Friedrich-Alexander-Universität
Technische Fakultät

Towards a Higher-Order Mathematical Operational Semantics

Sergey Goncharov

Friedrich-Alexander Universität Erlangen-Nürnberg

Topos Institute, 16 February 2023

Intro

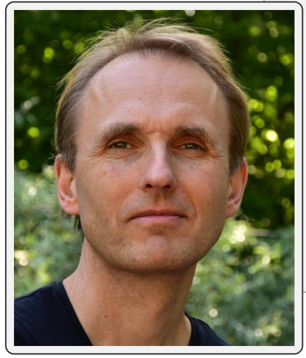
- Present talk is based on this year's POPL paper

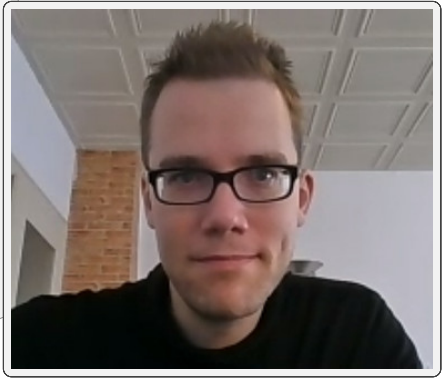
Towards a Higher-Order Mathematical Operational Semantics*

SERGEY GONCHAROV, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
STEFAN MILIUS[†], Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
LUTZ SCHRÖDER[‡], Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
STELIOS TSAMPAS[§], Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
HENNING URBAT[¶], Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Compositionality proofs in higher-order languages are notoriously involved, and general semantic frameworks guaranteeing compositionality are hard to come by. In particular, Turi and Plotkin's bialgebraic abstract GSOS framework, which has been successfully applied to obtain off-the-shelf compositionality results for first-order languages, so far does not apply to higher-order languages. In the present work, we develop a theory of abstract GSOS specifications for higher-order languages, in effect transferring the core principles of Turi and Plotkin's framework to a higher-order setting. In our theory, the operational semantics of higher-order languages is represented by certain dinatural transformations that we term *pointed higher-order GSOS laws*. We give a general compositionality result that applies to all systems specified in this way and discuss how compositionality of the SKI calculus and the λ-calculus w.r.t. a strong variant of Abramsky's applicative bisimilarity are obtained as instances.

CCS Concepts: • **Theory of computation** → **Categorical semantics**; **Operational semantics**.





Intro

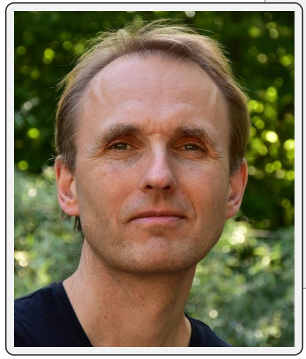
- Present talk is based on this year's POPL paper

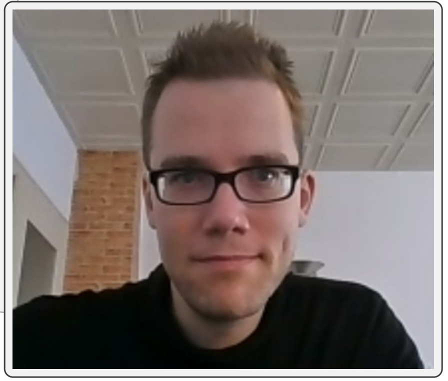
Towards a Higher-Order Mathematical Operational Semantics*

SERGEY GONCHAROV, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
STEFAN MILIUS[†], Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
LUTZ SCHRÖDER[‡], Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
STELIOS TSAMPAS[§], Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
HENNING URBAT[¶], Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Compositionality proofs in higher-order languages are notoriously involved, and general semantic frameworks guaranteeing compositionality are hard to come by. In particular, Turi and Plotkin's bialgebraic abstract GSOS framework, which has been successfully applied to obtain off-the-shelf compositionality results for first-order languages, so far does not apply to higher-order languages. In the present work, we develop a theory of abstract GSOS specifications for higher-order languages, in effect transferring the core principles of Turi and Plotkin's framework to a higher-order setting. In our theory, the operational semantics of higher-order languages is represented by certain dinatural transformations that we term *pointed higher-order GSOS laws*. We give a general compositionality result that applies to all systems specified in this way and discuss how compositionality of the SKI calculus and the λ-calculus w.r.t. a strong variant of Abramsky's applicative bisimilarity are obtained as instances.

CCS Concepts: • **Theory of computation** → **Categorical semantics**; **Operational semantics**.





- plus some perks!

Context

- Turi and Plotkin's abstract GSOS is a definite framework for relating operational and denotational semantics
- As any very general tool, it has numerous limitations
- One such limitation: no support for higher-order behaviour
- By contrast, reasoning about higher-order languages is complicated and largely boilerplate. Tools involved: applicative bisimilarity, Howe's method, environmental bisimilarity, logical relations
- We make first steps in reorganizing higher-order semantics, building on abstract GSOS, in particular, develop a (strong) colagebraic applicative bisimilarity

Outline

First-Order GSOS

Higher-Order GSOS

Abstract Congruence for Higher-Order GSOS

Higher-Order Bialgebras

Applicative Bisimilarity

Call-by-Value

Conclusions

Abstract GSOS

Turi and Plotkin's abstraction of GSOS[1]:

- A signature endofunctor $\Sigma \colon \mathbf{C} \to \mathbf{C}$
- A behaviour endofunctor $B \colon \mathbf{C} \to \mathbf{C}$
- A GSOS law – natural transformation $\rho_X \colon \Sigma(X \times BX) \to B\Sigma^{\star}X$

Typically: Σ is a polynomial functor, representing an algebraic signature, $BX = \mathcal{P}(L \times X)$, ρ is induced by operational semantic rules, "distributing syntax over semantics", e.g.

$$\frac{p \xrightarrow{a} p'}{p \mid q \xrightarrow{a} p' \mid q}$$

[1] Turi and Plotkin, "Towards a Mathematical Operational Semantics", 1997.

Abstract GSOS

Turi and Plotkin's abstraction of GSOS[1]:

- A signature endofunctor $\Sigma\colon \mathbf{C} \to \mathbf{C}$
- A behaviour endofunctor $B\colon \mathbf{C} \to \mathbf{C}$
- A GSOS law – natural transformation $\rho_X\colon \Sigma(X \times BX) \to B\Sigma^\star X$

behaviour from B

Typically: Σ is a polynomial functor, representing an algebraic signature, $BX = \mathcal{P}(L \times X)$, ρ is induced by operational semantic rules, "distributing syntax over semantics", e.g.

operation from Σ

$$\frac{p \xrightarrow{a} p'}{p \mid q \xrightarrow{a} p' \mid q}$$

[1] Turi and Plotkin, "Towards a Mathematical Operational Semantics", 1997.

Abstract GSOS

Turi and Plotkin's abstraction of GSOS[1]:

- A signature endofunctor $\Sigma\colon \mathbf{C} \to \mathbf{C}$
- A behaviour endofunctor $B\colon \mathbf{C} \to \mathbf{C}$
- A GSOS law – natural transformation $\rho_X\colon \Sigma(X \times BX) \to B\Sigma^\star X$

Typically: Σ is a polynomial functor, representing an algebraic signature, $BX = \mathcal{P}(L \times X)$, ρ is induced by operational semantic rules, "distributing syntax over semantics", e.g.

$$\frac{p \xrightarrow{a} p'}{p \mid q \xrightarrow{a} p' \mid q}$$

[1] Turi and Plotkin, "Towards a Mathematical Operational Semantics", 1997.

Abstract GSOS: Bialgebras

- A ρ-bialgebra interprets operations by an algebra $a\colon \Sigma X \to X$ and provides a behaviour via a coalgebra $c\colon X \to BX$ such that

$$\begin{tikzcd} \Sigma X \arrow[r, "a"] \arrow[d, "\Sigma\langle \mathsf{id}, c\rangle"'] & X \arrow[r, "c"] & BX \\ \Sigma(X \times BX) \arrow[rr, "\rho_X"] & & B\Sigma^\star X \arrow[u, "B\widehat{a}"'] \end{tikzcd}$$

  where $\widehat{a}\colon \Sigma^\star X \to X$ is the inductive extension of a
- Operational model: initial bialgebra $\Sigma\mu\Sigma \xrightarrow{\iota} \mu\Sigma \xrightarrow{\gamma} B\mu\Sigma$
- Denotational model: final bialgebra $\Sigma\nu B \xrightarrow{\alpha} \nu B \xrightarrow{\tau} B\nu B$
- Abstract behaviour: unique bialgebra morphism $[\![-]\!]_\rho\colon (\mu\Sigma, \iota, \gamma) \to (\nu B, \alpha, \tau)$
- Full abstraction: p and q are behaviourally equal iff $[\![p]\!]_\rho = [\![q]\!]_\rho$

Higher Order

Q: Why the theory of program equivalence of higher-order languages is so different (and difficult!)?

Higher Order

Q: Why the theory of program equivalence of higher-order languages is so different (and difficult!)?

A: Because it is a mixture of many things:

- higher-order languages typically involve variable binding and capture-avoiding substitution. That adds a bunch of technical issues, but does not itself make things higher-order (e.g. π-calculus)
- behavioural equivalence is weak from the outset (contrasting process algebra, which uses strong bisimulation as a stepping stone)
- denotational models (e.g. domains) do not come from behaviours

$\Rightarrow$ full abstraction tends to fail

(Extended) Combinatory Logic

- SKI language: S for $\lambda p.\lambda q.\lambda r.\,(pr)(qr)$, K for $\lambda p.\,q, p$, I for $\lambda p.\,p$
  - plus S', S'' and K' for partially reduced terms
- Operational semantics:

$$\frac{}{S \xrightarrow{t} S'(t)} \qquad \frac{}{S'(p) \xrightarrow{t} S''(p,t)} \qquad \frac{}{S''(p,q) \xrightarrow{t} (p\ t)\ (q\ t)}$$

$$\frac{}{K \xrightarrow{t} K'(t)} \qquad \frac{}{K'(p) \xrightarrow{t} p} \qquad \frac{}{I \xrightarrow{t} t} \qquad \frac{p \to p'}{p\ q \to p'\ q} \qquad \frac{p \xrightarrow{q} p'}{p\ q \to p'}$$

- This is not GSOS
- But it makes perfect sense, e,g.: $Spqr \to S'(p)qr \to S''(p,q)r \to (pr)(qr)$

Congruence for SKI

- With $\Sigma = \{S, K, I, S', S'', K'\}$, $\mu\Sigma$ is the set of SKI-terms
- Define strong applicative bisimilarity $\sim$ on $\mu\Sigma$: $\sim$ is the greatest relation $R \subseteq \mu\Sigma \times \mu\Sigma$ such that whenever pRq, then
  - either $p \to p'$, $q \to q'$ and $p'Rq'$,
  - or for every $t \in \mu\Sigma$, $p \xrightarrow{t} p'$, $q \xrightarrow{t} q'$ and $p'Rq'$

Congruence for SKI

- With $\Sigma = \{S, K, I, S', S'', K'\}$, $\mu\Sigma$ is the set of SKI-terms
- Define strong applicative bisimilarity $\sim$ on $\mu\Sigma$: $\sim$ is the greatest relation $R \subseteq \mu\Sigma \times \mu\Sigma$ such that whenever pRq, then
  - either $p \to p'$, $q \to q'$ and $p'Rq'$,
  - or for every $t \in \mu\Sigma$, $p \xrightarrow{t} p'$, $q \xrightarrow{t} q'$ and $p'Rq'$

Proposition: $\sim$ is a Σ-congruence
Proof Idea: For any $R \subseteq \mu\Sigma \times \mu\Sigma$, define

$$\widehat{R} = \{(C[s], C[t]) \in \mu\Sigma \times \mu\Sigma \mid C \text{ a (linear) context, } sRt\}$$

Show that $\widehat{\sim}^{\star} \subseteq \sim$. Essentially, this is up-to-congruence

Congruence for SKI

- With $\Sigma = \{S, K, I, S', S'', K'\}$, $\mu\Sigma$ is the set of SKI-terms
- Define strong applicative bisimilarity $\sim$ on $\mu\Sigma$: $\sim$ is the greatest relation $R \subseteq \mu\Sigma \times \mu\Sigma$ such that whenever pRq, then
  - either $p \to p'$, $q \to q'$ and $p'Rq'$,
  - or for every $t \in \mu\Sigma$, $p \xrightarrow{t} p'$, $q \xrightarrow{t} q'$ and $p'Rq'$

Proposition: $\sim$ is a Σ-congruence
Proof Idea: For any $R \subseteq \mu\Sigma \times \mu\Sigma$, define

$$\widehat{R} = \{(C[s], C[t]) \in \mu\Sigma \times \mu\Sigma \mid C \text{ a (linear) context, } sRt\}$$

Show that $\widehat{\sim}^{\star} \subseteq \sim$. Essentially, this is up-to-congruence

Q: Can we apply same trick to (standard) weak bisimulation?
A: Yes! But, we need “up-to” Howe’s closure

Dinaturality

A dinatural transformation from $F \colon \mathbf{C}^{\mathsf{op}} \times \mathbf{C} \to \mathbf{D}$ to $G \colon \mathbf{C}^{\mathsf{op}} \times \mathbf{C} \to \mathbf{D}$ is a family $(\sigma_X \colon F(X,X) \to G(X,X))_{X \in \mathbf{C}}$, such that

$$
\begin{tikzcd}
& F(X,X) \arrow[r, "\sigma_X"] & G(X,X) \arrow[dr, "G(X,f)"] & \\
F(Y,X) \arrow[ur, "F(f,X)"] \arrow[dr, "F(X,f)"'] & & & G(X,Y) \\
& F(Y,Y) \arrow[r, "\sigma_Y"] & G(Y,Y) \arrow[ur, "G(f,Y)"'] &
\end{tikzcd}
$$

for every $f \colon X \to Y$

Example: $\mathsf{apply} \colon X^Y \times Y \to X$

Defining HO(ly)-GSOS

Definition: A higher-order GSOS law of Σ over B is a family of morphisms

$$\rho_{X,Y}\colon \Sigma(X \times B(X, Y)) \to B(X, \Sigma^\star(X + Y)),$$

dinatural in X and natural in Y

Example: For combinatory logic: Σ is obvious, and $B(X, Y) = Y^X + Y$, and ρ is induced by the rules

In fact, for polynomial Σ and $B(X, Y) = Y^X + Y$, we have a complete syntactic characterization of higher-order GSOS (via Yoneda lemma)

Example: For λ-calculus: $\mathbf{C} = [\mathbb{F}, \mathbf{Set}]$, $\Sigma = V + \delta X + X^2$, and ρ must be V-pointed[2]

[2]Fiore, Plotkin, and Turi, "Abstract Syntax and Variable Binding", 1999.

Operational Model

- An operational model can be readily defined:

$$
\begin{tikzcd}
\Sigma\mu\Sigma \arrow[rr, "\iota"] \arrow[d, "\Sigma\langle \mathrm{id}, \iota^{\clubsuit}\rangle"'] & & \mu\Sigma \arrow[d, "\langle \mathrm{id}, \iota^{\clubsuit}\rangle"] \\
\Sigma(\mu\Sigma \times B(\mu\Sigma, \mu\Sigma)) \arrow[r, "\langle \iota\cdot\Sigma\,\mathrm{fst},\, \rho_{\mu\Sigma,\mu\Sigma}\rangle"] & \mu\Sigma \times B(\mu\Sigma, \Sigma^{\star}(\mu\Sigma + \mu\Sigma)) \arrow[r, "\mathrm{id}\times B(\mathrm{id},\mu\cdot\Sigma^{\star}\nabla)"] & \mu\Sigma \times B(\mu\Sigma, \mu\Sigma)
\end{tikzcd}
$$

where $\mu\colon \Sigma^{\star}\Sigma^{\star} \to \Sigma^{\star}$ is the obvious flattening

- The operational equivalence is the kernel of the map

$$\mathrm{coit}(\iota^{\clubsuit})\colon \mu\Sigma \to \nu\gamma.\, B(\mu\Sigma, \gamma)$$

Abstract Congruence for H/O GSOS

Theorem: Given that

1. $\mathbf{C}$ is regular (roughly, $\mathbf{C}$ admits a good notion of image factorization)
2. Σ preserves reflexive coequalizers (in particular, if Σ is finitary)
3. B preserves monomorphisms in both arguments (it sends epis on the first argument to monos)

the kernel pair of the final coalgebra map

$$\mathsf{coit}(\iota^{\clubsuit})\colon \mu\Sigma \to \nu\gamma.\, B(\mu\Sigma, \gamma)$$

is a congruence

Higher-Order Bialgebras

A ρ-bialgebra is a triple $(A, a\colon \Sigma A \to A, c\colon A \to B(A,A))$ such that the diagram

$$\begin{tikzcd}
\Sigma A \arrow[r, "a"] \arrow[d, "\Sigma\langle \mathrm{id}, c\rangle"'] & A \arrow[r, "c"] & B(A,A) \\
\Sigma(A \times B(A,A)) \arrow[rr, "\rho"] & & B(A, \Sigma^\star(A + A)) \arrow[u, "B(\mathrm{id}, \nabla^\sharp)"']
\end{tikzcd}$$

commutes

Higher-Order Bialgebras Morphisms

A ρ-bialgeba morphism from (A, a, c) to (A', a', c') is a Σ-algebra morphism $f\colon A \to A'$, such that the diagram

$$\begin{tikzcd}
A \arrow[rr, "c"] \arrow[d, "f"'] & & B(A,A) \arrow[d, "B(\mathrm{id},f)"] \\
A' \arrow[r, "c'"] & B(A',A') \arrow[r, "B(f,\mathrm{id})"] & B(A,A')
\end{tikzcd}$$

commutes

Higher-Order Bialgebras, Cont'd

- Operational model again yields an initial ρ-bialgebra

$$\Sigma\mu\Sigma \xrightarrow{\iota} \mu\Sigma \xrightarrow{\iota^{\clubsuit}} B(\mu\Sigma, \mu\Sigma)$$

- The behavioural quotient also extends to a ρ-bialgebra

$$\Sigma\mu\Sigma_{\sim} \xrightarrow{\iota_{\sim}} \mu\Sigma_{\sim} \to B(\mu\Sigma_{\sim}, \mu\Sigma_{\sim})$$

- The quotienting map $\mu\Sigma \to \mu\Sigma_{\sim}$ is a bialgebra morphism

Higher-Order Bialgebras, Cont'd

- Operational model again yields an initial ρ-bialgebra

$$\Sigma\mu\Sigma \xrightarrow{\iota} \mu\Sigma \xrightarrow{\iota^{\clubsuit}} B(\mu\Sigma, \mu\Sigma)$$

- The behavioural quotient also extends to a ρ-bialgebra

$$\Sigma\mu\Sigma_{\sim} \xrightarrow{\iota_{\sim}} \mu\Sigma_{\sim} \to B(\mu\Sigma_{\sim}, \mu\Sigma_{\sim})$$

- The quotienting map $\mu\Sigma \to \mu\Sigma_{\sim}$ is a bialgebra morphism
- However (!) a final bialgebra does not generally exist

Lambda-Calculus: Way of Combinators

- Let $\Lambda(n)$ be λ-terms with free variables from $\{1, \ldots, n\}$ modulo α-equivalence
- $\Sigma X = \coprod_{n \in \mathbb{N}} \Lambda(n+1) \times X^n + X^2$, so

$$(f, t_1, \ldots, t_n) \quad \text{represents} \quad \lambda(n+1).\, f[t_1/1, \ldots, t_n/n]$$

- $B(X, Y) = Y + Y^X$
- For every $f \in \Lambda(n+1)$ let $\lceil f \rceil \in \Sigma^\star(n+2)$ be obtained from f by recursively replacing topmost $\lambda x.\, t$ with $t[(n+2)/x] \in \Lambda(n+2)$

Examples: $S = \lambda yz.\, (1z)(yz) \in \Lambda(1)$, $S' = \lceil S \rceil = \lambda z.\, (1z)(2z) \in \Lambda(2)$,
$S'' = \lceil S' \rceil = (13)(23) \in \Sigma^\star(3)$

Way of Combinators, Cont'd

- The rules

$$\frac{}{f(x_1,\ldots,x_n) \xrightarrow{t} \lceil f \rceil [x_1/1\ldots,x_n/n,t/(n+1)]} \quad (f \in \Lambda(n+1))$$

$$\frac{p \to p'}{pq \to p'q} \qquad\qquad \frac{p \xrightarrow{q} p'}{pq \to p'}$$

  then mimic the standard call-by-name semantics of untyped λ-calculus

- Hence, we obtain a congruence result for the lazy λ-calculus circumventing presheave semantics!

Call-by-Value

- Call-by-value SKI: rules for combinators like before, plus

$$\frac{p \to p'}{pq \to p'q} \qquad \frac{p \xrightarrow{t} p' \quad q \to q'}{pq \to pq'} \qquad \frac{p \xrightarrow{q} p' \quad q \xrightarrow{t} q'}{pq \to p'}$$

- **Problem:** operational model must be $\mu\Sigma \to B(\mu\Sigma^{v}, \mu\Sigma)$ where $\mu\Sigma^{v} \hookrightarrow \mu\Sigma$ is a subobject of values, i.e. terms in normal form
- **Solution:** Two sorted sets!
  - The entire framework runs in $\mathbf{Set}^2 = [2, \mathbf{Set}] \cong \mathbf{Set}/2$, which provides a crisp separation between values and non-values
  - **Behaviour:** $B_v(X, Y) = (Y_v + Y_{\bar{v}})^{X_v}$ (value part), $B_{\bar{v}}(X, Y) = Y_v + Y_{\bar{v}}$ (non-value part)
  - **Signature:** $\Sigma_v(X) =$ 'combinators over $X_v + X_{\bar{v}}$', $\Sigma_{\bar{v}}X = (X_v + X_{\bar{v}})^2$

Further Work

Further work program is extensive:

- Modelling weak applicative bisimulation (arXive draft *“Weak Similarity in Higher-Order Mathematical Operational Semantics”* is comming next days)
  - Metric, quantialic, fibrational generalizations

    $\Rightarrow$ other variants of the framework needed, to bypass regularity
- Modelling other kinds of bisimilarity, e.g. environmental bisimilarity
- Modelling typed languages
- Modelling effectful languages
- …

1. Thank You for Your Attention!

Bialgebras Form a Category

If $f\colon A \to A'$ and $g\colon A' \to A''$ are ρ-bialgebra morphisms then so is the composition $g \cdot f$, for the diagram

$$\begin{tikzcd}
A \arrow[rrrr, "c"] \arrow[d, "f"'] & & & & B(A,A) \arrow[d, "{B(A,f)}"] \\
A' \arrow[rrr, "c'"] \arrow[d, "g"'] & & & B(A',A') \arrow[r, "{B(f,A')}"] \arrow[d, "{B(A',g)}"'] & B(A,A') \arrow[d, "{B(A,g)}"] \\
A'' \arrow[r, "c''"] & B(A'',A'') \arrow[rr, "{B(g,A'')}"] & & B(A',A'') \arrow[r, "{B(f,A'')}"] & B(A,A'')
\end{tikzcd}$$

obviously commutes.

Lambda-Calculus

- Operational semantics rules

$$\frac{s \to s'}{s\ t \to s'\ t} \qquad\qquad \frac{}{(\lambda x.s)\ t \to s[t/x]}$$

- $\mathbf{C} = \mathbf{Set}^{\mathbb{F}}$, where $\mathbb{F}$ is the category of finite cardinals

$$\Sigma\colon \mathbf{C} \to \mathbf{C}, \qquad \Sigma X = V + \delta X + X \times X,$$

$$B\colon \mathbf{C}^{\mathsf{op}} \times \mathbf{C} \to \mathbf{C}, \qquad B(X, Y) = \langle\!\langle X, Y \rangle\!\rangle \times (Y + Y^X + 1)$$

where Y^X is exponent in $\mathbf{Set}^{\mathbb{F}}$, V is the presheaf of variables $\mathbf{Set}^{\mathbb{F}}(n) = n$, $(\delta X)(n) = X(n+1)$, $\langle\!\langle X, Y \rangle\!\rangle(n) = \mathbf{Set}^{\mathbb{F}}(X^n, Y)$

- $\mu\Sigma$ is the presheaf $\Lambda \in \mathbf{Set}^{\mathbb{F}}$ of λ-terms over n free variables
- H/O GSOS law is pointed: $\rho_{X,Y}\colon \Sigma(jX \times B(jX, Y)) \to B(jX, \Sigma^{\star}(jX + Y))$[3]

[3]Fiore, Plotkin, and Turi, “Abstract Syntax and Variable Binding”, 1999.

References I

- Fiore, Marcelo P., Gordon D. Plotkin, and Daniele Turi. "Abstract Syntax and Variable Binding". In: *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*. IEEE Computer Society, 1999, pp. 193–202. URL: `https://doi.org/10.1109/LICS.1999.782615`.
- Turi, D. and G. Plotkin. "Towards a Mathematical Operational Semantics". In: *Logic in Computer Science*. IEEE. 1997, pp. 280–291.