Single-Variable Substitution for Substructural Theories

Sanjiv Ranchod

(joint work with Marcelo Fiore)

University of Cambridge

École Polytechnique, France

Syntax as Presheaves

Simply-typed single-sorted syntax

$$x_1, \ldots, x_n \vdash t$$

Syntax as Presheaves

Simply-typed single-sorted syntax

$$x_1, \ldots, x_n \vdash t$$

Context:

de Bruijn indices $\rightsquigarrow$ Given by $n \in \mathbb{N}$
Equipped with structural rules (eg. exchange, weakening)

Syntax as Presheaves

Simply-typed single-sorted syntax

$$x_1, \ldots, x_n \vdash t$$

Context:

de Bruijn indices $\rightsquigarrow$ Given by $n \in \mathbb{N}$
Equipped with structural rules (eg. exchange, weakening)

Category of Contexts $\mathbb{C}$:

Some (free) symmetric monoidal category
Objects: Contexts $\rightsquigarrow$ Typically $\mathbb{N}$ (PROP)
Morphisms: Context renaming $\rightsquigarrow$ from structural rules

Syntax as Presheaves

Simply-typed single-sorted syntax

$$x_1, \ldots, x_n \vdash t$$

Context:

de Bruijn indices $\rightsquigarrow$ Given by $n \in \mathbb{N}$
Equipped with structural rules (eg. exchange, weakening)

Category of Contexts $\mathbb{C}$:

Some (free) symmetric monoidal category
Objects: Contexts $\rightsquigarrow$ Typically $\mathbb{N}$ (PROP)
Morphisms: Context renaming $\rightsquigarrow$ from structural rules

Category of Syntax $\mathcal{C} = \mathbf{Set}^{\mathbb{C}}$

A presheaf P in $\mathcal{C}$ abstractly models syntax
$\rightsquigarrow P(n) = \{\text{terms for } P \text{ in context } n\}$
Presheaf action is term renaming

Syntax as Presheaves

Simply-typed single-sorted syntax

$$x_1, \dots, x_n \vdash t$$

Context:

de Bruijn indices $\rightsquigarrow$ Given by $n \in \mathbb{N}$
Equipped with structural rules (eg. exchange, weakening)

Category of Contexts $\mathbb{C}$:

Some (free) symmetric monoidal category
Objects: Contexts $\rightsquigarrow$ Typically $\mathbb{N}$ (PROP)
Morphisms: Context renaming $\rightsquigarrow$ from structural rules

Category of Syntax $\mathcal{C} = \mathbf{Set}^{\mathbb{C}}$

A presheaf P in $\mathcal{C}$ abstractly models syntax
$\rightsquigarrow P(n) = \{\text{terms for } P \text{ in context } n\}$
Presheaf action is term renaming
$\mathcal{C}$ is symmetric monoidal closed with Day convolution $\otimes$

The Cartesian Case

Structural rules on contexts:

Weakening, Contraction and Exchange

The Cartesian Case

Structural rules on contexts:

Weakening, Contraction and Exchange

Category of Cartesian Contexts: $\mathbb{F}$

The category of finite cardinals and functions

Objects: $n \in \mathbb{N}$

Morphisms: $n \to m$ is a function $[n] \to [m]$, where $[n] = \{1, \ldots, n\}$

The Cartesian Case

Structural rules on contexts:

Weakening, Contraction and Exchange

Category of Cartesian Contexts: $\mathbb{F}$

The category of finite cardinals and functions

Objects: $n \in \mathbb{N}$

Morphisms: $n \to m$ is a function $[n] \to [m]$, where $[n] = \{1, \ldots, n\}$

Category of Syntax: $\mathcal{F} = \mathbf{Set}^{\mathbb{F}}$

$\rightsquigarrow$ Object Classifier Topos

$\otimes$ and $\times$ coincide

The Cartesian Case

Structural rules on contexts:

Weakening, Contraction and Exchange

Category of Cartesian Contexts: $\mathbb{F}$

The category of finite cardinals and functions

Objects: $n \in \mathbb{N}$

Morphisms: $n \to m$ is a function $[n] \to [m]$, where $[n] = \{1, \ldots, n\}$

Category of Syntax: $\mathcal{F} = \mathbf{Set}^{\mathbb{F}}$

$\rightsquigarrow$ Object Classifier Topos

$\otimes$ and $\times$ coincide

Goal: Define a structure on a presheaf P in $\mathcal{F}$ which captures substitution for the syntax P.

Simultaneous Substitution for Cartesian Syntax

Simultaneous Substitution for Cartesian Syntax

Presheaf of Variables: V in $\mathcal{F}$

$V = \mathcal{Y}(1) = \mathbb{F}(1, -) : \mathbb{F} \hookrightarrow \mathbf{Set}$

$V(n) = \{1, \ldots, n\}$

V is the cartesian syntax with only variables as terms

Simultaneous Substitution for Cartesian Syntax

Presheaf of Variables: V in $\mathcal{F}$

$V = \mathcal{Y}(1) = \mathbb{F}(1, -) : \mathbb{F} \hookrightarrow \mathbf{Set}$

$V(n) = \{1, \ldots, n\}$

V is the cartesian syntax with only variables as terms

Substitution monoidal tensor: $\circ$ on $\mathcal{F}$

Universally induced from the base category $\mathbb{F}$

$(\mathcal{F}, \circ, V)$ monoidal closed category

Simultaneous Substitution for Cartesian Syntax

Presheaf of Variables: V in $\mathcal{F}$

$V = \mathcal{Y}(1) = \mathbb{F}(1, -) : \mathbb{F} \hookrightarrow \mathbf{Set}$

$V(n) = \{1, \ldots, n\}$

V is the cartesian syntax with only variables as terms

Substitution monoidal tensor: $\circ$ on $\mathcal{F}$

Universally induced from the base category $\mathbb{F}$

$(\mathcal{F}, \circ, V)$ monoidal closed category

Monoids model simultaneous substitution:

$(P, \mu : P \circ P \to P, \eta : V \to P)$

μ performs substitution

η specifies variables

unit laws express substituting with/into variables

associativity law expresses the substitution lemma

Simultaneous Substitution for Cartesian Syntax

Presheaf of Variables: V in $\mathcal{F}$

$V = \mathcal{Y}(1) = \mathbb{F}(1, -) : \mathbb{F} \hookrightarrow \mathbf{Set}$

$V(n) = \{1, \ldots, n\}$

V is the cartesian syntax with only variables as terms

Substitution monoidal tensor: $\circ$ on $\mathcal{F}$

Universally induced from the base category $\mathbb{F}$

$(\mathcal{F}, \circ, V)$ monoidal closed category

Monoids model simultaneous substitution:

$(P, \mu : P \circ P \to P, \eta : V \to P)$

μ performs substitution

η specifies variables

unit laws express substituting with/into variables

associativity law expresses the substitution lemma

Fiore-Plotkin-Turi (1999): $\mathrm{Mon}(\mathcal{F}) \cong \mathbf{Cln} \cong \mathbf{Law}$

The Approach of Superspecies

The Category of Contexts is a free symmetric monoidal category on a symmetric monoidal equational presentation

The Approach of Superspecies

The Category of Contexts is a free symmetric monoidal category on a symmetric monoidal equational presentation

Cartesian Case:

$$\mathfrak{F}: \qquad I \longrightarrow C \longleftarrow C, C \qquad \text{commutative monoid}$$

Operations correspond to structural rules:

Weakening $\rightsquigarrow$ unit
Contraction $\rightsquigarrow$ multiplication
Exchange $\rightsquigarrow$ symmetric monoidal structure

The Approach of Superspecies

The Category of Contexts is a free symmetric monoidal category on a symmetric monoidal equational presentation

Cartesian Case:

$\mathfrak{F}: \qquad I \longrightarrow C \longleftarrow C, C \qquad$ commutative monoid

Operations correspond to structural rules:

Weakening $\rightsquigarrow$ unit
Contraction $\rightsquigarrow$ multiplication
Exchange $\rightsquigarrow$ symmetric monoidal structure

Category of Cartesian Contexts: $\mathbb{F} = \mathrm{Th}(\mathfrak{F})$

$\mathbb{F}$ is the free symmetric monoidal category on a commutative monoid
$\mathrm{Mod}(\mathfrak{F}, \mathcal{E}) \cong \mathbf{SM}(\mathbb{F}, \mathcal{E})$

The Approach of Superspecies

The Category of Contexts is a free symmetric monoidal category on a symmetric monoidal equational presentation

Cartesian Case:

$$\mathfrak{F}: \qquad I \longrightarrow C \longleftarrow C, C \qquad \text{commutative monoid}$$

Operations correspond to structural rules:

Weakening $\rightsquigarrow$ unit
Contraction $\rightsquigarrow$ multiplication
Exchange $\rightsquigarrow$ symmetric monoidal structure

Category of Cartesian Contexts: $\mathbb{F} = \mathrm{Th}(\mathfrak{F})$

$\mathbb{F}$ is the free symmetric monoidal category on a commutative monoid
$\mathrm{Mod}(\mathfrak{F}, \mathcal{E}) \cong \mathbf{SM}(\mathbb{F}, \mathcal{E})$

This universal property induces the substitution tensor on $\mathcal{F}$

Linear, Affine and Relevant Syntax

Linear Case: Exchange

$\mathfrak{B}$: L no equations

$\mathbb{B} = \mathrm{Th}(\mathfrak{B}) \rightsquigarrow$ finite cardinals and bijections

$\mathcal{B} = \mathbf{Set}^{\mathbb{B}} \rightsquigarrow$ Joyal's species

Kelly (2005): $\mathrm{Mon}(\mathcal{B}) \cong \mathbf{SymOp}$

Linear, Affine and Relevant Syntax

Linear Case: Exchange

$\mathfrak{B}: \qquad L \qquad$ no equations

$\mathbb{B} = \mathrm{Th}(\mathfrak{B}) \rightsquigarrow$ finite cardinals and bijections

$\mathcal{B} = \mathbf{Set}^{\mathbb{B}} \rightsquigarrow$ Joyal's species

Kelly (2005): $\mathrm{Mon}(\mathcal{B}) \cong \mathbf{SymOp}$

Affine Case: Weakening and Exchange

$\mathfrak{I}: \qquad I \longrightarrow C \qquad$ no equations

$\mathbb{I} = \mathrm{Th}(\mathfrak{I}) \rightsquigarrow$ finite cardinals and injections

$\mathcal{I} = \mathbf{Set}^{\mathbb{I}}$

Tanaka-Power (2006): $\mathrm{Mon}(\mathcal{I}) \cong \mathbf{AffOp}$

Linear, Affine and Relevant Syntax

Linear Case: Exchange

$\mathfrak{B}:$ L no equations

$\mathbb{B} = \mathrm{Th}(\mathfrak{B}) \rightsquigarrow$ finite cardinals and bijections

$\mathcal{B} = \mathbf{Set}^{\mathbb{B}} \rightsquigarrow$ Joyal's species

Kelly (2005): $\mathrm{Mon}(\mathcal{B}) \cong \mathbf{SymOp}$

Affine Case: Weakening and Exchange

$\mathfrak{I}:$ $I \longrightarrow C$ no equations

$\mathbb{I} = \mathrm{Th}(\mathfrak{I}) \rightsquigarrow$ finite cardinals and injections

$\mathcal{I} = \mathbf{Set}^{\mathbb{I}}$

Tanaka-Power (2006): $\mathrm{Mon}(\mathcal{I}) \cong \mathbf{AffOp}$

Relevant Case: Contraction and Exchange

$\mathfrak{S}:$ $C \longleftarrow C, C$ commutative semigroup

$\mathbb{S} = \mathrm{Th}(\mathfrak{S}) \rightsquigarrow$ finite cardinals and surjections

$\mathcal{S} = \mathbf{Set}^{\mathbb{S}}$

Fiore-R.: $\mathrm{Mon}(\mathcal{S}) \cong \mathbf{RelOp}$

Single-Variable Substitution

Single-Variable Substitution

Syntactic Cartesian Substitution:

$$\frac{x_1, \ldots, x_{n+1} \vdash t \qquad x_1, \ldots, x_n \vdash u}{x_1, \ldots, x_n \vdash t[x_{n+1} := u]}$$

Single-Variable Substitution

Syntactic Cartesian Substitution:

$$\frac{x_1, \ldots, x_{n+1} \vdash t \qquad x_1, \ldots, x_n \vdash u}{x_1, \ldots, x_n \vdash t[x_{n+1} := u]}$$

For a presheaf P in $\mathcal{F}$:

An operation $P(n+1) \times P(n) \to P(n)$ natural in n

Single-Variable Substitution

Syntactic Cartesian Substitution:

$$\frac{x_1, \ldots, x_{n+1} \vdash t \qquad x_1, \ldots, x_n \vdash u}{x_1, \ldots, x_n \vdash t[x_{n+1} := u]}$$

For a presheaf P in $\mathcal{F}$:

An operation $P(n+1) \times P(n) \to P(n)$ natural in n

We want to model $P(n+1) \rightsquigarrow$ Context Extension

Single-Variable Substitution

Syntactic Cartesian Substitution:

$$\frac{x_1, \ldots, x_{n+1} \vdash t \qquad x_1, \ldots, x_n \vdash u}{x_1, \ldots, x_n \vdash t[x_{n+1} := u]}$$

For a presheaf P in $\mathcal{F}$:

An operation $P(n+1) \times P(n) \to P(n)$ natural in n

We want to model $P(n+1) \rightsquigarrow$ Context Extension

Grandis (2001):

$\mathbb{F}$ is the free strict monoidal category on a symmetric monoid

Symmetric Monoids

Q: How do you define a commutative monoid in a non-symmetric monoidal category?

A: By transferring the symmetry from the category to the monoid.

Symmetric Monoids

Q: How do you define a commutative monoid in a non-symmetric monoidal category?

A: By transferring the symmetry from the category to the monoid.

Symmetric Monoid: $(A, \mu, \eta, \varsigma)$ in monoidal $(\mathcal{E}, \otimes, I)$ where

$$\mu : A \otimes A \to A \qquad \eta : I \to A \qquad \varsigma : A \otimes A \to A \otimes A$$

such that the following commute:

Symmetric Monoids

Q: How do you define a commutative monoid in a non-symmetric monoidal category?

A: By transferring the symmetry from the category to the monoid.

Symmetric Monoid: $(A, \mu, \eta, \varsigma)$ in monoidal $(\mathcal{E}, \otimes, I)$ where

$$\mu : A \otimes A \to A \qquad \eta : I \to A \qquad \varsigma : A \otimes A \to A \otimes A$$

such that the following commute:

$$\begin{array}{ccc} A^{\otimes 3} & \xrightarrow{\mu \otimes A} & A^{\otimes 2} \\ {\scriptstyle A\otimes\mu}\downarrow & & \downarrow{\scriptstyle \mu} \\ A^{\otimes 2} & \xrightarrow[\mu]{} & A \end{array} \qquad \begin{array}{ccc} A & \xrightarrow{A \otimes \eta} & A^{\otimes 2} \\ {\scriptstyle \eta\otimes A}\downarrow & \searrow^{A} & \downarrow{\scriptstyle \mu} \\ A^{\otimes 2} & \xrightarrow[\mu]{} & A \end{array}$$

Symmetric Monoids

Q: How do you define a commutative monoid in a non-symmetric monoidal category?

A: By transferring the symmetry from the category to the monoid.

Symmetric Monoid: $(A, \mu, \eta, \varsigma)$ in monoidal $(\mathcal{E}, \otimes, I)$ where

$$\mu : A \otimes A \to A \qquad \eta : I \to A \qquad \varsigma : A \otimes A \to A \otimes A$$

such that the following commute:

$$\begin{array}{ccc} A^{\otimes 3} & \xrightarrow{\mu \otimes A} & A^{\otimes 2} \\ {\scriptstyle A\otimes\mu}\downarrow & & \downarrow{\scriptstyle \mu} \\ A^{\otimes 2} & \xrightarrow[\mu]{} & A \end{array} \qquad \begin{array}{ccc} A & \xrightarrow{A \otimes \eta} & A^{\otimes 2} \\ {\scriptstyle \eta\otimes A}\downarrow & \searrow^{A} & \downarrow{\scriptstyle \mu} \\ A^{\otimes 2} & \xrightarrow[\mu]{} & A \end{array} \qquad \begin{array}{ccc} A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\ & {\scriptstyle \mu}\searrow & \downarrow{\scriptstyle \mu} \\ & & A \end{array}$$

Symmetric Monoids

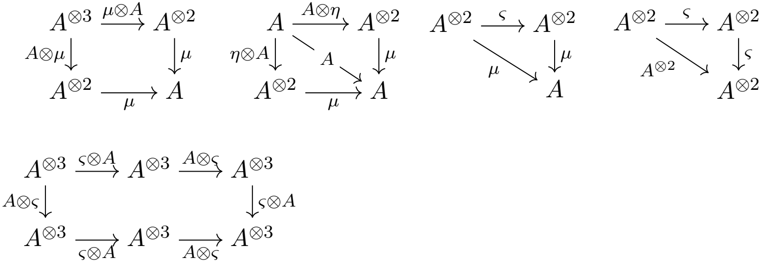
Q: How do you define a commutative monoid in a non-symmetric monoidal category?

A: By transferring the symmetry from the category to the monoid.

Symmetric Monoid: $(A, \mu, \eta, \varsigma)$ in monoidal $(\mathcal{E}, \otimes, I)$ where

$$\mu : A \otimes A \to A \qquad \eta : I \to A \qquad \varsigma : A \otimes A \to A \otimes A$$

such that the following commute:



Symmetric Monoids

Q: How do you define a commutative monoid in a non-symmetric monoidal category?

A: By transferring the symmetry from the category to the monoid.

Symmetric Monoid: $(A, \mu, \eta, \varsigma)$ in monoidal $(\mathcal{E}, \otimes, I)$ where

$$\mu : A \otimes A \to A \qquad \eta : I \to A \qquad \varsigma : A \otimes A \to A \otimes A$$

such that the following commute:

$$
\begin{array}{ccc}
A^{\otimes 3} & \xrightarrow{\mu \otimes A} & A^{\otimes 2} \\
{\scriptstyle A\otimes\mu}\downarrow & & \downarrow{\scriptstyle \mu} \\
A^{\otimes 2} & \xrightarrow[\mu]{} & A
\end{array}
\qquad
\begin{array}{ccc}
A & \xrightarrow{A \otimes \eta} & A^{\otimes 2} \\
{\scriptstyle \eta\otimes A}\downarrow & \searrow^{A} & \downarrow{\scriptstyle \mu} \\
A^{\otimes 2} & \xrightarrow[\mu]{} & A
\end{array}
\qquad
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
 & {}_{\mu}\searrow & \downarrow{\scriptstyle \mu} \\
 & & A
\end{array}
\qquad
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
 & {}_{A^{\otimes 2}}\searrow & \downarrow{\scriptstyle \varsigma} \\
 & & A^{\otimes 2}
\end{array}
$$

$$
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
{\scriptstyle A\otimes\varsigma}\downarrow & & & & \downarrow{\scriptstyle \varsigma\otimes A} \\
A^{\otimes 3} & \xrightarrow[\varsigma \otimes A]{} & A^{\otimes 3} & \xrightarrow[A \otimes \varsigma]{} & A^{\otimes 3}
\end{array}
\qquad
\begin{array}{ccc}
A & \xrightarrow{\eta \otimes A} & A^{\otimes 2} \\
 & {}_{A\otimes\eta}\searrow & \downarrow{\scriptstyle \varsigma} \\
 & & A^{\otimes 2}
\end{array}
\qquad
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
{\scriptstyle A\otimes\mu}\downarrow & & & & \downarrow{\scriptstyle \mu\otimes A} \\
A^{\otimes 2} & & \xrightarrow[\varsigma]{} & & A^{\otimes 2}
\end{array}
$$

Symmetric Monads

Symmetric Monad:

A symmetric monoid in a category of endofunctors

Symmetric Monads

Symmetric Monad:

A symmetric monoid in a category of endofunctors

Grandis (2001):

$\mathbb{F}$ is the free strict monoidal category on a symmetric monoid

Generating Structure: $(1, c : 2 \to 1, w : 0 \to 1, s : 2 \to 2)$

Operations directly correspond to structural rules

Symmetric Monads

Symmetric Monad:

A symmetric monoid in a category of endofunctors

Grandis (2001):

$\mathbb{F}$ is the free strict monoidal category on a symmetric monoid

Generating Structure: $(1, c : 2 \to 1, w : 0 \to 1, s : 2 \to 2)$

Operations directly correspond to structural rules

The endofunctor $- + 1 : \mathbb{F} \to \mathbb{F}$ is a symmetric monad:

$$- + c : - + 2 \to - + 1$$
$$- + w : - + 0 \to - + 1$$
$$- + s : - + 2 \to - + 2$$

Context Extension

This can be lifted to the presheaf category $\mathcal{F}$:

$$\begin{array}{ccc} \mathbb{F}^{\mathrm{op}} & \xrightarrow{\mathcal{Y}} & \mathcal{F} \\ {\scriptstyle (-+1)^{\mathrm{op}}}\downarrow & & \uparrow{\scriptstyle \delta} \\ \mathbb{F}^{\mathrm{op}} & \xrightarrow{\mathcal{Y}} & \mathcal{F} \end{array}$$

$$\delta(P)(n) = P(n+1)$$

Context Extension

This can be lifted to the presheaf category $\mathcal{F}$:

$$\begin{array}{ccc} \mathbb{F}^{\mathrm{op}} & \xrightarrow{\ \mathcal{Y}\ } & \mathcal{F} \\ {\scriptstyle (-+1)^{\mathrm{op}}}\downarrow & & \uparrow{\scriptstyle \delta} \\ \mathbb{F}^{\mathrm{op}} & \xrightarrow{\ \mathcal{Y}\ } & \mathcal{F} \end{array}$$

$$\delta(P)(n) = P(n+1)$$

δ is a symmetric monad:

$\mathbf{cont}_{P,n} = P(n+c) : \delta^2(P)(n) \to \delta(P)(n)$
$\mathbf{weak}_{P,n} = P(n+w) : P(n) \to \delta(P)(n)$
$\mathbf{swap}_{P,n} = P(n+s) : \delta^2(P)(n) \to \delta^2(P)(n)$

Context Extension

This can be lifted to the presheaf category $\mathcal{F}$:

$$\begin{array}{ccc} \mathbb{F}^{\mathrm{op}} & \xrightarrow{\;\mathcal{Y}\;} & \mathcal{F} \\ {\scriptstyle (-+1)^{\mathrm{op}}}\downarrow & & {\scriptstyle -\times V}\downarrow \dashv \overset{\uparrow}{\delta} \dashv \downarrow{\scriptstyle R} \\ \mathbb{F}^{\mathrm{op}} & \xrightarrow{\;\mathcal{Y}\;} & \mathcal{F} \end{array} \qquad \delta(P)(n) = P(n+1)$$

δ is a symmetric monad:

$\mathbf{cont}_{P,n} = P(n+c) : \delta^2(P)(n) \to \delta(P)(n)$
$\mathbf{weak}_{P,n} = P(n+w) : P(n) \to \delta(P)(n)$
$\mathbf{swap}_{P,n} = P(n+s) : \delta^2(P)(n) \to \delta^2(P)(n)$

$- \times V \dashv \delta \dashv R$:

Left and right adjoints given by Yoneda extensions
δ is cartesian and cocartesian monoidal
$\delta \cong (-)^V$

Context Extension

This can be lifted to the presheaf category $\mathcal{F}$:

$$
\begin{array}{ccc}
\mathbb{F}^{\mathrm{op}} & \xrightarrow{\quad \mathcal{Y} \quad} & \mathcal{F} \\
{\scriptstyle (-+1)^{\mathrm{op}}}\big\downarrow & & {\scriptstyle -\times V}\big\downarrow \dashv \big\uparrow\delta \dashv \big\downarrow{\scriptstyle R} \\
\mathbb{F}^{\mathrm{op}} & \xrightarrow{\quad \mathcal{Y} \quad} & \mathcal{F}
\end{array}
\qquad \delta(P)(n) = P(n+1)
$$

δ is a symmetric monad:

$\mathbf{cont}_{P,n} = P(n+c) : \delta^2(P)(n) \to \delta(P)(n)$
$\mathbf{weak}_{P,n} = P(n+w) : P(n) \to \delta(P)(n)$
$\mathbf{swap}_{P,n} = P(n+s) : \delta^2(P)(n) \to \delta^2(P)(n)$

$- \times V \dashv \delta \dashv R$:

Left and right adjoints given by Yoneda extensions
δ is cartesian and cocartesian monoidal
$\delta \cong (-)^V$

δ is a strong symmetric monad:

$\mathbf{str} : \delta(P) \times Q \to \delta(P \times Q)$
Strength respects the symmetric monad structure

Cartesian Substitution Algebra

Need: Variables and Substitution

Cartesian Substitution Algebra

Need: Variables and Substitution

Variables: $V \to P$ or equivalently, $\nu : 1 \to \delta(V)$

Cartesian Substitution Algebra

Need: Variables and Substitution

Variables: $V \to P$ or equivalently, $\nu : 1 \to \delta(V)$

Substitution: $\sigma : \delta(P) \times P \to P$

Cartesian Substitution Algebra

Need: Variables and Substitution

Variables: $V \to P$ or equivalently, $\nu : 1 \to \delta(V)$
Substitution: $\sigma : \delta(P) \times P \to P$

A cartesian substitution algebra is a triple (P, σ, ν) such that the following commute:

$$\begin{array}{ccc} 1 \times P & \xrightarrow{\cong} & P \\ {\scriptstyle \nu \times P}\downarrow & \nearrow_{\sigma} & \\ \delta(P) \times P & & \end{array}$$

Cartesian Substitution Algebra

Need: Variables and Substitution

Variables: $V \to P$ or equivalently, $\nu : 1 \to \delta(V)$

Substitution: $\sigma : \delta(P) \times P \to P$

A cartesian substitution algebra is a triple (P, σ, ν) such that the following commute:

$$\begin{array}{ccc} 1 \times P & \xrightarrow{\cong} & P \\ {\scriptstyle \nu \times P}\downarrow & \nearrow_{\sigma} & \\ \delta(P) \times P & & \end{array} \qquad \begin{array}{ccccc} \delta^2(P) \times 1 & & \xrightarrow{\cong} & & \delta^2(P) \\ {\scriptstyle \delta^2(P)\times\nu}\downarrow & & & & \downarrow{\scriptstyle \mathbf{cont}_P} \\ \delta^2(P) \times \delta(P) & \xrightarrow[\cong]{} & \delta(\delta(P) \times P) & \xrightarrow[\delta(\sigma)]{} & \delta(P) \end{array}$$

Cartesian Substitution Algebra

Need: Variables and Substitution

Variables: $V \to P$ or equivalently, $\nu : 1 \to \delta(V)$

Substitution: $\sigma : \delta(P) \times P \to P$

A cartesian substitution algebra is a triple (P, σ, ν) such that the following commute:

$$\begin{array}{ccc} 1 \times P & \xrightarrow{\cong} & P \\ {\scriptstyle \nu \times P} \downarrow & \nearrow {\scriptstyle \sigma} & \\ \delta(P) \times P & & \end{array}$$

$$\begin{array}{ccccc} \delta^2(P) \times 1 & & \xrightarrow{\cong} & & \delta^2(P) \\ {\scriptstyle \delta^2(P) \times \nu} \downarrow & & & & \downarrow {\scriptstyle \mathbf{cont}_P} \\ \delta^2(P) \times \delta(P) & \xrightarrow[\cong]{} & \delta(\delta(P) \times P) & \xrightarrow[\delta(\sigma)]{} & \delta(P) \end{array}$$

$$\begin{array}{ccc} P \times P & \xrightarrow{\pi_1} & P \\ {\scriptstyle \mathbf{weak}_P \times P} \downarrow & \nearrow {\scriptstyle \sigma} & \\ \delta(P) \times P & & \end{array}$$

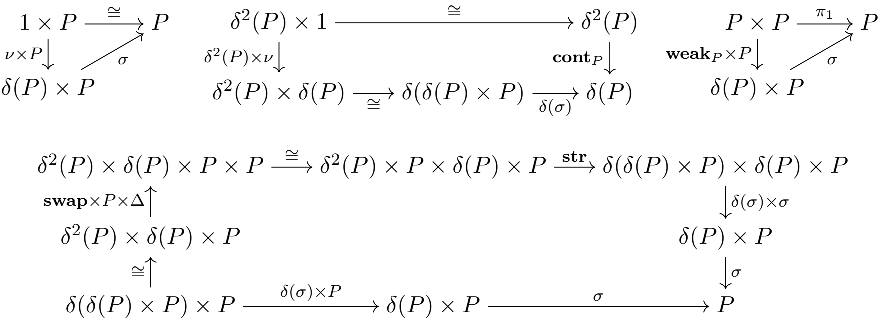
Cartesian Substitution Algebra

Need: Variables and Substitution

Variables: $V \to P$ or equivalently, $\nu : 1 \to \delta(V)$

Substitution: $\sigma : \delta(P) \times P \to P$

A cartesian substitution algebra is a triple (P, σ, ν) such that the following commute:



Cartesian Substitution Algebra

Need: Variables and Substitution

Variables: $V \to P$ or equivalently, $\nu : 1 \to \delta(V)$
Substitution: $\sigma : \delta(P) \times P \to P$

A cartesian substitution algebra is a triple (P, σ, ν) such that the following commute:

$$\begin{array}{ccc} 1 \times P & \xrightarrow{\cong} & P \\ {\scriptstyle \nu \times P}\downarrow & \nearrow{\scriptstyle \sigma} & \\ \delta(P) \times P & & \end{array}
\qquad
\begin{array}{ccccc} \delta^2(P) \times 1 & & \xrightarrow{\cong} & & \delta^2(P) \\ {\scriptstyle \delta^2(P)\times\nu}\downarrow & & & & \downarrow{\scriptstyle \mathbf{cont}_P} \\ \delta^2(P) \times \delta(P) & \xrightarrow[\cong]{} & \delta(\delta(P) \times P) & \xrightarrow[\delta(\sigma)]{} & \delta(P) \end{array}
\qquad
\begin{array}{ccc} P \times P & \xrightarrow{\pi_1} & P \\ {\scriptstyle \mathbf{weak}_P \times P}\downarrow & \nearrow{\scriptstyle \sigma} & \\ \delta(P) \times P & & \end{array}$$

$$\begin{array}{ccccc} \delta^2(P) \times \delta(P) \times P \times P & \xrightarrow{\cong} & \delta^2(P) \times P \times \delta(P) \times P & \xrightarrow{\mathbf{str}} & \delta(\delta(P) \times P) \times \delta(P) \times P \\ {\scriptstyle \mathbf{swap} \times P \times \Delta}\uparrow & & & & \downarrow{\scriptstyle \delta(\sigma)\times\sigma} \\ \delta^2(P) \times \delta(P) \times P & & & & \delta(P) \times P \\ {\scriptstyle \cong}\uparrow & & & & \downarrow{\scriptstyle \sigma} \\ \delta(\delta(P) \times P) \times P & \xrightarrow{\delta(\sigma)\times P} & \delta(P) \times P & \xrightarrow{\sigma} & P \end{array}$$

Fiore-Plotkin-Turi (1999): $\mathbf{CSubstAlg} \cong \mathrm{Mon}(\mathcal{F})$

General Approach

SMEqP: $\mathfrak{C}$

General Approach

SMEqP: $\mathfrak{C}$

Corresponding **MEqP**: $\bar{\mathfrak{C}}$

General Approach

SMEqP: $\mathfrak{C}$

Corresponding **MEqP**: $\bar{\mathfrak{C}}$

Category of Contexts: $\mathbb{C} = \mathrm{Th}(\mathfrak{C}) = \mathrm{Th}(\bar{\mathfrak{C}})$

General Approach

SMEqP: $\mathfrak{C}$

Corresponding **MEqP**: $\bar{\mathfrak{C}}$

Category of Contexts: $\mathbb{C} = \mathrm{Th}(\mathfrak{C}) = \mathrm{Th}(\bar{\mathfrak{C}})$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathbb{C}$): $(- + 1)$

General Approach

SMEqP: $\mathfrak{C}$

Corresponding **MEqP**: $\bar{\mathfrak{C}}$

Category of Contexts: $\mathbb{C} = \mathrm{Th}(\mathfrak{C}) = \mathrm{Th}(\bar{\mathfrak{C}})$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathbb{C}$): $(- + 1)$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathcal{C}$): $\delta \rightsquigarrow$ Yoneda restriction of $- + 1$

General Approach

SMEqP: $\mathfrak{C}$

Corresponding **MEqP**: $\bar{\mathfrak{C}}$

Category of Contexts: $\mathbb{C} = \mathrm{Th}(\mathfrak{C}) = \mathrm{Th}(\bar{\mathfrak{C}})$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathbb{C}$): $(- + 1)$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathcal{C}$): $\delta \rightsquigarrow$ Yoneda restriction of $- + 1$

Presheaf of Variables: $V = \mathcal{Y}(1)$

General Approach

SMEqP: $\mathfrak{C}$

Corresponding **MEqP**: $\bar{\mathfrak{C}}$

Category of Contexts: $\mathbb{C} = \mathrm{Th}(\mathfrak{C}) = \mathrm{Th}(\bar{\mathfrak{C}})$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathbb{C}$): $(- + 1)$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathcal{C}$): $\delta \rightsquigarrow$ Yoneda restriction of $- + 1$

Presheaf of Variables: $V = \mathcal{Y}(1)$

δ has $\otimes$-strength: $- \otimes V \dashv \delta \quad \rightsquigarrow \quad \delta \cong - \multimap V$

General Approach

SMEqP: $\mathfrak{C}$

Corresponding **MEqP**: $\bar{\mathfrak{C}}$

Category of Contexts: $\mathbb{C} = \mathrm{Th}(\mathfrak{C}) = \mathrm{Th}(\bar{\mathfrak{C}})$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathbb{C}$): $(- + 1)$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathcal{C}$): $\delta \rightsquigarrow$ Yoneda restriction of $- + 1$

Presheaf of Variables: $V = \mathcal{Y}(1)$

δ has $\otimes$-strength: $- \otimes V \dashv \delta \quad \rightsquigarrow \quad \delta \cong - \multimap V$

Substitution Algebra: $(P, \sigma : \delta(P) \otimes P \to P, \nu : I \to \delta(P))$

General Approach

SMEqP: $\mathfrak{C}$

Corresponding **MEqP**: $\bar{\mathfrak{C}}$

Category of Contexts: $\mathbb{C} = \mathrm{Th}(\mathfrak{C}) = \mathrm{Th}(\bar{\mathfrak{C}})$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathbb{C}$): $(- + 1)$

Model of $\bar{\mathfrak{C}}$ in Endo($\mathcal{C}$): $\delta \rightsquigarrow$ Yoneda restriction of $- + 1$

Presheaf of Variables: $V = \mathcal{Y}(1)$

δ has $\otimes$-strength: $- \otimes V \dashv \delta \quad \rightsquigarrow \quad \delta \cong - \multimap V$

Substitution Algebra: $(P, \sigma : \delta(P) \otimes P \to P, \nu : I \to \delta(P))$

But what about the axioms?

The Linear Case

Monoidal Presentation: Symmetric Object (A, ς) with

$$\begin{array}{ccc} A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\ & \searrow^{A^{\otimes 2}} & \downarrow{\scriptstyle \varsigma} \\ & & A^{\otimes 2} \end{array} \qquad \begin{array}{ccccc} A^{\otimes 3} & \xrightarrow{\varsigma\otimes A} & A^{\otimes 3} & \xrightarrow{A\otimes\varsigma} & A^{\otimes 3} \\ {\scriptstyle A\otimes\varsigma}\downarrow & & & & \downarrow{\scriptstyle \varsigma\otimes A} \\ A^{\otimes 3} & \xrightarrow[\varsigma\otimes A]{} & A^{\otimes 3} & \xrightarrow[A\otimes\varsigma]{} & A^{\otimes 3} \end{array}$$

A symmetric endofunctor is a symmetric object in a category of endofunctors

The Linear Case

Monoidal Presentation: Symmetric Object (A, ς) with

$$
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
& \searrow^{A^{\otimes 2}} & \downarrow \varsigma \\
& & A^{\otimes 2}
\end{array}
\qquad
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
{\scriptstyle A \otimes \varsigma} \downarrow & & & & \downarrow {\scriptstyle \varsigma \otimes A} \\
A^{\otimes 3} & \xrightarrow[\varsigma \otimes A]{} & A^{\otimes 3} & \xrightarrow[A \otimes \varsigma]{} & A^{\otimes 3}
\end{array}
$$

A symmetric endofunctor is a symmetric object in a category of endofunctors

$\mathbb{B}$ is the free monoidal category on a symmetric object $(1, s : 2 \to 2)$

The Linear Case

Monoidal Presentation: Symmetric Object (A, ς) with

$$
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
& {\scriptstyle A^{\otimes 2}}\searrow & \downarrow{\scriptstyle \varsigma} \\
& & A^{\otimes 2}
\end{array}
\qquad
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
{\scriptstyle A\otimes\varsigma}\downarrow & & & & \downarrow{\scriptstyle \varsigma\otimes A} \\
A^{\otimes 3} & \xrightarrow[\varsigma \otimes A]{} & A^{\otimes 3} & \xrightarrow[A \otimes \varsigma]{} & A^{\otimes 3}
\end{array}
$$

A symmetric endofunctor is a symmetric object in a category of endofunctors

$\mathbb{B}$ is the free monoidal category on a symmetric object $(1, s : 2 \to 2)$

On $\mathbb{B}$: $(- + 1, - + s)$ is a symmetric endofunctor

On $\mathcal{B}$: $(\delta, \mathbf{swap})$ is a strong symmetric endofunctor

Linear Substitution Algebra

A linear substitution algebra is a triple (P, σ, ν) such that:

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\downarrow & \nearrow_{\sigma} & \\
\delta(P) \otimes P & &
\end{array}
\qquad
\begin{array}{cccc}
\delta(P) \otimes I & & \xrightarrow{\cong} & \delta(P) \\
{\scriptstyle \delta(P) \otimes \nu}\downarrow & & & \nearrow_{\delta(\sigma)} \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P) \otimes P) &
\end{array}
$$

Linear Substitution Algebra

A linear substitution algebra is a triple (P, σ, ν) such that:

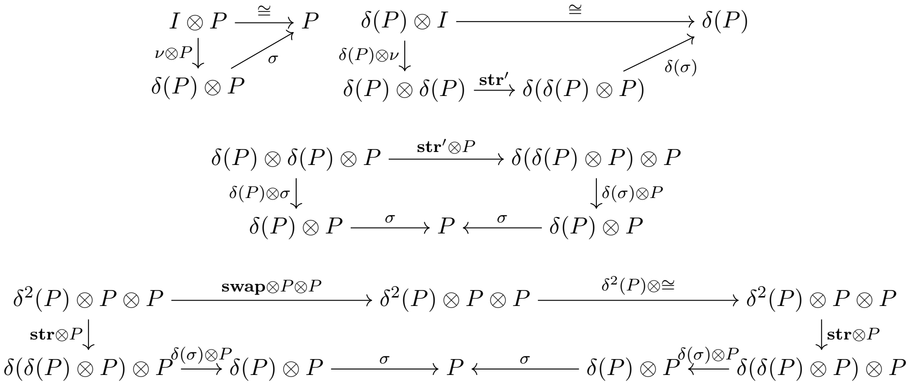
$$\begin{array}{ccc} I \otimes P & \xrightarrow{\cong} & P \\ {\scriptstyle \nu \otimes P}\downarrow & \nearrow{\scriptstyle \sigma} & \\ \delta(P) \otimes P & & \end{array} \qquad \begin{array}{ccccc} \delta(P) \otimes I & & \xrightarrow{\cong} & & \delta(P) \\ {\scriptstyle \delta(P)\otimes\nu}\downarrow & & & \nearrow{\scriptstyle \delta(\sigma)} & \\ \delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P) \otimes P) & & \end{array}$$

$$\begin{array}{ccccc} \delta(P) \otimes \delta(P) \otimes P & & \xrightarrow{\mathbf{str}' \otimes P} & & \delta(\delta(P) \otimes P) \otimes P \\ {\scriptstyle \delta(P)\otimes\sigma}\downarrow & & & & \downarrow{\scriptstyle \delta(\sigma)\otimes P} \\ \delta(P) \otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P) \otimes P \end{array}$$

$$\begin{array}{ccccccccc} \delta^2(P) \otimes P \otimes P & & & \xrightarrow{\mathbf{swap} \otimes P \otimes P} & \delta^2(P) \otimes P \otimes P & \xrightarrow{\delta^2(P) \otimes \cong} & & & \delta^2(P) \otimes P \otimes P \\ {\scriptstyle \mathbf{str}\otimes P}\downarrow & & & & & & & & \downarrow{\scriptstyle \mathbf{str}\otimes P} \\ \delta(\delta(P) \otimes P) \otimes P & \xrightarrow{\delta(\sigma)\otimes P} & \delta(P) \otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P) \otimes P & \xleftarrow{\delta(\sigma)\otimes P} & \delta(\delta(P) \otimes P) \otimes P \end{array}$$

Linear Substitution Algebra

A linear substitution algebra is a triple (P, σ, ν) such that:



$$\mathbf{LSubstAlg} \cong \mathrm{Mon}(\mathcal{B})$$

The Affine Case

Monoidal Presentation: Symmetric Pointed Object (A, η, ς) with

$$
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
& {\scriptstyle A^{\otimes 2}} \searrow & \downarrow{\scriptstyle \varsigma} \\
& & A^{\otimes 2}
\end{array}
\qquad
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
{\scriptstyle A \otimes \varsigma}\downarrow & & & & \downarrow{\scriptstyle \varsigma \otimes A} \\
A^{\otimes 3} & \xrightarrow[\varsigma \otimes A]{} & A^{\otimes 3} & \xrightarrow[A \otimes \varsigma]{} & A^{\otimes 3}
\end{array}
\qquad
\begin{array}{ccc}
A & \xrightarrow{\eta \otimes A} & A^{\otimes 2} \\
& {\scriptstyle A \otimes \eta} \searrow & \downarrow{\scriptstyle \varsigma} \\
& & A^{\otimes 2}
\end{array}
$$

A symmetric pointed endofunctor is a symmetric pointed object in a category of endofunctors

The Affine Case

Monoidal Presentation: Symmetric Pointed Object (A, η, ς) with

$$
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
& {\scriptstyle A^{\otimes 2}} \searrow & \downarrow{\scriptstyle \varsigma} \\
& & A^{\otimes 2}
\end{array}
\qquad
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma\otimes A} & A^{\otimes 3} & \xrightarrow{A\otimes\varsigma} & A^{\otimes 3} \\
{\scriptstyle A\otimes\varsigma}\downarrow & & & & \downarrow{\scriptstyle \varsigma\otimes A} \\
A^{\otimes 3} & \xrightarrow[\varsigma\otimes A]{} & A^{\otimes 3} & \xrightarrow[A\otimes\varsigma]{} & A^{\otimes 3}
\end{array}
\qquad
\begin{array}{ccc}
A & \xrightarrow{\eta\otimes A} & A^{\otimes 2} \\
& {\scriptstyle A\otimes\eta} \searrow & \downarrow{\scriptstyle \varsigma} \\
& & A^{\otimes 2}
\end{array}
$$

A symmetric pointed endofunctor is a symmetric pointed object in a category of endofunctors

$\mathbb{I}$ is the free monoidal category on a symmetric pointed object $(1, w : 0 \to 1, s : 2 \to 2)$

The Affine Case

Monoidal Presentation: Symmetric Pointed Object (A, η, ς) with

$$
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
& {\scriptstyle A^{\otimes 2}} \searrow & \downarrow{\scriptstyle \varsigma} \\
& & A^{\otimes 2}
\end{array}
\qquad
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
{\scriptstyle A \otimes \varsigma}\downarrow & & & & \downarrow{\scriptstyle \varsigma \otimes A} \\
A^{\otimes 3} & \xrightarrow[\varsigma \otimes A]{} & A^{\otimes 3} & \xrightarrow[A \otimes \varsigma]{} & A^{\otimes 3}
\end{array}
\qquad
\begin{array}{ccc}
A & \xrightarrow{\eta \otimes A} & A^{\otimes 2} \\
& {\scriptstyle A \otimes \eta} \searrow & \downarrow{\scriptstyle \varsigma} \\
& & A^{\otimes 2}
\end{array}
$$

A symmetric pointed endofunctor is a symmetric pointed object in a category of endofunctors

$\mathbb{I}$ is the free monoidal category on a symmetric pointed object $(1, w : 0 \to 1, s : 2 \to 2)$

On $\mathbb{I}$: $(- + 1, - + w, - + s)$ is a symmetric pointed endofunctor
On $\mathcal{I}$: $(\delta, \mathbf{weak}, \mathbf{swap})$ is a strong symmetric pointed endofunctor

Affine Substitution Algebra

An affine substitution algebra is a triple (P, σ, ν) such that:

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\downarrow & \nearrow{\scriptstyle \sigma} & \\
\delta(P) \otimes P & &
\end{array}
\qquad
\begin{array}{cccc}
\delta(P) \otimes I & & \xrightarrow{\cong} & \delta(P) \\
{\scriptstyle \delta(P) \otimes \nu}\downarrow & & \nearrow{\scriptstyle \delta(\sigma)} & \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P) \otimes P) &
\end{array}
$$

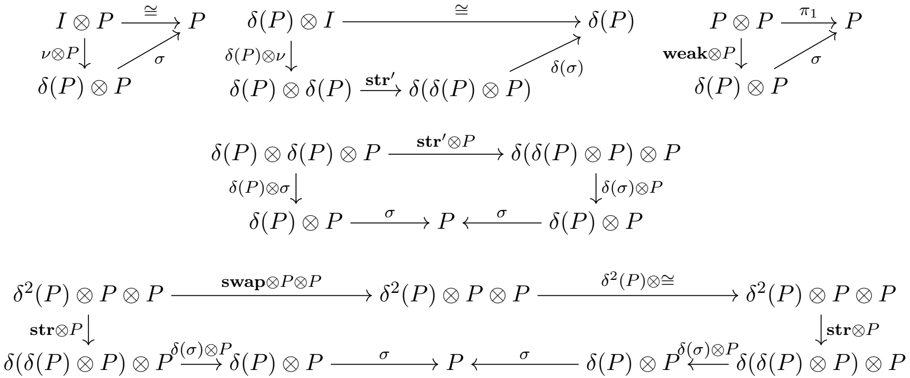
Affine Substitution Algebra

An affine substitution algebra is a triple (P, σ, ν) such that:

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\downarrow & \nearrow{\scriptstyle \sigma} & \\
\delta(P) \otimes P & &
\end{array}
\qquad
\begin{array}{ccccc}
\delta(P) \otimes I & & \xrightarrow{\cong} & & \delta(P) \\
{\scriptstyle \delta(P)\otimes\nu}\downarrow & & & \nearrow{\scriptstyle \delta(\sigma)} & \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str'}} & \delta(\delta(P) \otimes P) & &
\end{array}
\qquad
\begin{array}{ccc}
P \otimes P & \xrightarrow{\pi_1} & P \\
{\scriptstyle \mathbf{weak} \otimes P}\downarrow & \nearrow{\scriptstyle \sigma} & \\
\delta(P) \otimes P & &
\end{array}
$$

Affine Substitution Algebra

An affine substitution algebra is a triple (P, σ, ν) such that:



Affine Substitution Algebra

An affine substitution algebra is a triple (P, σ, ν) such that:

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\downarrow & \nearrow_{\sigma} & \\
\delta(P) \otimes P & &
\end{array}
\qquad
\begin{array}{ccc}
\delta(P) \otimes I & \xrightarrow{\cong} & \delta(P) \\
{\scriptstyle \delta(P) \otimes \nu}\downarrow & & \nearrow_{\delta(\sigma)} \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P) \otimes P)
\end{array}
\qquad
\begin{array}{ccc}
P \otimes P & \xrightarrow{\pi_1} & P \\
{\scriptstyle \mathbf{weak} \otimes P}\downarrow & \nearrow_{\sigma} & \\
\delta(P) \otimes P & &
\end{array}
$$

$$
\begin{array}{ccccc}
\delta(P) \otimes \delta(P) \otimes P & & \xrightarrow{\mathbf{str}' \otimes P} & & \delta(\delta(P) \otimes P) \otimes P \\
{\scriptstyle \delta(P) \otimes \sigma}\downarrow & & & & \downarrow{\scriptstyle \delta(\sigma) \otimes P} \\
\delta(P) \otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P) \otimes P
\end{array}
$$

$$
\begin{array}{ccccccccc}
\delta^2(P) \otimes P \otimes P & & & \xrightarrow{\mathbf{swap} \otimes P \otimes P} & \delta^2(P) \otimes P \otimes P & \xrightarrow{\delta^2(P) \otimes \cong} & & & \delta^2(P) \otimes P \otimes P \\
{\scriptstyle \mathbf{str} \otimes P}\downarrow & & & & & & & & \downarrow{\scriptstyle \mathbf{str} \otimes P} \\
\delta(\delta(P) \otimes P) \otimes P & \xrightarrow{\delta(\sigma) \otimes P} & \delta(P) \otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P) \otimes P & \xleftarrow{\delta(\sigma) \otimes P} & \delta(\delta(P) \otimes P) \otimes P
\end{array}
$$

$$\mathbf{ASubstAlg} \cong \mathrm{Mon}(\mathcal{I})$$

The Relevant Case

Monoidal Presentation: Symmetric Semigroup (A, μ, ς) with

$$\begin{array}{ccc} A^{\otimes 3} & \xrightarrow{\mu \otimes A} & A^{\otimes 2} \\ {\scriptstyle A\otimes\mu}\downarrow & & \downarrow{\scriptstyle \mu} \\ A^{\otimes 2} & \xrightarrow[\mu]{} & A \end{array} \qquad \begin{array}{ccc} A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\ & {\scriptstyle \mu}\searrow & \downarrow{\scriptstyle \mu} \\ & & A \end{array} \qquad \begin{array}{ccccc} A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\ {\scriptstyle A\otimes\mu}\downarrow & & & & \downarrow{\scriptstyle \mu\otimes A} \\ A^{\otimes 2} & & \xrightarrow[\varsigma]{} & & A^{\otimes 2} \end{array}$$

$$\begin{array}{ccc} A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\ & {\scriptstyle A^{\otimes 2}}\searrow & \downarrow{\scriptstyle \varsigma} \\ & & A^{\otimes 2} \end{array} \qquad \begin{array}{ccccc} A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\ {\scriptstyle A\otimes\varsigma}\downarrow & & & & \downarrow{\scriptstyle \varsigma\otimes A} \\ A^{\otimes 3} & \xrightarrow[\varsigma\otimes A]{} & A^{\otimes 3} & \xrightarrow[A\otimes\varsigma]{} & A^{\otimes 3} \end{array}$$

A symmetric multiplicative endofunctor is a symmetric semigroup in a category of endofunctors

The Relevant Case

Monoidal Presentation: Symmetric Semigroup (A, μ, ς) with

$$
\begin{array}{ccc}
A^{\otimes 3} & \xrightarrow{\mu \otimes A} & A^{\otimes 2} \\
{\scriptstyle A\otimes\mu}\downarrow & & \downarrow{\scriptstyle \mu} \\
A^{\otimes 2} & \xrightarrow[\mu]{} & A
\end{array}
\qquad
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
& {\scriptstyle \mu}\searrow & \downarrow{\scriptstyle \mu} \\
& & A
\end{array}
\qquad
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
{\scriptstyle A\otimes\mu}\downarrow & & & & \downarrow{\scriptstyle \mu\otimes A} \\
A^{\otimes 2} & & \xrightarrow[\varsigma]{} & & A^{\otimes 2}
\end{array}
$$

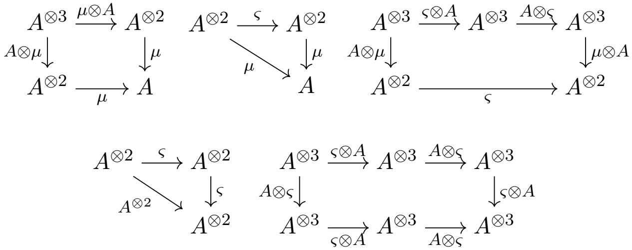
$$
\begin{array}{ccc}
A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\
& {\scriptstyle A^{\otimes 2}}\searrow & \downarrow{\scriptstyle \varsigma} \\
& & A^{\otimes 2}
\end{array}
\qquad
\begin{array}{ccccc}
A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
{\scriptstyle A\otimes\varsigma}\downarrow & & & & \downarrow{\scriptstyle \varsigma\otimes A} \\
A^{\otimes 3} & \xrightarrow[\varsigma \otimes A]{} & A^{\otimes 3} & \xrightarrow[A \otimes \varsigma]{} & A^{\otimes 3}
\end{array}
$$

A symmetric multiplicative endofunctor is a symmetric semigroup in a category of endofunctors

$\mathbb{S}$ is the free monoidal category on a symmetric semigroup $(1, c : 2 \to 1, s : 2 \to 2)$

The Relevant Case

Monoidal Presentation: Symmetric Semigroup (A, μ, ς) with



A symmetric multiplicative endofunctor is a symmetric semigroup in a category of endofunctors

$\mathbb{S}$ is the free monoidal category on a symmetric semigroup $(1, c : 2 \to 1, s : 2 \to 2)$

On $\mathbb{S}$: $(- + 1, - + c, - + s)$ is a symmetric multiplicative endofunctor
On $\mathcal{S}$: $(\delta, \mathbf{cont}, \mathbf{swap})$ is a strong symmetric multiplicative endofunctor

Relevant Substitution Algebra

A relevant substitution algebra is a triple (P, σ, ν) such that:

$$\begin{array}{ccc} I \otimes P & \xrightarrow{\cong} & P \\ {\scriptstyle \nu \otimes P}\downarrow & \nearrow_{\sigma} & \\ \delta(P) \otimes P & & \end{array} \qquad \begin{array}{ccccc} \delta(P) \otimes I & & \xrightarrow{\cong} & & \delta(P) \\ {\scriptstyle \delta(P)\otimes\nu}\downarrow & & & \nearrow_{\delta(\sigma)} & \\ \delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P) \otimes P) & & \end{array}$$

Relevant Substitution Algebra

A relevant substitution algebra is a triple (P, σ, ν) such that:

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\big\downarrow & \nearrow_{\sigma} & \\
\delta(P) \otimes P & &
\end{array}
\qquad
\begin{array}{ccccc}
\delta(P) \otimes I & & \xrightarrow{\cong} & & \delta(P) \\
{\scriptstyle \delta(P) \otimes \nu}\big\downarrow & & & \nearrow_{\delta(\sigma)} & \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P) \otimes P) & &
\end{array}
$$

$$
\begin{array}{ccccc}
\delta(P) \otimes \delta(P) \otimes P & & \xrightarrow{\mathbf{str}' \otimes P} & & \delta(\delta(P) \otimes P) \otimes P \\
{\scriptstyle \delta(P) \otimes \sigma}\big\downarrow & & & & \big\downarrow{\scriptstyle \delta(\sigma) \otimes P} \\
\delta(P) \otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P) \otimes P
\end{array}
$$

Relevant Substitution Algebra

A relevant substitution algebra is a triple (P, σ, ν) such that:

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\downarrow & \nearrow{\scriptstyle \sigma} & \\
\delta(P) \otimes P & &
\end{array}
\qquad
\begin{array}{ccccc}
\delta(P) \otimes I & & \xrightarrow{\cong} & & \delta(P) \\
{\scriptstyle \delta(P)\otimes\nu}\downarrow & & & \nearrow{\scriptstyle \delta(\sigma)} & \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P) \otimes P) & &
\end{array}
$$

$$
\begin{array}{ccccc}
\delta(P) \otimes \delta(P) \otimes P & & \xrightarrow{\mathbf{str}' \otimes P} & & \delta(\delta(P) \otimes P) \otimes P \\
{\scriptstyle \delta(P)\otimes\sigma}\downarrow & & & & \downarrow{\scriptstyle \delta(\sigma)\otimes P} \\
\delta(P) \otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P) \otimes P
\end{array}
$$

$$
\begin{array}{ccccccc}
\delta^2(P) \otimes P \otimes P & & \xrightarrow{\mathbf{swap} \otimes P \otimes P} & \delta^2(P) \otimes P \otimes P & \xrightarrow{\delta^2(P)\otimes\cong} & & \delta^2(P) \otimes P \otimes P \\
{\scriptstyle \mathbf{str}\otimes P}\downarrow & & & & & & \downarrow{\scriptstyle \mathbf{str}\otimes P} \\
\delta(\delta(P) \otimes P) \otimes P & \xrightarrow{\delta(\sigma)\otimes P} & \delta(P) \otimes P & \xrightarrow{\sigma} P \xleftarrow{\sigma} & \delta(P) \otimes P & \xleftarrow{\delta(\sigma)\otimes P} & \delta(\delta(P) \otimes P) \otimes P
\end{array}
$$

Relevant Substitution Algebra

A relevant substitution algebra is a triple (P, σ, ν) such that:

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\downarrow & \nearrow{\scriptstyle \sigma} & \\
\delta(P) \otimes P & &
\end{array}
\qquad
\begin{array}{ccccc}
\delta(P) \otimes I & & \xrightarrow{\cong} & & \delta(P) \\
{\scriptstyle \delta(P)\otimes\nu}\downarrow & & & \nearrow{\scriptstyle \delta(\sigma)} & \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P) \otimes P) & &
\end{array}
$$

$$
\begin{array}{ccccc}
\delta(P) \otimes \delta(P) \otimes P & & \xrightarrow{\mathbf{str}' \otimes P} & & \delta(\delta(P) \otimes P) \otimes P \\
{\scriptstyle \delta(P)\otimes\sigma}\downarrow & & & & \downarrow{\scriptstyle \delta(\sigma)\otimes P} \\
\delta(P) \otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P) \otimes P
\end{array}
$$

$$
\begin{array}{ccccccccc}
\delta^2(P) \otimes P \otimes P & & & \xrightarrow{\mathbf{swap} \otimes P \otimes P} & \delta^2(P) \otimes P \otimes P & \xrightarrow{\delta^2(P) \otimes \cong} & & & \delta^2(P) \otimes P \otimes P \\
{\scriptstyle \mathbf{str}\otimes P}\downarrow & & & & & & & & \downarrow{\scriptstyle \mathbf{str}\otimes P} \\
\delta(\delta(P) \otimes P) \otimes P & \xrightarrow{\delta(\sigma)\otimes P} & \delta(P) \otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P) \otimes P & \xleftarrow{\delta(\sigma)\otimes P} & \delta(\delta(P) \otimes P) \otimes P
\end{array}
$$

$$
\begin{array}{ccccc}
\delta^2(P) \otimes \delta(P) \otimes P \otimes P & \xrightarrow{\cong} & \delta^2(P) \otimes P \otimes \delta(P) \otimes P & \xrightarrow{\mathbf{str}} & \delta(\delta(P) \otimes P) \otimes \delta(P) \otimes P \\
{\scriptstyle \Delta}\uparrow & & & & \downarrow{\scriptstyle \delta(\sigma)\otimes\sigma} \\
\delta^2(P) \otimes \delta(P) \otimes P & & & & \delta(P) \otimes P \\
{\scriptstyle \rho\otimes P}\downarrow & & & & \downarrow{\scriptstyle \sigma} \\
\delta(\delta(P) \otimes P) \otimes P & \xrightarrow[\delta(\sigma)\otimes P]{} & \delta(P) \otimes P & \xrightarrow[\sigma]{} & P
\end{array}
$$

Relevant Substitution Algebra

A relevant substitution algebra is a triple (P, σ, ν) such that:

$$\begin{array}{ccc} I\otimes P & \xrightarrow{\cong} & P \\ {\scriptstyle \nu\otimes P}\downarrow & \nearrow{\scriptstyle\sigma} & \\ \delta(P)\otimes P & & \end{array} \qquad \begin{array}{ccc} \delta(P)\otimes I & \xrightarrow{\cong} & \delta(P) \\ {\scriptstyle\delta(P)\otimes\nu}\downarrow & & \nearrow{\scriptstyle\delta(\sigma)} \\ \delta(P)\otimes\delta(P) & \xrightarrow{\mathbf{str}'} & \delta(\delta(P)\otimes P) \end{array}$$

$$\begin{array}{ccccc} \delta(P)\otimes\delta(P)\otimes P & & \xrightarrow{\mathbf{str}'\otimes P} & & \delta(\delta(P)\otimes P)\otimes P \\ {\scriptstyle\delta(P)\otimes\sigma}\downarrow & & & & \downarrow{\scriptstyle\delta(\sigma)\otimes P} \\ \delta(P)\otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P)\otimes P \end{array}$$

$$\begin{array}{ccccccccc} \delta^2(P)\otimes P\otimes P & & & \xrightarrow{\mathbf{swap}\otimes P\otimes P} & \delta^2(P)\otimes P\otimes P & \xrightarrow{\delta^2(P)\otimes\cong} & & & \delta^2(P)\otimes P\otimes P \\ {\scriptstyle\mathbf{str}\otimes P}\downarrow & & & & & & & & \downarrow{\scriptstyle\mathbf{str}\otimes P} \\ \delta(\delta(P)\otimes P)\otimes P & \xrightarrow{\delta(\sigma)\otimes P} & \delta(P)\otimes P & \xrightarrow{\sigma} & P & \xleftarrow{\sigma} & \delta(P)\otimes P & \xleftarrow{\delta(\sigma)\otimes P} & \delta(\delta(P)\otimes P)\otimes P \end{array}$$

$$\begin{array}{ccccc} \delta^2(P)\otimes\delta(P)\otimes P\otimes P & \xrightarrow{\cong} & \delta^2(P)\otimes P\otimes\delta(P)\otimes P & \xrightarrow{\mathbf{str}} & \delta(\delta(P)\otimes P)\otimes\delta(P)\otimes P \\ {\scriptstyle\Delta}\uparrow & & & & \downarrow{\scriptstyle\delta(\sigma)\otimes\sigma} \\ \delta^2(P)\otimes\delta(P)\otimes P & & & & \delta(P)\otimes P \\ {\scriptstyle\rho\otimes P}\downarrow & & & & \downarrow{\scriptstyle\sigma} \\ \delta(\delta(P)\otimes P)\otimes P & \xrightarrow[\delta(\sigma)\otimes P]{} & \delta(P)\otimes P & \xrightarrow[\sigma]{} & P \end{array}$$

$$\mathbf{RSubstAlg} \cong \mathrm{Mon}(\mathcal{S})$$

Binding Signatures

A binding signature is a pair $\Sigma = (\Omega, \alpha)$ where

Ω is a set of operations

$\alpha : \Omega \to \mathbb{N}^*$ is an arity map

Binding Signatures

A binding signature is a pair $\Sigma = (\Omega, \alpha)$ where
- Ω is a set of operations
- $\alpha : \Omega \to \mathbb{N}^*$ is an arity map

For $\omega \in \Omega$ with $\alpha(\omega) = (n_1, \ldots, n_k)$
- k is the usual arity $\rightsquigarrow$ how many inputs
- Each n_i say how many variables ω binds in the ith input

Binding Signatures

A binding signature is a pair $\Sigma = (\Omega, \alpha)$ where
Ω is a set of operations
$\alpha : \Omega \to \mathbb{N}^*$ is an arity map

For $\omega \in \Omega$ with $\alpha(\omega) = (n_1, \ldots, n_k)$
k is the usual arity $\rightsquigarrow$ how many inputs
Each n_i say how many variables ω binds in the ith input

eg. Lambda Calulus
$\Omega = \{\mathbf{abs}, \mathbf{app}\}$
$\alpha(\mathbf{abs}) = (1)$ $\rightsquigarrow$ 1 input, binding 1 variable
$\alpha(\mathbf{app}) = (0, 0)$ $\rightsquigarrow$ 2 inputs, no binding

Binding Signatures

A binding signature is a pair $\Sigma = (\Omega, \alpha)$ where
- Ω is a set of operations
- $\alpha : \Omega \to \mathbb{N}^*$ is an arity map

For $\omega \in \Omega$ with $\alpha(\omega) = (n_1, \ldots, n_k)$
- k is the usual arity $\rightsquigarrow$ how many inputs
- Each n_i say how many variables ω binds in the ith input

eg. Lambda Calulus
- $\Omega = \{\mathbf{abs}, \mathbf{app}\}$
- $\alpha(\mathbf{abs}) = (1) \rightsquigarrow$ 1 input, binding 1 variable
- $\alpha(\mathbf{app}) = (0, 0) \rightsquigarrow$ 2 inputs, no binding

Second-Order Theories
- Impose equations on the operations
- Requires meta-variables and meta-subtitution
- See Fiore, Mahmoud, Hur, Szamozvancev for cartesian case

Binding for Cartesian Syntax

Given some binding signature $\Sigma = (\Omega, \alpha)$, define

$$\Sigma(P) = \coprod_{\omega \in \Omega} \prod_{i \in [k]} \delta^{n_i}(P)$$

Binding for Cartesian Syntax

Given some binding signature $\Sigma = (\Omega, \alpha)$, define

$$\Sigma(P) = \coprod_{\omega \in \Omega} \prod_{i \in [k]} \delta^{n_i}(P)$$

eg. Lambda Calculus

$\Sigma(P) = \delta(P) + P \times P$

Binding for Cartesian Syntax

Given some binding signature $\Sigma = (\Omega, \alpha)$, define

$$\Sigma(P) = \coprod_{\omega \in \Omega} \prod_{i \in [k]} \delta^{n_i}(P)$$

eg. Lambda Calculus

$\Sigma(P) = \delta(P) + P \times P$

Does Σ inherit properties of δ?

Strength: Yes!

Symmetric Monad: No $\rightsquigarrow$ But Σ has **swap**

Symmetric Distributive Laws

A symmetric distributive law between a symmetric monad $(T, \mu, \eta, \varsigma)$ and an endofunctor S is

$$\tau : TS \to ST$$

$$\begin{array}{ccc} S & & \\ {\scriptstyle \eta S}\downarrow & \searrow^{S\eta} & \\ TS & \xrightarrow[\tau]{} & ST \end{array} \qquad \begin{array}{ccccc} TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\ {\scriptstyle \mu S}\downarrow & & & & \downarrow{\scriptstyle S\mu} \\ TS & & \xrightarrow[\tau]{} & & ST \end{array} \qquad \begin{array}{ccccc} TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\ {\scriptstyle \varsigma S}\downarrow & & & & \downarrow{\scriptstyle S\varsigma} \\ TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \end{array}$$

Symmetric Distributive Laws

A symmetric distributive law between a symmetric monad $(T, \mu, \eta, \varsigma)$ and an endofunctor S is

$$\tau : TS \to ST$$

$$\begin{array}{ccc} S & & \\ {\scriptstyle \eta S}\downarrow & \searrow^{S\eta} & \\ TS & \xrightarrow[\tau]{} & ST \end{array} \qquad \begin{array}{ccccc} TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\ {\scriptstyle \mu S}\downarrow & & & & \downarrow{\scriptstyle S\mu} \\ TS & & \xrightarrow[\tau]{} & & ST \end{array} \qquad \begin{array}{ccccc} TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\ {\scriptstyle \varsigma S}\downarrow & & & & \downarrow{\scriptstyle S\varsigma} \\ TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \end{array}$$

For symmetric monad $(T, \mu, \eta, \varsigma)$, ς is a symmetric distributive law between T and itself

swap : $\delta^2 \to \delta^2$ symmetric distributive law

Symmetric Distributive Laws

A symmetric distributive law between a symmetric monad $(T, \mu, \eta, \varsigma)$ and an endofunctor S is

$$\tau : TS \to ST$$

$$
\begin{array}{ccc}
S & & \\
{\scriptstyle \eta S}\downarrow & \searrow^{S\eta} & \\
TS & \xrightarrow[\tau]{} & ST
\end{array}
\qquad
\begin{array}{ccccc}
TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\
{\scriptstyle \mu S}\downarrow & & & & \downarrow{\scriptstyle S\mu} \\
TS & & \xrightarrow[\tau]{} & & ST
\end{array}
\qquad
\begin{array}{ccccc}
TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\
{\scriptstyle \varsigma S}\downarrow & & & & \downarrow{\scriptstyle S\varsigma} \\
TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT
\end{array}
$$

For symmetric monad $(T, \mu, \eta, \varsigma)$, ς is a symmetric distributive law between T and itself

- **swap** $: \delta^2 \to \delta^2$ symmetric distributive law

For a endofunctor with (cartesian) strength $(F, \mathbf{str})$

- $- \times Q$ is a symmetric comonad
- $\mathbf{str}_{-,Q} : (- \times Q)F \to F(- \times Q)$ is a symmetric codistributive law

Lifting Symmetric Distributive Laws

Let T oplax monoidal symmetric monad and G_1 and G_2 endofunctors and $\psi_1 : TG_1 \to G_1T$ and $\psi_2 : TG_2 \to G_2T$ symmetric distributive laws , then

$$\tilde{\psi} : T(G_1 \times G_2) \to TG_1 \times TG_2 \to G_1T \times G_2T = (G_1 \times G_2)T$$

is a symmetric distributive law

Lifting Symmetric Distributive Laws

Let T oplax monoidal symmetric monad and G_1 and G_2 endofunctors and $\psi_1 : TG_1 \to G_1T$ and $\psi_2 : TG_2 \to G_2T$ symmetric distributive laws , then

$$\tilde{\psi} : T(G_1 \times G_2) \to TG_1 \times TG_2 \to G_1T \times G_2T = (G_1 \times G_2)T$$

is a symmetric distributive law

δ cartesian monoidal and cocartesian monoidal

swap : $\delta\Sigma \to \Sigma\delta$

Lifting Symmetric Distributive Laws

Let T oplax monoidal symmetric monad and G_1 and G_2 endofunctors and $\psi_1 : TG_1 \to G_1T$ and $\psi_2 : TG_2 \to G_2T$ symmetric distributive laws , then

$$\tilde{\psi} : T(G_1 \times G_2) \to TG_1 \times TG_2 \to G_1T \times G_2T = (G_1 \times G_2)T$$

is a symmetric distributive law

δ cartesian monoidal and cocartesian monoidal

$\textbf{swap} : \delta\Sigma \to \Sigma\delta$

$- \times Q$ oplax cartesian monoidal and cocartesian monoidal

$\textbf{str} : \Sigma(P) \times Q \to \Sigma(P \times Q)$

Abstract Syntax of a Binding Signature

Σ-algebra: (P, h) where $h : \Sigma(P) \to P$
Σ-**Alg**: Category of Σ-algebras and homomorphisms

Abstract Syntax of a Binding Signature

Σ-algebra: (P, h) where $h : \Sigma(P) \to P$
Σ-**Alg**: Category of Σ-algebras and homomorphisms

Forgetful functor: $U : \Sigma\text{-}\mathbf{Alg} \to \mathcal{F}$

Abstract Syntax of a Binding Signature

Σ-algebra: (P, h) where $h : \Sigma(P) \to P$
Σ-**Alg**: Category of Σ-algebras and homomorphisms

Forgetful functor: $U : \Sigma\text{-}\mathbf{Alg} \to \mathcal{F}$

Free functor: $F : \mathcal{F} \to \Sigma\text{-}\mathbf{Alg}$
Left adjoint to U
For P in $\mathcal{F}$, consider the endofunctor $P + \Sigma : \mathcal{F} \to \mathcal{F}$
Let $[\eta_P, \varphi_P] : P + \Sigma(TP) \to TP$ be the initial $(P + \Sigma)$-algebra
Lambek: $[\eta_P, \varphi_P]$ isomorphism
Adamek: $F(P) = (TP, \varphi_P)$

Abstract Syntax of a Binding Signature

Σ-algebra: (P, h) where $h : \Sigma(P) \to P$
Σ-**Alg**: Category of Σ-algebras and homomorphisms

Forgetful functor: $U : \Sigma\text{-}\mathbf{Alg} \to \mathcal{F}$

Free functor: $F : \mathcal{F} \to \Sigma\text{-}\mathbf{Alg}$
Left adjoint to U
For P in $\mathcal{F}$, consider the endofunctor $P + \Sigma : \mathcal{F} \to \mathcal{F}$
Let $[\eta_P, \varphi_P] : P + \Sigma(TP) \to TP$ be the initial $(P + \Sigma)$-algebra
Lambek: $[\eta_P, \varphi_P]$ isomorphism
Adamek: $F(P) = (TP, \varphi_P)$

Abstract Syntax: Terms freely generated by operations and variables
Initial $(V + \Sigma)$-algebra $\rightsquigarrow [\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

Abstract Syntax of a Binding Signature

Σ-algebra: (P, h) where $h : \Sigma(P) \to P$
Σ-**Alg**: Category of Σ-algebras and homomorphisms

Forgetful functor: $U : \Sigma\text{-}\mathbf{Alg} \to \mathcal{F}$

Free functor: $F : \mathcal{F} \to \Sigma\text{-}\mathbf{Alg}$
Left adjoint to U
For P in $\mathcal{F}$, consider the endofunctor $P + \Sigma : \mathcal{F} \to \mathcal{F}$
Let $[\eta_P, \varphi_P] : P + \Sigma(TP) \to TP$ be the initial $(P + \Sigma)$-algebra
Lambek: $[\eta_P, \varphi_P]$ isomorphism
Adamek: $F(P) = (TP, \varphi_P)$

Abstract Syntax: Terms freely generated by operations and variables
Initial $(V + \Sigma)$-algebra $\rightsquigarrow [\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

Want: A cartesian substitution algebra structure on TV

Generalised Recursion

Syntactically: Substitution defined recursively

Generalised Recursion

Syntactically: Substitution defined recursively

Categorically:
Universal property of initial algebra = structural recursion

Bird-Patterson (1999): Very general generalisation
Matthes-Uustalu (2004): More appropriate specialisation

Generalised Recursion

Syntactically: Substitution defined recursively

Categorically:
Universal property of initial algebra = structural recursion

Bird-Patterson (1999): Very general generalisation
Matthes-Uustalu (2004): More appropriate specialisation

Given adjoint endofunctors $F \dashv G$ and endofunctors S and S'
with initial S-algebra $\alpha : S(A) \to A$ and S'-algebra $\beta : S'(B) \to B$,
and a natural transformation $\psi : FS \to S'F$
there exists a unique $f : F(A) \to B$ such that

$$\begin{array}{ccc}
S'(F) & \xrightarrow{S'(f)} & S'(B) \\
{\scriptstyle \psi_A}\uparrow & & \\
FS(A) & & \downarrow{\scriptstyle \beta} \\
{\scriptstyle F(\alpha)}\downarrow{\scriptstyle \cong} & & \\
F(A) & \overset{f}{\dashrightarrow} & B
\end{array}$$

Substitution for Abstract Syntax

Abstract Syntax: TV in $\mathcal{F}$, with $\eta_V : V \to TV$ and $\varphi_V : \Sigma(TV) \to TV$

Substitution for Abstract Syntax

Abstract Syntax: TV in $\mathcal{F}$, with $\eta_V : V \to TV$ and $\varphi_V : \Sigma(TV) \to TV$

Variables: $\nu : 1 \to \delta(TV)$ is transpose of η_V

Substitution for Abstract Syntax

Abstract Syntax: TV in $\mathcal{F}$, with $\eta_V : V \to TV$ and $\varphi_V : \Sigma(TV) \to TV$

Variables: $\nu : 1 \to \delta(TV)$ is transpose of η_V

Basic Substitution: Substituting into a variable

Note: $\delta(V) \cong \mathcal{Y}(1, - + 1) \cong V + 1$

$$\beta : \delta(V) \times TV \cong V \times TV + 1 \times TV \xrightarrow{[\eta_V \pi_1, \pi_2]} TV$$

Substitution for Abstract Syntax

Abstract Syntax: TV in $\mathcal{F}$, with $\eta_V : V \to TV$ and $\varphi_V : \Sigma(TV) \to TV$

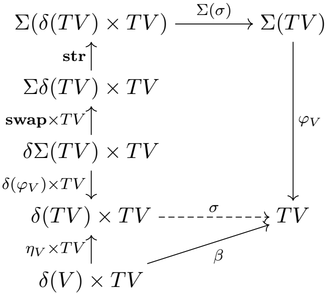
Variables: $\nu : 1 \to \delta(TV)$ is transpose of η_V

Basic Substitution: Substituting into a variable

Note: $\delta(V) \cong \mathcal{Y}(1, - + 1) \cong V + 1$

$$\beta : \delta(V) \times TV \cong V \times TV + 1 \times TV \xrightarrow{[\eta_V \pi_1, \pi_2]} TV$$

Substitution:



Substitution for Abstract Syntax

Abstract Syntax: TV in $\mathcal{F}$, with $\eta_V : V \to TV$ and $\varphi_V : \Sigma(TV) \to TV$

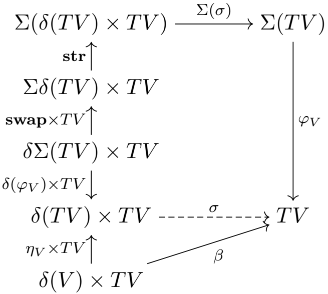
Variables: $\nu : 1 \to \delta(TV)$ is transpose of η_V

Basic Substitution: Substituting into a variable

Note: $\delta(V) \cong \mathcal{Y}(1, - + 1) \cong V + 1$

$$\beta : \delta(V) \times TV \cong V \times TV + 1 \times TV \xrightarrow{[\eta_V \pi_1, \pi_2]} TV$$

Substitution:



Fiore-Plotkin-Turi (1999): (TV, σ, ν) is a cartesian substitution algebra

Substitution for Abstract Syntax

Abstract Syntax: TV in $\mathcal{F}$, with $\eta_V : V \to TV$ and $\varphi_V : \Sigma(TV) \to TV$

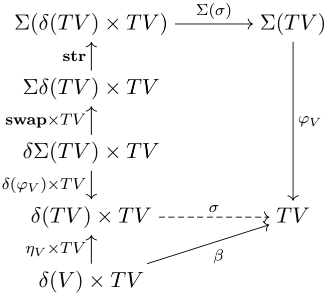
Variables: $\nu : 1 \to \delta(TV)$ is transpose of η_V

Basic Substitution: Substituting into a variable

Note: $\delta(V) \cong \mathcal{Y}(1, - + 1) \cong V + 1$

$$\beta : \delta(V) \times TV \cong V \times TV + 1 \times TV \xrightarrow{[\eta_V \pi_1, \pi_2]} TV$$

Substitution:



Fiore-Plotkin-Turi (1999): (TV, σ, ν) is a cartesian substitution algebra

Fiore-Plotkin-Turi (1999): $(TV, \sigma, \nu, \varphi_V)$ is initial **Σ-CSubstAlg**

The Linear Case

Binding Signature Endofunctor on $\mathcal{B}$:

$$\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$$

The Linear Case

Binding Signature Endofunctor on $\mathcal{B}$:

$$\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$$

Σ-algebras:

$$\Sigma\text{-}\mathbf{Alg} \underset{U}{\overset{F}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{B}$$

The Linear Case

Binding Signature Endofunctor on $\mathcal{B}$:

$$\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$$

Σ-algebras:

$$\Sigma\text{-}\mathbf{Alg} \underset{U}{\overset{F}{\underset{\longrightarrow}{\overset{\longleftarrow}{\perp}}}} \mathcal{B}$$

Abstract Syntax: $[\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

The Linear Case

Binding Signature Endofunctor on $\mathcal{B}$:

$$\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$$

Σ-algebras:

$$\Sigma\text{-}\mathbf{Alg} \underset{U}{\overset{F}{\underset{\perp}{\leftrightarrows}}} \mathcal{B}$$

Abstract Syntax: $[\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

Problem:

δ and $- \otimes Q$ are not (oplax) $\otimes$-monoidal

So we can't lift **swap** and **str** to Σ

The Linear Case

Binding Signature Endofunctor on $\mathcal{B}$:

$$\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$$

Σ-algebras:

$$\Sigma\text{-}\mathbf{Alg} \underset{U}{\overset{F}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{B}$$

Abstract Syntax: $[\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

Problem:

δ and $- \otimes Q$ are not (oplax) $\otimes$-monoidal
So we can't lift **swap** and **str** to Σ

Leibniz Rule: $\mathcal{L} : \delta(P \otimes Q) \cong \delta(P) \otimes Q + P \otimes \delta(Q)$

Derived Signature Functors

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P)$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P)$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

$$\delta(P) + P{\otimes}P$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

$$\delta(\delta(P) + P\otimes P)$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

$$\delta\delta(P) + \delta(P \otimes P)$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

$$\delta\delta(P) + \delta(P) \otimes P + P \otimes \delta(P)$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

$$\delta\delta(P) + \delta(P)\otimes P + P\otimes\delta(P)$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

$$\delta(Q) + Q{\otimes}P + P{\otimes}Q$$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^{\dagger}(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

$$\delta(Q) + Q{\otimes}P + P{\otimes}Q$$

Derived Signature Endofunctor: $\Sigma^{\dagger} : \mathcal{B} \times \mathcal{B} \to \mathcal{B}$

Derived Signature Functors

$$\delta\Sigma(P) = \delta \coprod_{\omega\in\Omega} \bigotimes_{i\in[k]} \delta^{n_i}(P) \cong \coprod_{\omega\in\Omega} \delta \bigotimes_{i\in[k]} \delta^{n_i}(P) \xrightarrow{\mathbf{swap}\mathcal{L}} \Sigma^\dagger(P, \delta(P))$$

eg. Lambda Calculus: $\Sigma(P) = \delta(P) + P \otimes P$

$$\delta(Q) + Q{\otimes}P + P{\otimes}Q$$

Derived Signature Endofunctor: $\Sigma^\dagger : \mathcal{B} \times \mathcal{B} \to \mathcal{B}$

By definition: $\mathbf{swap} : \delta\Sigma(P) \cong \Sigma^\dagger(P, \delta(P))$

$\Sigma^\dagger$ linear in Q: $\mathbf{str} : \Sigma^\dagger(P, Q) \otimes R \to \Sigma^\dagger(P, Q \otimes R)$

Every $h : \Sigma(P) \to P$ induces $h^\dagger : \Sigma^\dagger(P) = \Sigma^\dagger(P, P) \to P$

Substitution for Linear Abstract Syntax

Variables: $\nu : I \to \delta(TV)$ transpose of $\eta_V : V \to TV$

Substitution for Linear Abstract Syntax

Variables: $\nu : I \to \delta(TV)$ transpose of $\eta_V : V \to TV$

Basic Substitution:

Note: $\delta(V) \cong I$

$\beta : \delta(V) \otimes TV \to TV$

Substitution for Linear Abstract Syntax

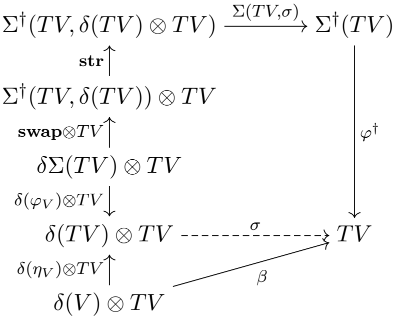
Variables: $\nu : I \to \delta(TV)$ transpose of $\eta_V : V \to TV$

Basic Substitution:

Note: $\delta(V) \cong I$

$\beta : \delta(V) \otimes TV \to TV$

Substitution:



Substitution for Linear Abstract Syntax

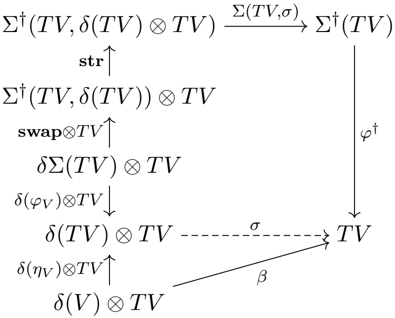
Variables: $\nu : I \to \delta(TV)$ transpose of $\eta_V : V \to TV$

Basic Substitution:

Note: $\delta(V) \cong I$

$\beta : \delta(V) \otimes TV \to TV$

Substitution:



(TV, σ, ν) is a linear substitution algebra

$(TV, \sigma, \nu, \varphi_V)$ is initial Σ-**LSubstAlg**

Affine and Relevant Cases

Binding Signature Endofunctor: $\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$

Abstract Syntax: $[\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

Affine and Relevant Cases

Binding Signature Endofunctor: $\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$

Abstract Syntax: $[\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

Leibniz Rules:

Affine $\mathcal{I}$: $\delta(P \otimes Q) \cong \delta(P) \otimes Q + P \otimes \delta(Q) + P \otimes Q$

Relevant $\mathcal{S}$: $\delta(P \otimes Q) \cong \delta(P) \otimes Q + P \otimes \delta(Q) + \delta(P) \otimes \delta(Q)$

Affine and Relevant Cases

Binding Signature Endofunctor: $\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$

Abstract Syntax: $[\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

Leibniz Rules:

Affine $\mathcal{I}$: $\delta(P \otimes Q) \cong \delta(P) \otimes Q + P \otimes \delta(Q) + P \otimes Q$

Relevant $\mathcal{S}$: $\delta(P \otimes Q) \cong \delta(P) \otimes Q + P \otimes \delta(Q) + \delta(P) \otimes \delta(Q)$

Derived Signature Endofunctors: Appropriately defined in each case

By definition: $\mathbf{swap} : \delta\Sigma(P) \cong \Sigma^\dagger(P, \delta(P))$

Day convolution structure: $\mathbf{str} : \Sigma^\dagger(P, Q) \otimes R \to \Sigma^\dagger(P, Q \otimes R)$

Affine and Relevant Cases

Binding Signature Endofunctor: $\Sigma(P) = \coprod_{\omega \in \Omega} \bigotimes_{i \in [k]} \delta^{n_i}(P)$

Abstract Syntax: $[\eta_V, \varphi_V] : V + \Sigma(TV) \to TV$

Leibniz Rules:

Affine $\mathcal{I}$: $\delta(P \otimes Q) \cong \delta(P) \otimes Q + P \otimes \delta(Q) + P \otimes Q$

Relevant $\mathcal{S}$: $\delta(P \otimes Q) \cong \delta(P) \otimes Q + P \otimes \delta(Q) + \delta(P) \otimes \delta(Q)$

Derived Signature Endofunctors: Appropriately defined in each case

By definition: $\mathbf{swap} : \delta\Sigma(P) \cong \Sigma^\dagger(P, \delta(P))$

Day convolution structure: $\mathbf{str} : \Sigma^\dagger(P, Q) \otimes R \to \Sigma^\dagger(P, Q \otimes R)$

Substitution Algebras: Induced as before

Affine $\mathcal{I}$: $(TV, \sigma, \nu, \varphi_V)$ initial Σ-**ASubstAlg**

Relevant $\mathcal{S}$: $(TV, \sigma, \nu, \varphi_V)$ initial Σ-**RSubstAlg**

Towards Uniform Axiomatisation

Substitution: $\sigma : \delta(P) \otimes P \to P$

Substitution Signature: $\Sigma_{\text{sub}}(P) = \delta(P) \otimes P$

$\Sigma^{\dagger}_{\text{sub}}$ has **swap** and **str**

We can use this to rewrite the axioms of a substitution algebra

Towards Uniform Axiomatisation

Substitution: $\sigma : \delta(P) \otimes P \to P$

Substitution Signature: $\Sigma_{\text{sub}}(P) = \delta(P) \otimes P$

$\Sigma^{\dagger}_{\text{sub}}$ has **swap** and **str**

We can use this to rewrite the axioms of a substitution algebra

A substitution algebra is a triple $(P, \sigma : \Sigma_{\text{sub}}(P) \to P, \nu : I \to P)$ such that

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\downarrow & \nearrow_{\sigma} & \\
\Sigma_{\text{sub}}(P) & &
\end{array}
\qquad
\begin{array}{ccccc}
\delta(P) \otimes I & & \xrightarrow{\cong} & & \delta(P) \\
{\scriptstyle \delta(P)\otimes\nu}\downarrow & & & \nearrow_{\delta(\sigma)} & \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta\Sigma_{\text{sub}}(P) & &
\end{array}
$$

$$
\begin{array}{ccccccc}
\delta\Sigma_{\text{sub}}(P) \otimes P & \xrightarrow{\mathbf{swap}\otimes P} & \Sigma^{\dagger}_{\text{sub}}(P, \delta(P)) \otimes P & \xrightarrow{\mathbf{str}} & \Sigma^{\dagger}_{\text{sub}}(P, \Sigma_{\text{sub}}(P)) & \xrightarrow{\Sigma^{\dagger}_{\text{sub}}(P,\sigma)} & \Sigma^{\dagger}_{\text{sub}}(P) \\
{\scriptstyle \delta(\sigma)\otimes P}\downarrow & & & & & & \downarrow{\scriptstyle \sigma^{\dagger}} \\
\Sigma_{\text{sub}}(P) & & & \xrightarrow{\sigma} & & & P
\end{array}
$$

Towards Uniform Axiomatisation

Substitution: $\sigma : \delta(P) \otimes P \to P$

Substitution Signature: $\Sigma_{\text{sub}}(P) = \delta(P) \otimes P$

$\Sigma^{\dagger}_{\text{sub}}$ has **swap** and **str**

We can use this to rewrite the axioms of a substitution algebra

A substitution algebra is a triple $(P, \sigma : \Sigma_{\text{sub}}(P) \to P, \nu : I \to P)$ such that

$$
\begin{array}{ccc}
I \otimes P & \xrightarrow{\cong} & P \\
{\scriptstyle \nu \otimes P}\downarrow & \nearrow{\scriptstyle \sigma} & \\
\Sigma_{\text{sub}}(P) & &
\end{array}
\qquad
\begin{array}{ccccc}
\delta(P) \otimes I & & \xrightarrow{\cong} & & \delta(P) \\
{\scriptstyle \delta(P)\otimes\nu}\downarrow & & & \nearrow{\scriptstyle \delta(\sigma)} & \\
\delta(P) \otimes \delta(P) & \xrightarrow{\mathbf{str}'} & \delta\Sigma_{\text{sub}}(P) & &
\end{array}
$$

$$
\begin{array}{ccccccc}
\delta\Sigma_{\text{sub}}(P) \otimes P & \xrightarrow{\mathbf{swap}\otimes P} & \Sigma^{\dagger}_{\text{sub}}(P, \delta(P)) \otimes P & \xrightarrow{\mathbf{str}} & \Sigma^{\dagger}_{\text{sub}}(P, \Sigma_{\text{sub}}(P)) & \xrightarrow{\Sigma^{\dagger}_{\text{sub}}(P,\sigma)} & \Sigma^{\dagger}_{\text{sub}}(P) \\
{\scriptstyle \delta(\sigma)\otimes P}\downarrow & & & & & & \downarrow{\scriptstyle \sigma^{\dagger}} \\
\Sigma_{\text{sub}}(P) & & & \xrightarrow{\sigma} & & & P
\end{array}
$$

This works for Linear, Affine and Relevant, but not for Cartesian!