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]

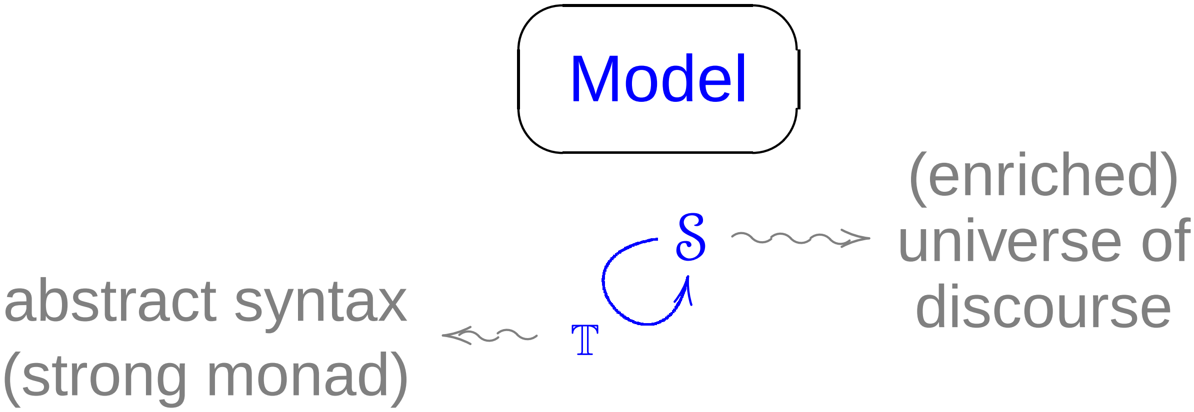
Motivation II

Research Programme

MATHEMATICAL FRAMEWORK FOR
EQUATIONAL AND REWRITING
LOGICAL FRAMEWORKS

see Fiore & Hur [9]

Idea (I)



Model

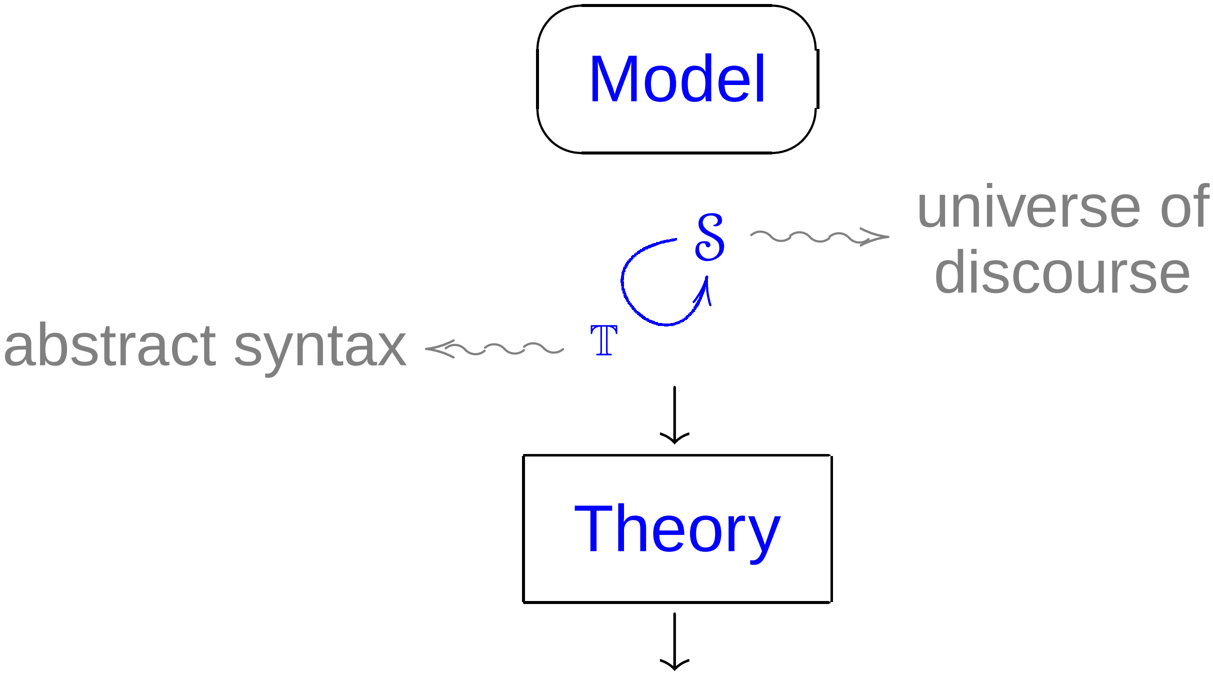
abstract syntax
(strong monad)

$\mathbb{T}$

$\mathcal{S}$

(enriched)
universe of
discourse

Idea (II)



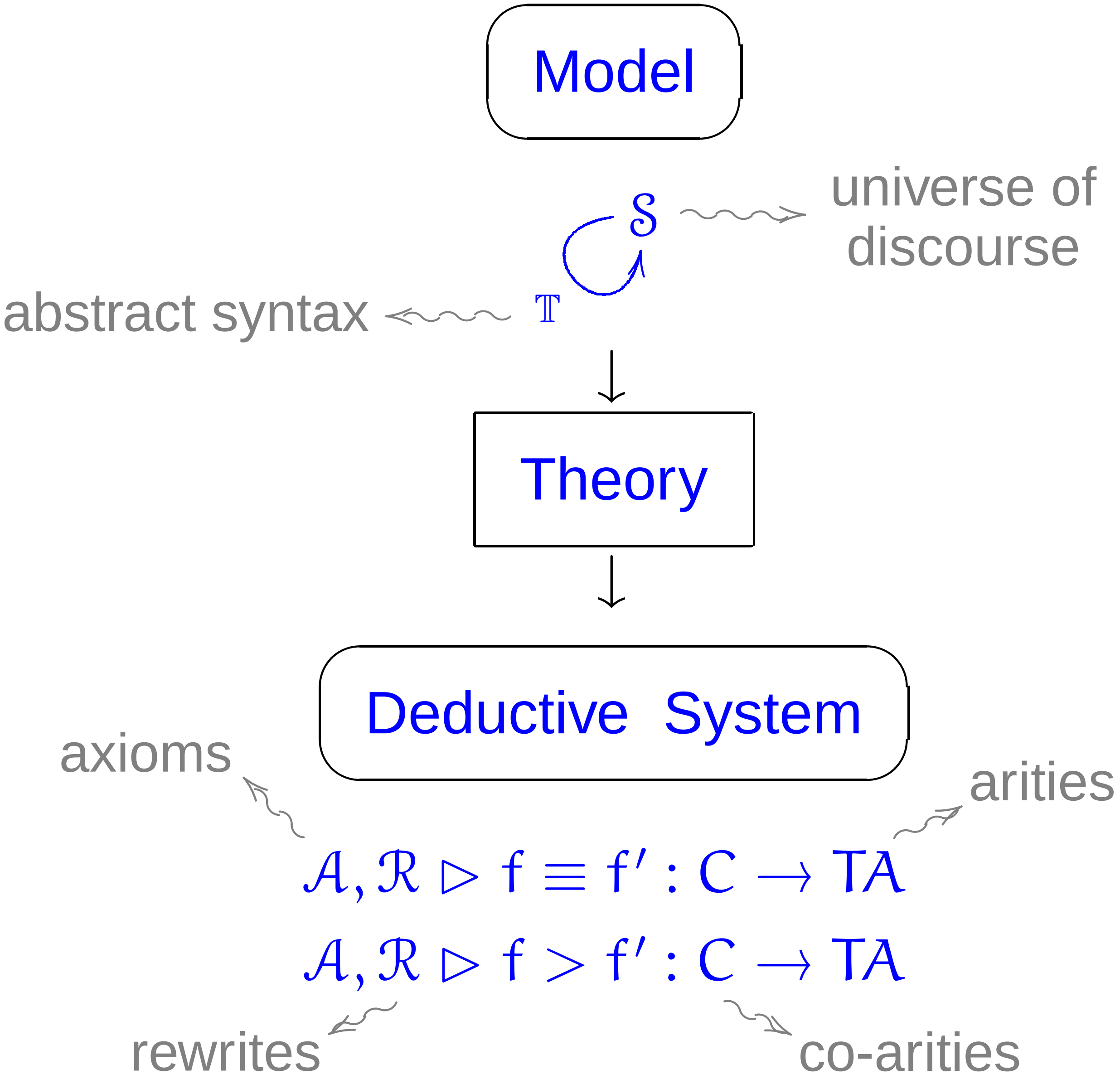
Model

$\mathcal{S}$ universe of discourse

abstract syntax $\mathbb{T}$

Theory

Idea (II)



Model

$\mathbb{T} \curvearrowright \mathcal{S}$

universe of discourse

abstract syntax

Theory

Deductive System

axioms

arities

$$\mathcal{A},\mathcal{R} \rhd f \equiv f' : C \to TA$$

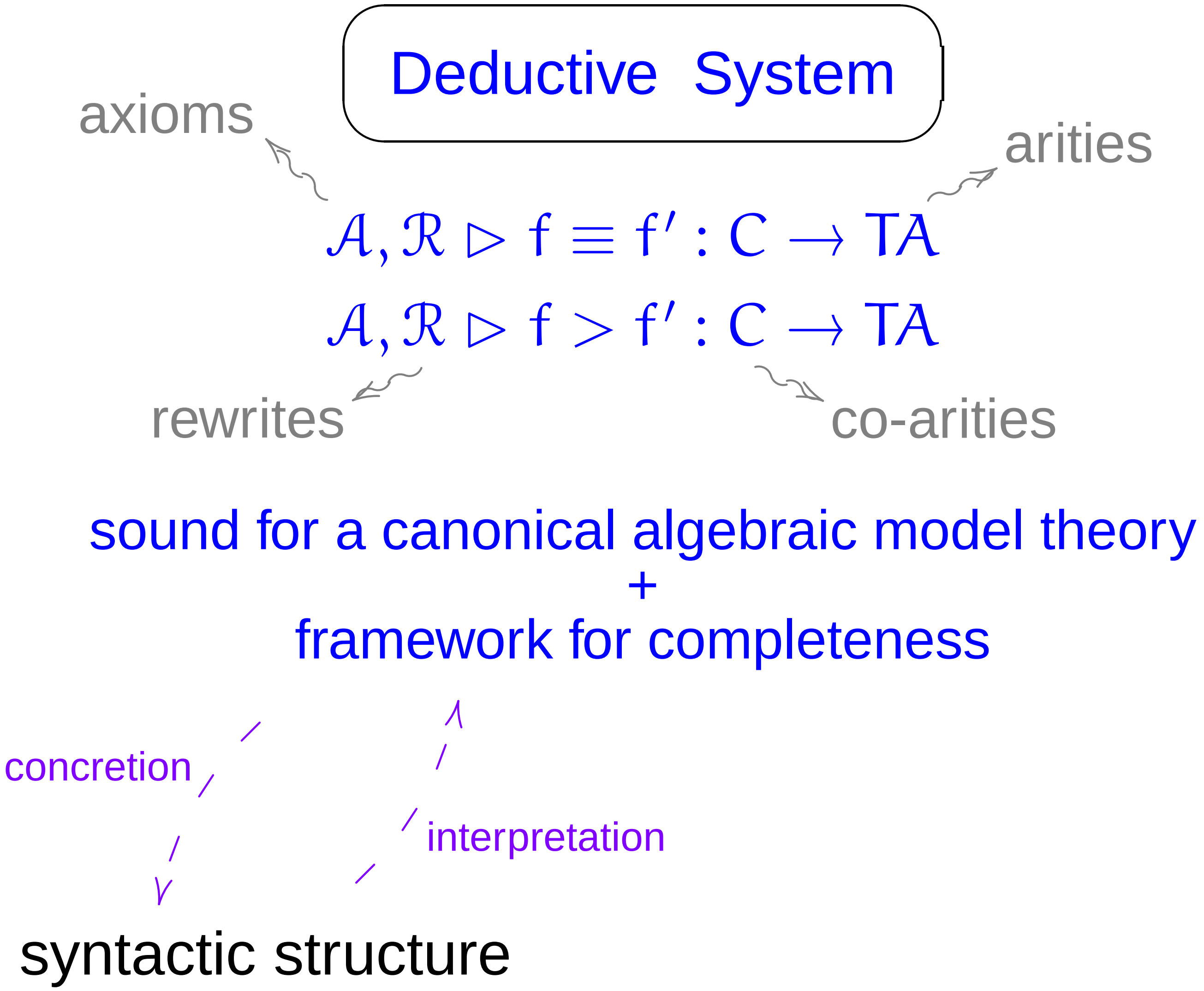
$$\mathcal{A},\mathcal{R} \rhd f > f' : C \to TA$$

rewrites

co-arities

sound for a canonical algebraic model theory
+
framework for completeness

Idea (III)



Deductive System

axioms

arities

$$\mathcal{A},\mathcal{R} \rhd f \equiv f' : C \to TA$$

$$\mathcal{A},\mathcal{R} \rhd f > f' : C \to TA$$

rewrites

co-arities

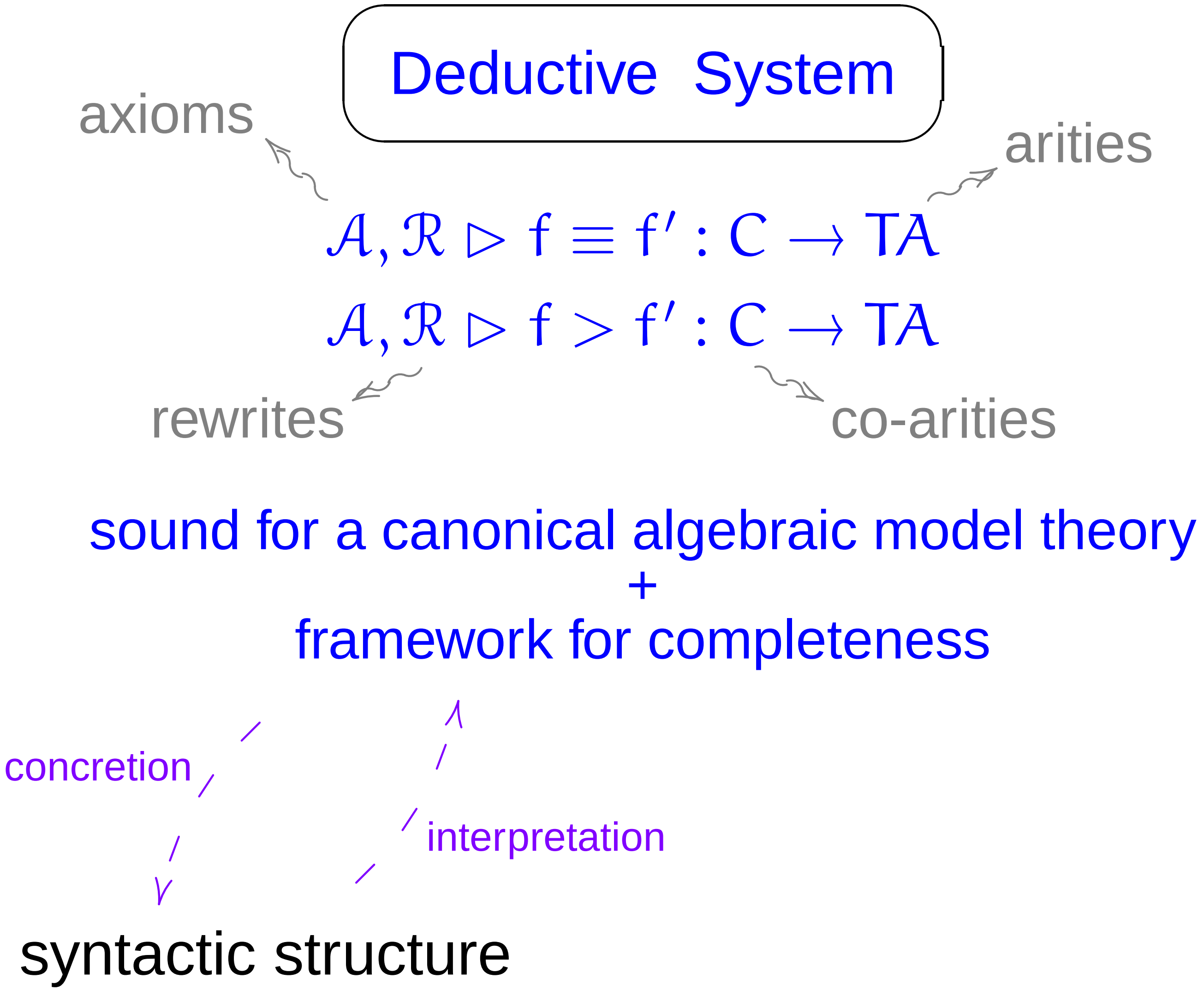
sound for a canonical algebraic model theory
+
framework for completeness

concretion

interpretation

syntactic structure

Idea (III)



Deductive System

axioms

arities

$$\mathcal{A},\mathcal{R} \rhd f \equiv f' : C \to TA$$

$$\mathcal{A},\mathcal{R} \rhd f > f' : C \to TA$$

rewrites

co-arities

sound for a canonical algebraic model theory
+
framework for completeness

concretion

interpretation

syntactic structure

Equational and Rewriting Logical Framework

$$\mathcal{A},\mathcal{R} \rhd \Gamma \vdash t \equiv t'$$

$$\mathcal{A},\mathcal{R} \rhd \Gamma \vdash t > t'$$

Example I

$$\boldsymbol{\mathcal{S}et}$$

syntactic structure

= signature: $\Sigma = \{ \Sigma_n \in \boldsymbol{\mathcal{S}et} \}_{n \in \mathbb{N}}$

Example I

Equational and Rewriting Logic

$$\begin{tikzcd}
\mathbf{\Sigma\text{-}Alg} \arrow[d, shift left=2] \arrow[d, phantom, "\dashv"] \\
\boldsymbol{\mathcal{S}et} \arrow[u, shift left=2] \arrow[loop below, "\mathbb{T}_\Sigma"]
\end{tikzcd}$$

$$\mathsf{T}_\Sigma(\mathsf{X}) \;\cong\; \mathsf{X} + \coprod_{n \in \mathbb{N}} \Sigma_n \times (\mathsf{T}_\Sigma \mathsf{X})^n$$

interpretation

syntactic structure

= signature: $\Sigma = \{\, \Sigma_n \in \boldsymbol{\mathcal{S}et} \,\}_{n \in \mathbb{N}}$

Example II

$$\mathcal{S}et^{\mathsf{S}}$$

syntactic structure

= signature: $\Sigma = \{\, \Sigma_\sigma \in \mathcal{S}et^{\mathsf{S}} \,\}_{\sigma \in \mathsf{S}^*}$

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

Example II

Multi-Sorted
Equational and Rewriting Logic

$$\begin{tikzcd} \Sigma\text{-}\mathbf{Alg} \arrow[d, shift left=2] \arrow[d, phantom, "\dashv"] \\ \mathcal{S}et^{S} \arrow[u, shift left=2] \arrow[loop below, "\mathbb{T}_\Sigma"] \end{tikzcd}$$

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

interpretation

syntactic structure

= signature: $\Sigma = \{\,\Sigma_\sigma \in \mathcal{S}et^{S}\,\}_{\sigma\in S^*}$

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

Example III

see Fiore & Hur [9]

$$\mathcal{N}om$$

syntactic structure see Clouston & Pitts [6]

= signature: $\Sigma = \{ \Sigma_n \in \mathcal{N}om \}_{n \in \mathbb{N}}$

Example III

see Fiore & Hur [9]

$$\begin{tikzcd} \boldsymbol{\Sigma\text{-}\mathbf{Alg}} \arrow[d, shift left=1.5ex] \arrow[d, phantom, "\dashv"] \\ \boldsymbol{\mathcal{N}om} \arrow[u, shift left=1.5ex] \arrow[loop below left, "\mathbb{T}_\Sigma"] \end{tikzcd}$$

$$\mathsf{T}_\Sigma(\mathsf{X}) \cong \mathsf{X} + \coprod_{n\in\mathbb{N}} \Sigma_n \times (\mathsf{T}_\Sigma \mathsf{X})^n$$

interpretation

syntactic structure see Clouston & Pitts [6]

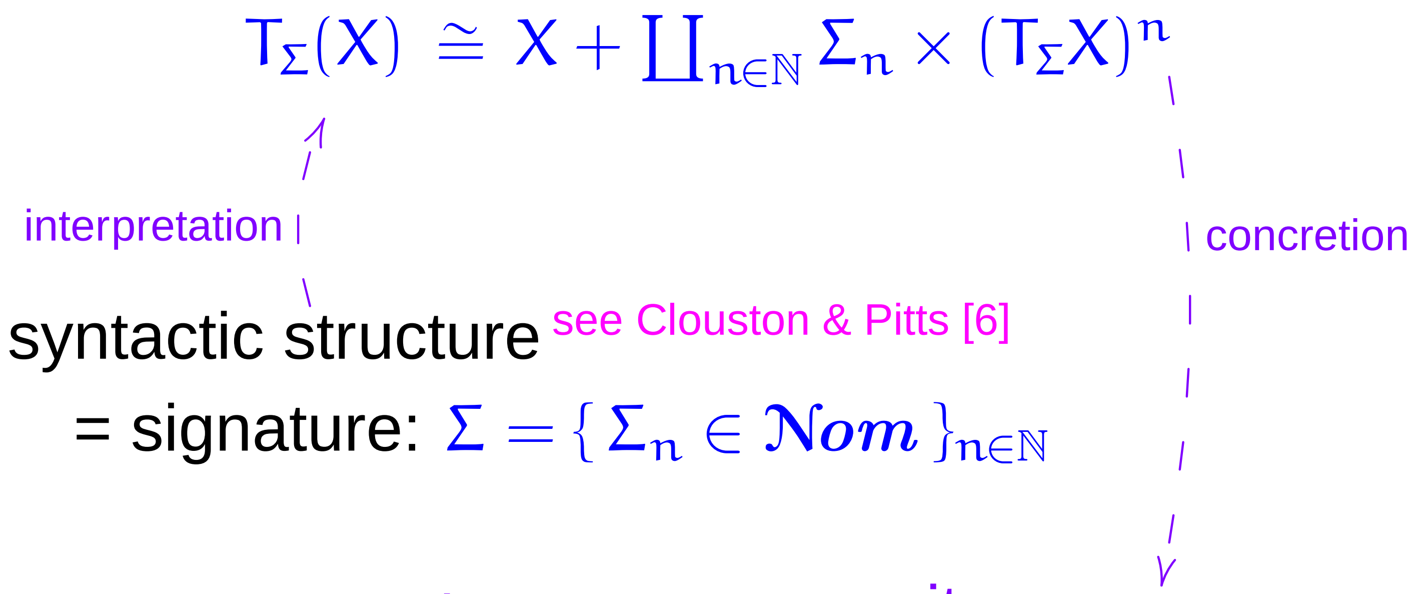
= signature: $\Sigma = \{\, \Sigma_n \in \boldsymbol{\mathcal{N}om} \,\}_{n\in\mathbb{N}}$

Example III

Synthetic Nominal Equational and Rewriting Logic

see Fiore & Hur [9]

$$\begin{tikzcd} \Sigma\text{-}\mathbf{Alg} \arrow[d, shift left] \arrow[d, phantom, "\dashv"] \\ \mathcal{N}om \arrow[u, shift left] \arrow[loop below, "\mathbb{T}_\Sigma"] \end{tikzcd}$$



$$T_\Sigma(X) \;\cong\; X + \textstyle\coprod_{n\in\mathbb{N}} \Sigma_n \times (T_\Sigma X)^n$$

syntactic structure see Clouston & Pitts [6]

= signature: $\Sigma = \{\, \Sigma_n \in \mathcal{N}om \,\}_{n\in\mathbb{N}}$

$$\overbrace{a_1,\ldots,a_m}^{\text{arity}} \;;\; \overbrace{x_1[\ell_1],\ldots,x_n[\ell_n]}^{\text{co-arity}} \;\vdash t \equiv t'$$

$$a_1,\ldots,a_m \;;\; x_1[\ell_1],\ldots,x_n[\ell_n] \;\vdash t > t'$$

e.g. (η) $a, b \;;\; x[1] \;\vdash \lambda_a\, x[a] \equiv \lambda_b\, x[b]$

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:

$$\Sigma_\lambda = \; 0 \mathbin{@} 0 \quad \text{(application)}$$

$$\lambda(1) \quad \text{(abstraction)}$$

$$(\beta) \;\; \mathsf{M}[1], \mathsf{N}[0] \vdash \lambda\big((x)\mathsf{M}[\lfloor x \rfloor]\big) \mathbin{@} \mathsf{N}[\,] = \mathsf{M}\big[\mathsf{N}[\,]\big]$$

$$(\eta) \;\; \mathsf{M}[\,] \vdash \lambda\big((x)\; M[\,] \mathbin{@} \lfloor x \rfloor\big) = M[\,]$$

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

- The paradigmatic second-order theory:

$$\Sigma_\lambda = \begin{array}{ll} 0 \mathbin{@} 0 & \text{(application)} \\ \lambda(1) & \text{(abstraction)} \end{array}$$

$$(\beta)\ \ \mathsf{M}[1], \mathsf{N}[0] \\ \qquad \vdash \lambda\big((x)\mathsf{M}[\lfloor x \rfloor]\big) \mathbin{@} \mathsf{N}[\,] = \mathsf{M}\big[\mathsf{N}[\,]\big]$$

$$(\eta)\ \ \mathsf{M}[\,] \vdash \lambda\big((x)\ M[\,] \mathbin{@} \lfloor x \rfloor\big) = M[\,]$$

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

- The syntactic theory should account for:
  - variables and meta-variables
  - variable binding and α-equivalence
  - capture-avoiding and meta substitution
  - mono and multi sorting

Second-Order Abstract Syntax

Model

finite sets and functions

$\mathcal{S}et^{\mathbb{F}}$

see Fiore, Plotkin & Turi [4]

$$X \in \mathcal{S}et^{\mathbb{F}} \text{ is a functor } \begin{cases} X\Gamma \ (\Gamma \in \mathbb{F}) \\ \mathbb{F}(\Gamma, \Delta) \to \mathcal{S}et(X\Gamma, X\Delta) \end{cases}$$

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

syntactic structure =

- signature: $\Sigma = \{ \Sigma_n \in \mathcal{S}et^{\mathbb{F}} \}_{n \in \mathbb{N}^*}$

Second-Order Abstract Syntax

Model

finite sets and functions

$$\begin{tikzcd} \mathcal{S}et^{\mathbb{F}} \arrow[loop left, "\Sigma"] \end{tikzcd}$$

see Fiore, Plotkin & Turi [4]

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

$X \in \mathcal{S}et^{\mathbb{F}}$ is a functor $\left\{ \begin{array}{l} X\Gamma \ (\Gamma \in \mathbb{F}) \\ \mathbb{F}(\Gamma, \Delta) \to \mathcal{S}et(X\Gamma, X\Delta) \end{array} \right.$

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

interpretation

syntactic structure =

- signature: $\Sigma = \{ \Sigma_n \in \mathcal{S}et^{\mathbb{F}} \}_{n \in \mathbb{N}^*}$

Second-Order Abstract Syntax

Model

finite sets and functions

$$\begin{tikzcd} \mathcal{S}et^{\mathbb{F}} \arrow[loop left, "\Sigma"] \end{tikzcd}$$

see Fiore, Plotkin & Turi [4]

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

$X \in \mathcal{S}et^{\mathbb{F}}$ is a functor $\left\{\begin{array}{l} X\Gamma \ (\Gamma \in \mathbb{F}) \\ \mathbb{F}(\Gamma,\Delta) \to \mathcal{S}et(X\Gamma, X\Delta) \end{array}\right.$

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

interpretation

syntactic structure =

- signature: $\Sigma = \{\, \Sigma_n \in \mathcal{S}et^{\mathbb{F}} \,\}_{n\in\mathbb{N}^*}$

\+

- **substitution**

Algebras with substitution

(Σ-monoids)

see Fiore, Plotkin & Turi [4]

- algebra structure:

$$\Sigma X \xrightarrow{\xi} X$$

- substitution structure:

$$\text{monoid} \quad V \xrightarrow{e} X \xleftarrow{m} X \bullet X$$

$$\left(\equiv \begin{array}{c} \Gamma \longrightarrow X\Gamma \longleftarrow X\Delta \times (X\Gamma)^{\Delta} \\ \text{subject to the laws of substitution} \end{array} \right)$$

subject to the compatibility condition:

$$\begin{tikzcd}
\Sigma(X) \bullet X \arrow[r] \arrow[d, "\xi \bullet X"'] & \Sigma(X \bullet X) \arrow[r, "\Sigma m"] & \Sigma X \arrow[d, "\xi"] \\
X \bullet X \arrow[rr, "m"'] & & X
\end{tikzcd}$$

Model

$$\begin{tikzcd}
\Sigma\text{-}\mathbf{Mon} \arrow[d, shift left=2] \\
\boldsymbol{\mathcal{S}et}^{\mathbb{F}} \arrow[u, shift left=2] \arrow[u, phantom, "\dashv"] \arrow[loop below, "\mathcal{M}_\Sigma"]
\end{tikzcd}$$

Thm:

1. General result:

$$\mathcal{M}_\Sigma(X) \cong V + X \bullet \mathcal{M}_\Sigma(X) + \Sigma(\mathcal{M}_\Sigma X)$$

Model

$$\begin{tikzcd}
\Sigma\text{-}\mathbf{Mon} \arrow[d, shift left=2] \arrow[d, phantom, "\dashv"] \\
\boldsymbol{\mathcal{S}et}^{\mathbb{F}} \arrow[u, shift left=2] \arrow[loop below, "\mathcal{M}_\Sigma"]
\end{tikzcd}$$

Thm:

1. General result:

$$\mathcal{M}_\Sigma(X) \cong V + X \bullet \mathcal{M}_\Sigma(X) + \Sigma(\mathcal{M}_\Sigma X)$$

2. For Σ induced by a binding signature,

$\mathcal{M}_\Sigma$ is a strong monad .

Rem: Need to develop a theory of strengths.

Model

$$\begin{tikzcd}
\Sigma\text{-}\mathbf{Mon} \arrow[d, shift left=2] \arrow[d, phantom, "\dashv"] \\
\mathcal{S}et^{\mathbb{F}} \arrow[u, shift left=2] \arrow[loop below, "\mathcal{M}_\Sigma"]
\end{tikzcd}$$

Thm:

1. General result:

$$\mathcal{M}_\Sigma(\mathsf{X}) \cong \mathrm{V} + \mathsf{X} \bullet \mathcal{M}_\Sigma(\mathsf{X}) + \Sigma(\mathcal{M}_\Sigma \mathsf{X})$$

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

Rem: Need to develop a theory of strengths.

concretion

Syntactic theory for variables, meta-variables, variable binding, α-equivalence, capture-avoiding substitution, meta-substitution.

Syntactic Theory (I)

- Syntax:

  For X an object of meta-variables,

$$
\begin{array}{lll}
t \in \mathcal{M}_\Sigma(X)_\Gamma & & \\
::= \lfloor x \rfloor & & (x \in \Gamma) \\
\quad \mid & \mathsf{M}[t_1, \ldots, t_\ell] & \left(\begin{array}{l} \mathsf{M} \in X(\ell) \\ t_i \in (M_\Sigma X)_\Gamma \end{array} \right) \\
\quad \mid & \mathsf{f}\big((\vec{x_1})t_1, \ldots, (\vec{x_k})t_k\big) & \left(\begin{array}{l} \mathsf{f} \in \Sigma_{(|\vec{x_1}|\ldots|\vec{x_k}|)} \\ t_i \in (M_\Sigma X)_{\Gamma,\vec{x_i}} \end{array} \right)
\end{array}
$$

Syntactic Theory (I)

- Syntax:

  For X an object of meta-variables,

$$
\begin{array}{ll}
t \in \mathcal{M}_\Sigma(X)_\Gamma & \\
\ ::= \lfloor x \rfloor & (x \in \Gamma) \\
\quad \mid \quad M[t_1, \ldots, t_\ell] & \left(\begin{array}{l} M \in X(\ell) \\ t_i \in (M_\Sigma X)_\Gamma \end{array} \right) \\
\quad \mid \quad f\big((\vec{x_1})t_1, \ldots, (\vec{x_k})t_k\big) & \left(\begin{array}{l} f \in \Sigma_{(|\vec{x_1}|\ldots|\vec{x_k}|)} \\ t_i \in (M_\Sigma X)_{\Gamma, \vec{x_i}} \end{array} \right)
\end{array}
$$

- Capture-avoiding substitution:

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

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

Syntactic Theory (I)

- Syntax:

  For X an object of meta-variables,

$$
\begin{array}{lll}
t \in \mathcal{M}_\Sigma(X)_\Gamma & & \\
\quad ::= \lfloor x \rfloor & & (x \in \Gamma) \\
\quad \mid \; M[t_1, \ldots, t_\ell] & & \left(\begin{array}{l} M \in X(\ell) \\ t_i \in (\mathrm{M}_\Sigma X)_\Gamma \end{array} \right) \\
\quad \mid \; f\big((\vec{x_1})t_1, \ldots, (\vec{x_k})t_k\big) & & \left(\begin{array}{l} f \in \Sigma_{(|\vec{x_1}|\ldots|\vec{x_k}|)} \\ t_i \in (\mathrm{M}_\Sigma X)_{\Gamma,\vec{x_i}} \end{array} \right)
\end{array}
$$

- Capture-avoiding substitution:

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

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

- Meta-substitution:

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

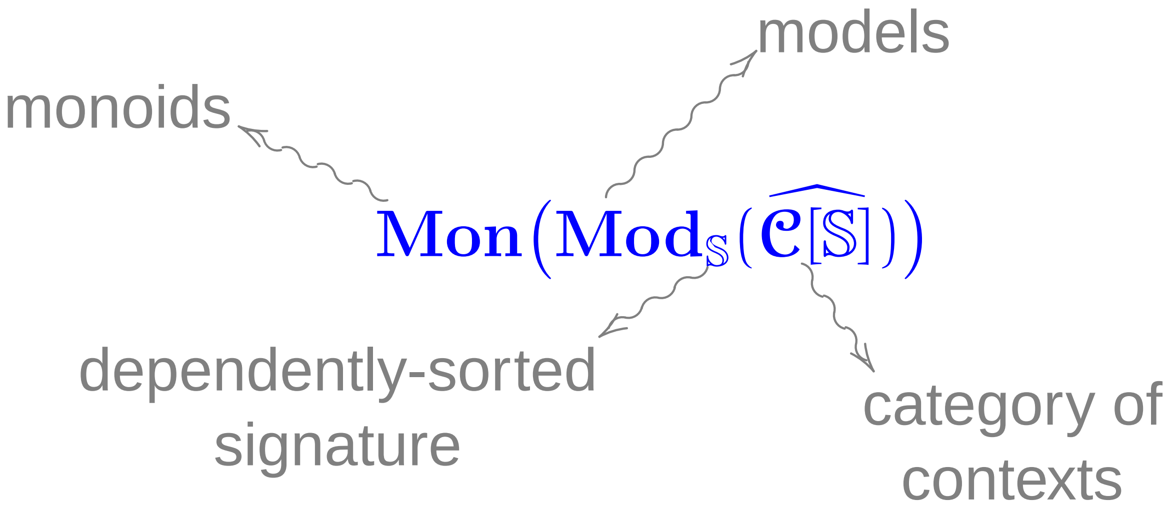
$$\left(\equiv \mathcal{M}_\Sigma(X)_\Gamma \times \prod_{\ell \in \mathbb{N}} X(\ell) \Rightarrow \left((\mathcal{M}_\Sigma Y)^{V^\ell} \right)_\Gamma \to \mathcal{M}_\Sigma(Y)_\Gamma \right)$$

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:



monoids

models

$$\mathbf{Mon}\Big(\mathbf{Mod}_{\mathbb{S}}\big(\widehat{\mathcal{C}[\mathbb{S}]}\big)\Big)$$

dependently-sorted signature

category of contexts

- embodies *sort dependency*

compare Makkai [3]

- induces *dependently-sorted substitution*.

Some Further Directions

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

References

[1] J. W. Klop. *Combinatory Reduction Systems.* Ph.D. thesis, Mathematical Centre Tracts 127, CWI, Amsterdam (1980).

[2] D. Pigozzi and A Salibra. The abstract variable-binding calculus. Studia Logica, Volume 55, Number 1, 1995.

[3] M. Makkai. First-order logic with dependent sorts, with applications to category theory. Preprint, 1997.

[4] M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *14th Annual IEEE Symposium on Logic in Computer Science (LICS'99)*, pages 193–202, 1999.

[5] M. Fiore. On the structure of substitution. Invited address for the *22nd Mathematical Foundations of Programming Semantics Conference (MFPS XXII)*, DISI, University of Genova, 2006. (Available from `http://www.cl.cam.ac.uk/~mpf23/`).

[6] R. Clouston and A. Pitts. Nominal Equational Logic. Electronic Notes in Theoretical Computer Science 172 (2007), pages 223–257.

[7] M. Fiore. A mathematical theory of substitution and its applications to syntax and semantics. Invited tutorial for the *Workshop on Mathematical Theories of Abstraction, Substitution and Naming in Computer Science*, International Centre for Mathematical Sciences (ICMS), 2007. (Available from `http://www.cl.cam.ac.uk/~mpf23/`).

[8] M. Fiore. Towards a mathematical theory of substitution. Invited talk for the *Annual International Conference on Category Theory*, Carvoeiro, Algarve (Portugal), 2007. (Available from `http://www.cl.cam.ac.uk/~mpf23/`).

[9] M. Fiore and C.-K. Hur. Term equational systems and logics. In *XXIV Conference on the Mathematical Foundations of Programming Semantics*. Electronic Notes in Theoretical Computer Science, 2008.