \in \Gamma)$$

$$| \ M[t_1, \ldots, t_\ell] \quad \left( \begin{array}{c} M \in X(\ell) \\ t_i \in (\mathcal{M}_\Sigma X) \Gamma \end{array} \right)$$

$$| \ f((\vec{x}_1 t_1, \ldots, (\vec{x}_k t_k)) \quad \left( \begin{array}{c} f \in \Sigma(|\vec{x}_1| \ldots |\vec{x}_k|) \\ t_i \in (\mathcal{M}_\Sigma X) \Gamma, \vec{x}_i \end{array} \right)$$

Capture-avoiding substitution:

$$\mathcal{M}_\Sigma(X) \bullet \mathcal{M}_\Sigma(X) \rightarrow \mathcal{M}_\Sigma(X)$$

$$\left( \equiv \mathcal{M}_\Sigma(X) \Delta \times (\mathcal{M}_\Sigma(X) \Gamma)^\Delta \rightarrow \mathcal{M}_\Sigma(X) \Gamma \right)$$
Syntactic Theory (I)

Syntax:

For $X$ an object of meta-variables,

$$t \in \mathcal{M}_\Sigma(X)_\Gamma$$

$$::= [x] \quad (x \in \Gamma)$$

$$\mid M[t_1, \ldots, t_\ell] \quad \left( \begin{array}{c} M \in X(\ell) \\ t_i \in (\mathcal{M}_\Sigma X)_\Gamma \end{array} \right)$$

$$\mid f((x_1^\tau t_1, \ldots, (x_k^\tau t_k)) \quad \left( \begin{array}{c} f \in \Sigma(|x_1^\tau| \ldots |x_k^\tau|) \\ t_i \in (\mathcal{M}_\Sigma X)_{\Gamma, x_i^\tau} \end{array} \right)$$

Capture-avoiding substitution:

$$\mathcal{M}_\Sigma(X) \bullet \mathcal{M}_\Sigma(X) \longrightarrow \mathcal{M}_\Sigma(X)$$

$$\left( \equiv \mathcal{M}_\Sigma(X)_\Delta \times (\mathcal{M}_\Sigma(X)_\Gamma)^\Delta \longrightarrow \mathcal{M}_\Sigma(X)_\Gamma \right)$$

Meta-substitution:

$$\mathcal{M}_\Sigma(X) \times (\mathcal{M}_\Sigma(Y))^X \longrightarrow \mathcal{M}_\Sigma(Y)$$

$$\left( \equiv \mathcal{M}_\Sigma(X)_\Gamma \times \prod_{\ell \in \mathbb{N}} X(\ell) \Rightarrow ((\mathcal{M}_\Sigma Y)^\nu_{\ell})_\Gamma \rightarrow \mathcal{M}_\Sigma(Y)_\Gamma \right)$$
Syntactic Theory (II)

- *Canonical specification* and derived *correct definition* of
  - variable renaming,
  - capture-avoiding simultaneous substitution,
  - meta-variable renaming,
  - meta-substitution.

- Canonical *algebraic model theory*.
Dependently-Sorted Abstract Syntax

Universe of discourse:

\[ \text{Mon} \left( \text{Mod}_S (\overline{\mathcal{C}[S]}) \right) \]

- monoids
- models
- dependently-sorted signature
- category of contexts

- embodies \textit{sort dependency}
  
- induces \textit{dependently-sorted substitution}.
Some Further Directions

► Abstract syntax with sharing.
► Applications to rewriting theory.
► Second-order theory translations.
► Algebraic foundations for type theory.
References


