Modular abstract syntax trees (MAST): substitution tensors with second-class sorts

Marcelo Fiore, Ohad Kammar, Georg Moser, and Sam Staton

Scottish Programming Languages Seminar
Heriot-Watt University
4 June 2025

THE UNIVERSITY of EDINBURGH informatics lfcs Laboratory for Foundations of Computer Science BayesCentre

Funded by: THE ROYAL SOCIETY Advanced Research + Invention Agency ARIA erc European Research Council Established by the European Commission FWF Austrian Science Fund

Call-by-Value λ-calculus

$A, B, C ::=$	type		$V, W ::=$	value	
	β	base		x	variable
$\mid$	$A \to B$	function	$\mid$	$\lambda x : A.M$	function abst.
$\mid$	$(\!\|C_i : A_i \mid i \in I\|\!)$	record (I finite)	$\mid$	$(C_i : V_i \mid i \in I)$	record c'tor
$\mid$	$\{\!\|C_i : A_i \mid i \in I\|\!\}$	variant (I finite)	$\mid$	$A.C_i\ V$	variant c'tor
$\vdots$			$\vdots$		

$M, N, K, L ::=$		term
	$\mathsf{val}\ V$	value
$\mid$	$\mathbf{let}\ x_1 = M_1; \ldots; x_n = M_n\ \mathbf{in}\ N$	sequencing
$\mid$	$M @ N$	function application
$\mid$	$(C_1 : M_1, \ldots, C_n : M_n)$	record constructor
$\mid$	$\mathbf{case}\ M\ \mathbf{of}\ (C_1 x_1, \ldots, C_n x_n) \Rightarrow N$	record pattern match
$\mid$	$A.C_i M$	variant constructor
$\mid$	$\mathbf{case}\ M\ \mathbf{of}\ \{C_i x_i \Rightarrow M_i \mid i \in I\}\ N$	variant pattern match
$\vdots$		

Semantic perspective

Initial Algebra Semantics Programme

[Goguen and Thatcher'74]

Denotational semantics á la carte

homage to [Swierstra'08, Forster and Stark'20]

CBV customisation menu

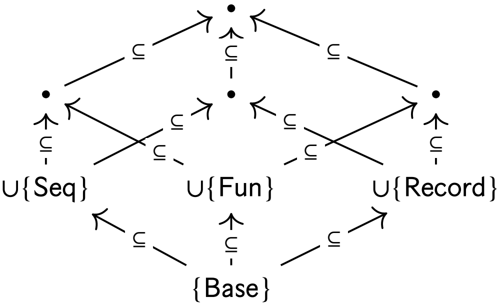
fragment	syntactic constructs	types	semantics
base	returning a value: **val**		strong monad over a Cartesian category
sequential	sequencing: **let**		
functions	abst., app. $(\lambda x.\ :\ A), (@)$	function $(\rightarrow)$	Kleisli exponentials
variants	c'tors, pattern match $A.C_i-$, $\textbf{case} - \textbf{of}$ $\{C_i x_i \Rightarrow - \mid i \in I\}$	variant $\{\!\{C_i : - \mid i \in I\}\!\}$	distributive category

⋮

Dream

Iterative semantic development

- Add syntax
- Add semantics

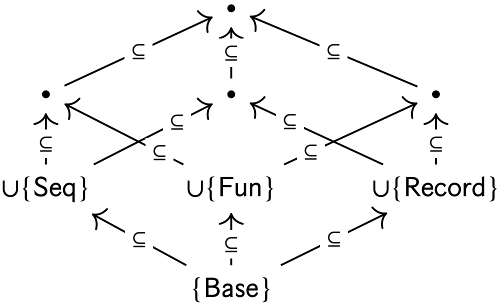


- Profit!

Dream vs. Reality

Iterative semantic development

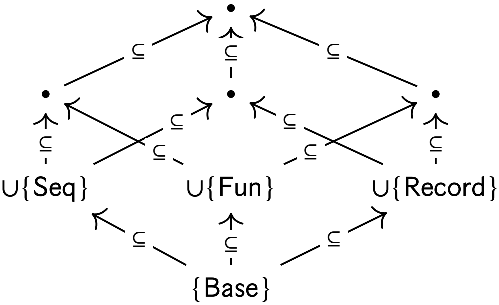
- Add syntax
- Add semantics
- Develop meta-theory:
  - Substitution lemma
  - Compositionality
  - Soundness
  - Adequacy
- Profit!



Dream vs. Bleak Reality

Iterative semantic development

- Add syntax
- Add semantics
- Develop meta-theory:
  - Substitution lemma
    Tedious and boring
  - Compositionality
    Tedious and boring
  - Soundness
  - Adequacy
- Profit!



Meta-theory: the tedious parts

Lemma (substitution)

Syntactic substitution corresponds to semantic composition:

$$[\![M\,[\theta]]\!] = [\![M]\!] \circ [\![\theta]\!]$$

Lemma (compositionality)

Composite semantics is independent of component syntax:

$$[\![C[M]]\!] = \mathrm{plug}([\![C[-]]\!]\,,[\![M]\!])$$

Meta-theory: the tedious parts

Lemma (substitution)

Syntactic substitution corresponds to semantic composition:

$$[\![M\,[\theta]]\!] = [\![M]\!] \circ [\![\theta]\!]$$

Proof.

Presupposes a syntactic substitution lemma. Typically several inductions over all constructs. □

Lemma (compositionality)

Composite semantics is independent of component syntax:

$$[\![C[M]]\!] = \mathrm{plug}([\![C[-]]\!]\,, [\![M]\!])$$

Proof.

Tediously define terms with holes, plugging holes syntactically, carefully capturing some variables but not others. Then induction over semantics. □

Dream

It would be nice if tedious bits were...

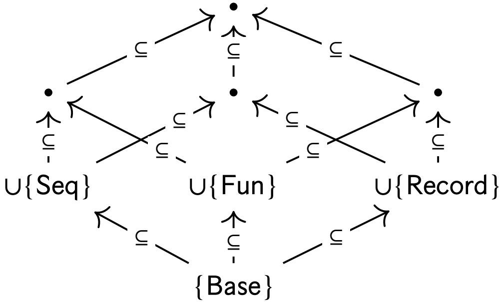
... free

Dream vs. Reality

It would be nice if tedious bits were…

… ~~free~~

… syntactically scaleable: additive syntactic work per new feature



SOAS: Second-Order Abstract Syntax

[Fiore, Plotkin, and Turi '99]

- CBN works smoothly.
- Robust to extensions:

  polymorphism [Fiore and Hamana'13]
  mechanisation [Crole'11, Allais et al.'18, Fiore and Szamoszvancev'22]
  substructurality [Fiore and Ranchod'25]

- Doesn't cover CBV.
  Technical reasons later:
  - Substitute **in**: values and terms
  - Substitute for variables: values only

  Slogan [cf. Levy's CBPV, '04]:

  values are 1^{st}-class

  but

  terms are 2^{nd}-class

Contribution

Modular Abstract Syntax Trees (MAST)

- SOAS $\overset{\text{generalise}}{\rightsquigarrow}$ 2[nd]-class sorts
  Using **skew** bicategories/monoidal categories, and:
  - Kleisli bicategories [Gambino, Fiore, Hyland, and Winskel'19]
  - Familial theory of SOAS [Fiore and Szamoszvancev'25]
- MAST tutorial
- Case-study: CBV semantics á la carte
  (128 substitution lemmata)

WIP

- Idris 2 implementation of computational fragment
  [cf. Fiore and Szamoszvancev'22]
- Replace skew monoidal structure and monoids with
  monoidal structure and actions
  [cf. Fiore and Turi'01]

Capstone: abstract syntax and substitution universality

Thm (representation)

abstract syntax with operators in $\mathbf{O}$ *and holes in* $\mathbf{H}$
amounts to
free substitution $\mathbf{O}$*-monoid over* $\mathbf{H}$*:*

$$\begin{array}{ccccc} & & \mathbf{H} & & \\ & & \downarrow ?\!-\![\mathsf{id}] & & \\ \$\mathbf{H}\otimes\$\mathbf{H} & \xrightarrow{-[-]} & \$\mathbf{H} & \xleftarrow[\mathsf{var}]{} & \mathbb{I} \\ & & [\![-]\!] \uparrow & & \\ & & \mathbf{O}(\$\mathbf{H}) & & \end{array}$$

Capstone: semantics

Key propaganda

compositional, binding-respecting denotational semantics
amounts to
substitution $\mathbf{O}$-monoid:

$$\mathbf{M} \otimes \mathbf{M} \xrightarrow{-[-]} \mathbf{M} \xleftarrow[\mathsf{var}]{} \mathbb{I}$$
$$[\![-]\!] \uparrow$$
$$\mathbf{OM}$$

The denotational semantics for terms with holes in $\mathbf{H}$ is the unique substitution $\mathbf{O}$-monoid homomorphism over $\mathbf{H}$:

$$\left(\$\mathbf{H}, -[-], \mathsf{var}, [\![-]\!], ?{-}[\mathsf{id}]\right) \xrightarrow{[\![-]\!]} (\mathbf{M}, -[-], \mathsf{var}, [\![-]\!], \mathsf{menv})$$
$$(\mathbf{H} \xrightarrow{\mathsf{menv}} \mathbf{M})$$

Meta-theory in one line

Lemma (substitution)

Syntactic substitution corresponds to semantic composition:

$$[\![M\,[\theta]]\!] = [\![M]\!] \circ [\![\theta]\!]$$

Lemma (compositionality)

Composite semantics is independent of component syntax:

$$[\![C[M]]\!] = \mathrm{plug}([\![C[-]]\!]\,, [\![M]\!])$$

Meta-theory in one line

Lemma (substitution)

Syntactic substitution corresponds to semantic composition:

substitution monoid homomorphism

$$[\![M\,[\theta]]\!] = [\![-[-]\,[M,\theta]]\!] \overset{\downarrow}{=} -[-]\left[[\![M]\!]\,,[\![\theta]\!]\right] \coloneqq [\![M]\!] \circ [\![\theta]\!]$$

Lemma (compositionality)

Composite semantics is independent of component syntax:

$$[\![C[M]]\!] = \mathrm{plug}([\![C[-]]\!]\,,[\![M]\!])$$

Meta-theory in one line

Lemma (substitution)

Syntactic substitution corresponds to semantic composition:

substitution monoid homomorphism

$$[\![M\,[\theta]]\!] = [\![-[-]\,[M,\theta]]\!] \overset{\downarrow}{=} -[-]\left[[\![M]\!]\,,[\![\theta]\!]\right] := [\![M]\!]\circ[\![\theta]\!]$$

Lemma (compositionality)

Composite semantics is independent of component syntax:

$$[\![C[M]]\!] = [\![(?m\,[M]) \ggg (m \mapsto C[-])]\!] = [\![?m\,[M]]\!] \ggg (m \mapsto [\![C[-]]\!])$$
$$=: \mathrm{plug}([\![C[-]]\!]\,,[\![M]\!])$$

↑ *$\ggg$ is homomorphic extension*

MAST taster: heterogeneous sorting

Sorting signature $\mathbf{R}$

- set sort

partitioned into

- bindable/1^{st}-class sorts
  $s \in \mathsf{Bind}$
- non-bindable/2^{st}-class sorts

Example

CBV sorting signature

- $\mathsf{sort} \coloneqq \{A, \mathsf{comp}\, A | A \in \mathsf{Type}\}$
- $\mathsf{Bind} \coloneqq \mathsf{Type}$

MAST taster: semantic domain for syntax and semantics

MAST provides $(\mathbf{R} = (\mathsf{sort}, \mathsf{Bind})$ sorting system$)$

- Contexts $\mathsf{sort}_{\vdash} \ni \Gamma ::= [x_1 : s_1, \ldots, x_n : s_n]$
- Renamings $\mathsf{sort}_{\vdash}(\Gamma, \Delta) \ni \Gamma \vdash \rho : \Delta$
- $\mathbf{R}$-structures: $\mathbf{PSh}(\mathsf{sort} \times \mathsf{sort}_{\vdash}) \ni P : \mathsf{sort} \times \mathsf{sort}_{\vdash}^{\mathsf{op}} \to \mathbf{Set}$

  $P_s\Gamma \ni p$: sort s element with variables in Γ
- Variables structure: $\mathbf{R\text{-}Struct} \ni \mathbb{I}_s\Gamma \coloneqq \{x | (x : s) \in \Gamma\}$
- substitution tensors: $\mathbf{R\text{-}Struct} \ni P \otimes Q, P \otimes_{\bullet} \left(\mathsf{var} \overset{\mathbb{I}}{\underset{A}{\downarrow}}\right)$

  $(P \otimes Q)_s\Gamma \ni [p, \theta]_\Delta$: P-element: $p \in P_s\Gamma$

  Q-closure : $\theta \in \prod_{(y:r)\in\Delta} Q_r\Gamma$

  identifying, e.g.:

$$[p[\mathsf{weaken}], \theta]_{\Delta_1 + \Delta_2} = [p, \theta \circ \rho]_{\Delta_1} \qquad \left[p[x', x'' \mapsto x]\right]_{x\in\Delta}, \theta]_\Delta = [p, \theta + \theta]_{\Delta + \Delta}$$

Allow us to define:

MAST taster: semantic domain for syntax and semantics

MAST provides ($\mathbf{R} = (\mathsf{sort}, \mathsf{Bind})$ sorting system)

- Contexts $\qquad \mathsf{sort}_\vdash \ni \Gamma ::= [x_1 : s_1, \ldots, x_n : s_n]$
- Renamings $\qquad \mathsf{sort}_\vdash(\Gamma, \Delta) \ni \Gamma \vdash \rho : \Delta$
- $\mathbf{R}$-structures: $\qquad \mathbf{PSh}(\mathsf{sort} \times \mathsf{sort}_\vdash) \ni P : \mathsf{sort} \times \mathsf{sort}_\vdash^{\mathsf{op}} \to \mathbf{Set}$

  $P_s\Gamma \ni p$: sort s element with variables in Γ
- Variables structure: $\qquad \mathbf{R\text{-}Struct} \ni \mathbb{I}_s\Gamma := \{x | (x : s) \in \Gamma\}$
- substitution tensors: $\qquad \mathbf{R\text{-}Struct} \ni P \otimes Q, P \otimes_{\bullet} \left(\mathsf{var} \begin{smallmatrix} \mathbb{I} \\ \downarrow \\ A \end{smallmatrix}\right)$

  $(P \otimes Q)_s\Gamma \ni [p, \theta]_\Delta$: $\quad P$-element: $p \in P_s\Gamma$

  Q-closure : $\theta \in \prod_{(y:r)\in\Delta} Q_r\Gamma$

  identifying, e.g.:

$$[p[\mathsf{weaken}], \theta]_{\Delta_1 + \Delta_2} = [p, \theta \circ \rho]_{\Delta_1} \qquad \left[p[x', x'' \mapsto x]_{x\in\Delta}, \theta\right]_\Delta = [p, \theta + \theta]_{\Delta + \Delta}$$

Allow us to define:

Scope-change as tensorial strength

$$\mathsf{str}^{\mathbf{O}} : (\mathbf{O}P) \otimes_{\bullet} \left(\mathsf{var} \begin{smallmatrix} \mathbb{I} \\ \downarrow \\ A \end{smallmatrix}\right) \to \mathbf{O}\left(P \otimes_{\bullet} \left(\mathsf{var} \begin{smallmatrix} \mathbb{I} \\ \downarrow \\ A \end{smallmatrix}\right)\right)$$

MAST taster: semantic domain for syntax and semantics

MAST provides ($\mathbf{R} = (\mathsf{sort}, \mathsf{Bind})$ sorting system)

- Contexts $\quad \mathsf{sort}_\vdash \ni \Gamma ::= [x_1 : s_1, \ldots, x_n : s_n]$
- Renamings $\quad \mathsf{sort}_\vdash(\Gamma, \Delta) \ni \Gamma \vdash \rho : \Delta$
- $\mathbf{R}$-structures: $\quad \mathbf{PSh}(\mathsf{sort} \times \mathsf{sort}_\vdash) \ni P : \mathsf{sort} \times \mathsf{sort}_\vdash^{\mathsf{op}} \to \mathbf{Set}$

  $P_s\Gamma \ni p$: sort s element with variables in Γ
- Variables structure: $\quad \mathbf{R\text{-}Struct} \ni \mathbb{I}_s\Gamma \coloneqq \{x | (x : s) \in \Gamma\}$
- substitution tensors: $\quad \mathbf{R\text{-}Struct} \ni P \otimes Q, P \otimes_{\bullet} \left(\mathsf{var} \; \overset{\mathbb{I}}{\underset{A}{\downarrow}}\right)$

  $(P \otimes Q)_s\Gamma \ni [p, \theta]_\Delta$: $\quad$ P-element: $p \in P_s\Gamma$

  Q-closure : $\theta \in \prod_{(y:r)\in\Delta} Q_r\Gamma$

  identifying, e.g.:

$$[p[\mathsf{weaken}], \theta]_{\Delta_1 + \Delta_2} = [p, \theta \circ \rho]_{\Delta_1} \qquad \left[p[x', x'' \mapsto x]_{x\in\Delta}, \theta\right]_\Delta = [p, \theta + \theta]_{\Delta + \Delta}$$

Allow us to define:

Scope-change as tensorial strength

$$\mathsf{str}^{\mathbf{O}} : (\mathbf{O}P) \otimes_{\bullet} \left(\mathsf{var} \; \overset{\mathbb{I}}{\underset{A}{\downarrow}}\right) \to \mathbf{O}\left(P \otimes_{\bullet} \left(\mathsf{var} \; \overset{\mathbb{I}}{\underset{A}{\downarrow}}\right)\right)$$

Substitution monoids

$$\mathbf{M} \otimes \mathbf{M} \xrightarrow{-[-]} \mathbf{M} \xleftarrow{\mathsf{var}} \mathbb{I}$$

MAST taster: semantic domain for syntax

Signature functors

$$\mathbf{O} \circlearrowright \textbf{R-Struct}$$

Scope-change as tensorial strength

$$\mathsf{str}^{\mathbf{O}} : (\mathbf{O}P)\otimes_{\bullet}\left(\mathsf{var}\ \overset{\mathbb{I}}{\underset{A}{\downarrow}}\right) \to \mathbf{O}\left(P\otimes_{\bullet}\left(\mathsf{var}\ \overset{\mathbb{I}}{\underset{A}{\downarrow}}\right)\right)$$

Example

Sequential fragment signature functor:

$$(\mathsf{Seq}\ X)_{\mathsf{comp}\ B}\Gamma \coloneqq \coprod_{A\in\mathsf{Type}} \begin{pmatrix}(\mathbf{let}\ x : A = \_\ \mathbf{in}\ \_) : \\ (X_{\mathsf{comp}\ A}\Gamma \times X_{\mathsf{comp}\ B}(\Gamma, x : A))\end{pmatrix}$$

$$(\mathsf{Seq}\ X)_A\Gamma \coloneqq \emptyset$$

$$\mathsf{str}^{\mathsf{Seq}}\left[\mathbf{let}\ x : A = (p \in P_{\mathsf{comp}\ A}\Delta)\ \mathbf{in}\ (q \in P_{\mathsf{comp}\ B}(\Delta, x : A)), \theta\right]_{\Delta}$$
$$\coloneqq \left(\mathbf{let}\ x : A = [p,\theta]_{\Delta}\ \mathbf{in}\ [q, (\theta, x : \mathsf{var}\ x)]_{\Delta, x : A}\right)$$

Takeaway

Modular spec. for binding, renaming, and substitution structure

$\textsc{mast}$ taster: semantic domain for syntax

Abstract syntax: inductive representation

Every initial algebra:

$$\mathbb{S}^{\mathbf{O}}\mathbf{H} \coloneqq \mu X.(\mathbf{O}X) \amalg \mathbb{I} \amalg \mathbf{H} \otimes X$$

Supports standard definitions:

$$\begin{array}{ccccc}
 & & \mathbf{H} & & \\
 & & \downarrow ?\!-\![-] & & \\
\mathbb{S}\mathbf{H} \otimes \mathbb{S}\mathbf{H} & \xrightarrow{-[-]} & \mathbb{S}\mathbf{H} & \xleftarrow[\text{var}]{} & \mathbb{I} \\
 & & [\![-]\!] \uparrow & & \\
 & & \mathbf{O}(\mathbb{S}\mathbf{H}) & &
\end{array}$$

Independently of concrete representation, e.g.,:

- De-Bruijn
- Nominal
- Locally nameless
- Co-de Bruijn
- Graphical

Example

$\mathbf{M} = \left(\mathcal{C}, \mathrm{T}, \mathrm{return}, \gg\!=, [\![-]\!]\right)$:

- $\mathcal{C}$: Cartesian category with chosen finite products
- $(\mathrm{T}, \mathrm{return}, \gg\!=)$ strong monad over $\mathcal{C}$
- $[\![-]\!]$: Type $\to \mathcal{C}$ type interpretation

induces:

- A CBV-structure: $\text{CBV-}\mathbf{Struct} \ni \mathbf{M}_s\Gamma := \mathcal{C}([\![\Gamma]\!], [\![s]\!])$
- Standard interpretation of contexts, computations, renaming:

$$\mathcal{C} \ni [\![\Gamma]\!] := \prod_{(x:A)\in\Gamma} [\![A]\!] \qquad \mathcal{C} \ni [\![\mathsf{comp}\ A]\!] := \mathrm{T}\,[\![A]\!]$$

$$[\![\rho]\!] : [\![\Gamma]\!] \xrightarrow{\left(\pi_{x[\rho]} : [\![\Gamma]\!] \to [\![A]\!]\right)_{(x:A)\in\Delta}} \prod_{(x:A)\in\Delta} [\![A]\!] = [\![\Delta]\!]$$

MAST taster: common structure for substitution

Syntactic substitution monoid

$$\$^{\mathbf{O}}\mathbf{H} \otimes \$^{\mathbf{O}}\mathbf{H} \xrightarrow{-[-]} \$^{\mathbf{O}}\mathbf{H} \xleftarrow{\mathsf{var}} \mathbb{I}$$

Monoid axioms amounts to syntactic substitution lemma

Example

Semantic substitution monoid: $\mathbf{M} \otimes \mathbf{M} \xrightarrow{-[-]} \mathbf{M} \xleftarrow{\mathsf{var}} \mathbb{I}$

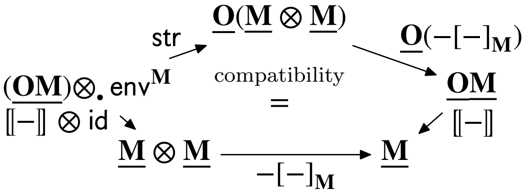
- Substitution via composition:

$$\left(\llbracket\Delta\rrbracket \xrightarrow{f} \llbracket s\rrbracket\right)\left[\llbracket\Gamma\rrbracket \xrightarrow{\theta} \llbracket\Delta\rrbracket\right] : \llbracket\Gamma\rrbracket \xrightarrow{\theta} \llbracket\Delta\rrbracket \xrightarrow{f} \llbracket s\rrbracket$$

- Variables: (1st-class sorts only) $\mathsf{var} : \Big((x : A) \in \Gamma \mapsto \Big(\llbracket\Gamma\rrbracket \xrightarrow{\pi_x} \llbracket A\rrbracket\Big)\Big)$

Substitution-compatible algebra

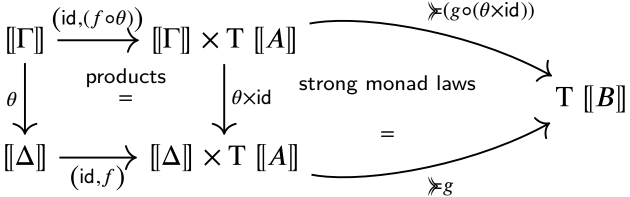
$[\![-]\!] : \mathbf{OM} \to \mathbf{M}$:



Example

$$\left[\!\!\left[\begin{array}{l}\mathbf{let}\ x : A = ([\![\Gamma]\!] \xrightarrow{f} \mathrm{T}\,[\![A]\!]) \\ \mathbf{in}\ ([\![\Gamma]\!] \times [\![A]\!] \xrightarrow{g} \mathrm{T}\,[\![B]\!])\end{array}\right]\!\!\right] : [\![\Gamma]\!] \xrightarrow{(\mathsf{id}, f)} [\![\Gamma]\!] \times \mathrm{T}\,[\![A]\!] \xrightarrow{\succcurlyeq g} \mathrm{T}\,[\![B]\!]$$

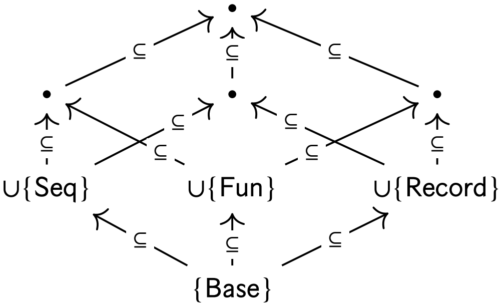
Compatibility:



MAST: modularity and scalability

Substitution **O**-monoid

Substitution monoid with compatible **O**-algebra structure



Core contribution

classical theory (SOAS)

$\mathbf{PSh}\,(\mathsf{sort} \times \mathsf{sort}_{\vdash}), \otimes$
monoidal product

generalise $\rightsquigarrow$

this work (MAST)

$\mathbf{PSh}\,(\mathsf{sort} \times \text{-}\mathsf{Bind}_{\vdash})\otimes$
right-unital associative
skew monoidal product

Skew tensor products

$$(P \otimes Q) \otimes L \cong P \otimes (Q \otimes L) \qquad \text{(associative)}$$

$$P \otimes \mathbb{I} \cong P \qquad \text{(right-unital)}$$

$$\mathbb{I} \otimes Q \xrightarrow{\mathbf{r}'} Q \qquad \text{(non-invertible!)}$$

What breaks the unitor?

Substitution tensor

$$(P \otimes Q)_s\Gamma \coloneqq \int^{\Delta} P_s\Gamma \times \prod_{(y:r)\in\Delta} Q_r\Gamma$$

for $s \notin \mathsf{Bind}$, $Q = \mathbb{1}$, $\mathbb{I}_s\Delta = \emptyset$:

$$(\mathbb{I} \otimes Q)_s\Gamma \coloneqq \int^{\Delta} \overbrace{\emptyset}^{\mathbb{I}_s\Delta} \times \prod_{(y:r)\in\Delta} Q_r\Gamma = \int^{\Delta} \emptyset = \emptyset \neq \mathbb{1} = Q_s\Gamma$$

Want more?

In the paper:

- All the details
- A CBV case-study (128 substitution lemmata)

In the future:

- Idris 2 implementation of computational fragment
  [cf. Fiore and Szamoszvancev'22]
- Replace skew monoidal structure and monoids with
  monoidal structure and actions

Contribution

Modular Abstract Syntax Trees (MAST)

- SOAS $\overset{\text{generalise}}{\rightsquigarrow}$ 2^{nd}-class sorts
  Using **skew** bicategories/monoidal categories, and:
  - Kleisli bicategories [Gambino, Fiore, Hyland, and Winskel'19]
  - Familial theory of SOAS [Fiore and Szamoszvancev'25]
- MAST tutorial
- Case-study: CBV semantics á la carte
  (128 substitution lemmata)

WIP

- Idris 2 implementation of computational fragment
  [cf. Fiore and Szamoszvancev'22]
- Replace skew monoidal structure and monoids with
  monoidal structure and actions
  [cf. Fiore and Turi'01]