

Single-Variable Substitution for Substructural Theories

Sanjiv Ranchod

(joint work with Marcelo Fiore)

University of Cambridge

IRIF, Paris, France

Syntax as Presheaves

Simply-typed single-sorted syntax

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

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)

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

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

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: $n \rightarrow m$ is a function $[n] \rightarrow [m]$, where $[n] = \{1, \dots, n\}$

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: $n \rightarrow m$ is a function $[n] \rightarrow [m]$, where $[n] = \{1, \dots, n\}$

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: $n \rightarrow m$ is a function $[n] \rightarrow [m]$, where $[n] = \{1, \dots, n\}$

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}

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

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

V is the cartesian syntax with only variables as terms

Simultaneous Substitution for Cartesian Syntax

Presheaf of Variables: V in \mathcal{F}

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

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

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}

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

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

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:

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

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}

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

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

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:

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

μ performs substitution

η specifies variables

unit laws express substituting with/into variables

associativity law expresses the substitution lemma

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

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:

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

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:

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

Operations correspond to structural rules:

Weakening \rightsquigarrow unit

Contraction \rightsquigarrow multiplication

Exchange \rightsquigarrow symmetric monoidal structure

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

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

$\text{Mod}(\mathfrak{F}, \mathcal{E}) \cong \mathbf{SM}(\mathbb{F}, \mathcal{E})$

The Approach of Superspecies

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

Cartesian Case:

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

Operations correspond to structural rules:

Weakening \rightsquigarrow unit

Contraction \rightsquigarrow multiplication

Exchange \rightsquigarrow symmetric monoidal structure

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

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

$\text{Mod}(\mathfrak{F}, \mathcal{E}) \cong \mathbf{SM}(\mathbb{F}, \mathcal{E})$

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

Linear, Affine and Relevant Syntax

Linear Case: Exchange

\mathfrak{B} : L no equations

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

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

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

Linear, Affine and Relevant Syntax

Linear Case: Exchange

$\mathfrak{B} : \quad L \quad \text{no equations}$

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

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

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

Affine Case: Weakening and Exchange

$\mathfrak{J} : \quad I \longrightarrow C \quad \text{no equations}$

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

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

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

Linear, Affine and Relevant Syntax

Linear Case: Exchange

$\mathfrak{B} : L \quad \text{no equations}$

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

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

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

Affine Case: Weakening and Exchange

$\mathfrak{J} : I \longrightarrow C \quad \text{no equations}$

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

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

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

Relevant Case: Contraction and Exchange

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

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

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

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

Single-Variable Substitution

Single-Variable Substitution

Syntactic Cartesian Substitution:

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

Single-Variable Substitution

Syntactic Cartesian Substitution:

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

For a presheaf P in \mathcal{F} :

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

Single-Variable Substitution

Syntactic Cartesian Substitution:

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

For a presheaf P in \mathcal{F} :

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

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

Single-Variable Substitution

Syntactic Cartesian Substitution:

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

For a presheaf P in \mathcal{F} :

An operation $P(n+1) \times P(n) \rightarrow P(n)$ 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

$$\mu : A \otimes A \rightarrow A \quad \eta : I \rightarrow A \quad \varsigma : A \otimes A \rightarrow A \otimes A$$

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

$$\mu : A \otimes A \rightarrow A \quad \eta : I \rightarrow A \quad \varsigma : A \otimes A \rightarrow A \otimes A$$

such that the following commute:

$$\begin{array}{ccc} A^{\otimes 3} & \xrightarrow{\mu \otimes A} & A^{\otimes 2} \\ A \otimes \mu \downarrow & & \downarrow \mu \\ A^{\otimes 2} & \xrightarrow{\mu} & A \end{array} \quad \begin{array}{ccc} A & \xrightarrow{A \otimes \eta} & A^{\otimes 2} \\ \eta \otimes A \downarrow & \searrow A & \downarrow \mu \\ A^{\otimes 2} & \xrightarrow{\mu} & A \end{array}$$

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

$$\mu : A \otimes A \rightarrow A \quad \eta : I \rightarrow A \quad \varsigma : A \otimes A \rightarrow A \otimes A$$

such that the following commute:

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

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

$$\mu : A \otimes A \rightarrow A \quad \eta : I \rightarrow A \quad \varsigma : A \otimes A \rightarrow A \otimes A$$

such that the following commute:

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

$$\begin{array}{ccc}
 A^{\otimes 3} \xrightarrow{\varsigma \otimes A} A^{\otimes 3} & A^{\otimes 3} \xrightarrow{A \otimes \varsigma} A^{\otimes 3} & \\
 A \otimes \varsigma \downarrow & \downarrow \varsigma \otimes A & \\
 A^{\otimes 3} \xrightarrow{\varsigma \otimes A} A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} A^{\otimes 3} &
 \end{array}$$

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

$$\mu : A \otimes A \rightarrow A \quad \eta : I \rightarrow A \quad \varsigma : A \otimes A \rightarrow A \otimes A$$

such that the following commute:

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

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

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: $(1, c : 2 \rightarrow 1, w : 0 \rightarrow 1, s : 2 \rightarrow 2)$

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: $(1, c : 2 \rightarrow 1, w : 0 \rightarrow 1, s : 2 \rightarrow 2)$

Operations directly correspond to structural rules

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

$$- + c : - + 2 \rightarrow - + 1$$

$$- + w : - + 0 \rightarrow - + 1$$

$$- + s : - + 2 \rightarrow - + 2$$

Context Extension

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

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

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

Context Extension

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

$$\begin{array}{ccc} \mathbb{F}^{\text{op}} & \xrightarrow{\mathcal{Y}} & \mathcal{F} \\ (-+1)^{\text{op}} \downarrow & & \uparrow \delta \\ \mathbb{F}^{\text{op}} & \xrightarrow{\mathcal{Y}} & \mathcal{F} \end{array} \quad \delta(P)(n) = P(n+1)$$

δ is a symmetric monad:

$$\mathbf{cont}_{P,n} = P(n+c) : \delta^2(P)(n) \rightarrow \delta(P)(n)$$

$$\mathbf{weak}_{P,n} = P(n+w) : P(n) \rightarrow \delta(P)(n)$$

$$\mathbf{swap}_{P,n} = P(n+s) : \delta^2(P)(n) \rightarrow \delta^2(P)(n)$$

Context Extension

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

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

δ is a symmetric monad:

$$\mathbf{cont}_{P,n} = P(n+c) : \delta^2(P)(n) \rightarrow \delta(P)(n)$$

$$\mathbf{weak}_{P,n} = P(n+w) : P(n) \rightarrow \delta(P)(n)$$

$$\mathbf{swap}_{P,n} = P(n+s) : \delta^2(P)(n) \rightarrow \delta^2(P)(n)$$

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

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} :

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

δ is a symmetric monad:

$$\mathbf{cont}_{P,n} = P(n+c) : \delta^2(P)(n) \rightarrow \delta(P)(n)$$

$$\mathbf{weak}_{P,n} = P(n+w) : P(n) \rightarrow \delta(P)(n)$$

$$\mathbf{swap}_{P,n} = P(n+s) : \delta^2(P)(n) \rightarrow \delta^2(P)(n)$$

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

Left and right adjoints given by Yoneda extensions

δ is cartesian and cocartesian monoidal

$$\delta \cong (-)^V$$

δ is a strong symmetric monad:

$$\mathbf{str} : \delta(P) \times Q \rightarrow \delta(P \times Q)$$

Strength respects the symmetric monad structure

Cartesian Substitution Algebra

Need: Variables and Substitution

Cartesian Substitution Algebra

Need: Variables and Substitution

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

Cartesian Substitution Algebra

Need: Variables and Substitution

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

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

Cartesian Substitution Algebra

Need: Variables and Substitution

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

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

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

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

Cartesian Substitution Algebra

Need: Variables and Substitution

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

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

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

$$\begin{array}{ccc} 1 \times P & \xrightarrow{\cong} & P \\ \nu \times P \downarrow & \nearrow \sigma & \uparrow \\ \delta(P) \times P & & \end{array} \quad \begin{array}{ccc} \delta^2(P) \times 1 & \xrightarrow{\cong} & \delta^2(P) \\ \delta^2(P) \times \nu \downarrow & & \mathbf{cont}_P \downarrow \\ \delta^2(P) \times \delta(P) & \xrightarrow{\cong} & \delta(\delta(P) \times P) \xrightarrow{\delta(\sigma)} \delta(P) \end{array}$$

Cartesian Substitution Algebra

Need: Variables and Substitution

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

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

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

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

Cartesian Substitution Algebra

Need: Variables and Substitution

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

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

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

$$\begin{array}{ccccc}
 1 \times P & \xrightarrow{\cong} & P & & \delta^2(P) \times 1 & \xrightarrow{\cong} & \delta^2(P) & & P \times P & \xrightarrow{\pi_1} & P \\
 \nu \times P \downarrow & & \nearrow \sigma & & \delta^2(P) \times \nu \downarrow & & \text{cont}_P \downarrow & & \text{weak}_{P \times P} \downarrow & & \nearrow \sigma \\
 \delta(P) \times P & & & & \delta^2(P) \times \delta(P) & \xrightarrow{\cong} & \delta(\delta(P) \times P) & \xrightarrow{\delta(\sigma)} & \delta(P) & & \delta(P) \times P
 \end{array}$$

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

Cartesian Substitution Algebra

Need: Variables and Substitution

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

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

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

$$\begin{array}{ccccc}
 1 \times P & \xrightarrow{\cong} & P & & \delta^2(P) \times 1 & \xrightarrow{\cong} & \delta^2(P) & & P \times P & \xrightarrow{\pi_1} & P \\
 \nu \times P \downarrow & & \nearrow \sigma & & \delta^2(P) \times \nu \downarrow & & \text{cont}_P \downarrow & & \text{weak}_{P \times P} \downarrow & & \nearrow \sigma \\
 \delta(P) \times P & & & & \delta^2(P) \times \delta(P) & \xrightarrow{\cong} & \delta(\delta(P) \times P) & \xrightarrow{\delta(\sigma)} & \delta(P) & & \delta(P) \times P
 \end{array}$$

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

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

General Approach

SMEqP: \mathfrak{e}

General Approach

SMEqP: \mathfrak{e}

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

General Approach

SMEqP: \mathfrak{C}

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

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

General Approach

SMEqP: \mathfrak{C}

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

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

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

General Approach

SMEqP: \mathfrak{C}

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

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

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

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

General Approach

SMEqP: \mathfrak{C}

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

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

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

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

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

General Approach

SMEqP: \mathfrak{C}

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

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

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

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

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

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

General Approach

SMEqP: \mathfrak{C}

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

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

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

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

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

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

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

General Approach

SMEqP: \mathfrak{C}

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

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

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

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

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

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

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

But what about the axioms?

The Linear Case

Monoidal Presentation: Symmetric Object (A, ς) with

$$\begin{array}{ccccc} A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\ & \searrow & \downarrow \varsigma & & A^{\otimes 3} \downarrow & & & & \downarrow \varsigma \otimes A \\ & & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \end{array}$$

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

The Linear Case

Monoidal Presentation: Symmetric Object (A, ς) with

$$\begin{array}{ccccc} A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\ & \searrow & \downarrow \varsigma & & A^{\otimes 3} \downarrow & & & & \downarrow \varsigma \otimes A \\ & & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \end{array}$$

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

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

The Linear Case

Monoidal Presentation: Symmetric Object (A, ς) with

$$\begin{array}{ccc} A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} \\ & \searrow & \downarrow \varsigma \\ & A^{\otimes 2} & \end{array} \quad \begin{array}{ccccc} A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\ A^{\otimes 3} \downarrow & & & & \downarrow \varsigma \otimes A \\ A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \end{array}$$

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

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

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

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

Linear Substitution Algebra

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

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

Linear Substitution Algebra

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

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

$$\begin{array}{ccc}
 \delta(P) \otimes \delta(P) \otimes P & \xrightarrow{\text{str}' \otimes P} & \delta(\delta(P) \otimes P) \otimes P \\
 \delta(P) \otimes \sigma \downarrow & & \downarrow \delta(\sigma) \otimes P \\
 \delta(P) \otimes P & \xrightarrow{\sigma} & P \xleftarrow{\sigma} \delta(P) \otimes P
 \end{array}$$

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

Linear Substitution Algebra

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

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

$$\begin{array}{ccc}
 \delta(P) \otimes \delta(P) \otimes P & \xrightarrow{\text{str}' \otimes P} & \delta(\delta(P) \otimes P) \otimes P \\
 \delta(P) \otimes \sigma \downarrow & & \downarrow \delta(\sigma) \otimes P \\
 \delta(P) \otimes P & \xrightarrow{\sigma} & P \xleftarrow{\sigma} \delta(P) \otimes P
 \end{array}$$

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

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

The Affine Case

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

$$\begin{array}{ccccc}
 A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} & & A & \xrightarrow{\eta \otimes A} & A^{\otimes 2} \\
 & \searrow & \downarrow \varsigma & & A \otimes \varsigma \downarrow & & & & \downarrow \varsigma \otimes A & & A \otimes \eta \searrow & & \downarrow \varsigma \\
 & & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} & & & & A^{\otimes 2}
 \end{array}$$

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

$$\begin{array}{ccccc}
 A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} & & A & \xrightarrow{\eta \otimes A} & A^{\otimes 2} \\
 & \searrow & & & & & & & & & & \searrow & & \downarrow \varsigma \\
 & & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} & & & & & & & A^{\otimes 2} \\
 & & & & & & & & & & & & & & \downarrow \varsigma \\
 & & & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} & & & & & & A^{\otimes 2} \\
 & & & & & & & & & & & & & & & \downarrow \varsigma \\
 & & & & & & & & & & & & & & & A^{\otimes 2}
 \end{array}$$

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
 $(1, w : 0 \rightarrow 1, s : 2 \rightarrow 2)$

The Affine Case

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

$$\begin{array}{ccccc}
 A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} & & A & \xrightarrow{\eta \otimes A} & A^{\otimes 2} \\
 & \searrow^{A^{\otimes 2}} & \downarrow \varsigma & & A^{\otimes 3} & \downarrow^{A \otimes \varsigma} & & & \downarrow \varsigma \otimes A & & A & \searrow^{A \otimes \eta} & \downarrow \varsigma \\
 & & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} & & & & A^{\otimes 2}
 \end{array}$$

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
 $(1, w : 0 \rightarrow 1, s : 2 \rightarrow 2)$

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:

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

Affine Substitution Algebra

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

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

Affine Substitution Algebra

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

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

$$\begin{array}{ccc}
 \delta(P) \otimes \delta(P) \otimes P \xrightarrow{\text{str}' \otimes P} \delta(\delta(P) \otimes P) \otimes P & & \\
 \delta(P) \otimes \sigma \downarrow & & \downarrow \delta(\sigma) \otimes P \\
 \delta(P) \otimes P \xrightarrow{\sigma} P \xleftarrow{\sigma} \delta(P) \otimes P & &
 \end{array}$$

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

Affine Substitution Algebra

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

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

$$\begin{array}{ccc}
 \delta(P) \otimes \delta(P) \otimes P \xrightarrow{\text{str}' \otimes P} \delta(\delta(P) \otimes P) \otimes P & & \\
 \delta(P) \otimes \sigma \downarrow & & \downarrow \delta(\sigma) \otimes P \\
 \delta(P) \otimes P \xrightarrow{\sigma} P \xleftarrow{\sigma} \delta(P) \otimes P & &
 \end{array}$$

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

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

The Relevant Case

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

$$\begin{array}{ccccc}
 A^{\otimes 3} & \xrightarrow{\mu^{\otimes A}} & A^{\otimes 2} & & A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma^{\otimes A}} & A^{\otimes 3} & \xrightarrow{A^{\otimes} \varsigma} & A^{\otimes 3} \\
 A^{\otimes} \mu \downarrow & & \downarrow \mu & & \mu \searrow & & \downarrow \mu & & A^{\otimes} \mu \downarrow & & \downarrow \mu^{\otimes A} \\
 A^{\otimes 2} & \xrightarrow{\mu} & A & & & & A & & A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & &
 \end{array}$$

$$\begin{array}{ccccc}
 A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma^{\otimes A}} & A^{\otimes 3} & \xrightarrow{A^{\otimes} \varsigma} & A^{\otimes 3} \\
 & \searrow & \downarrow \varsigma & & A^{\otimes} \varsigma \downarrow & & \downarrow \varsigma^{\otimes A} \\
 & & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma^{\otimes A}} & A^{\otimes 3} & \xrightarrow{A^{\otimes} \varsigma} & A^{\otimes 3}
 \end{array}$$

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

The Relevant Case

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

$$\begin{array}{ccccc}
 A^{\otimes 3} & \xrightarrow{\mu \otimes A} & A^{\otimes 2} & & A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
 A \otimes \mu \downarrow & & \downarrow \mu & & \searrow \mu & & \downarrow \mu & & A \otimes \mu \downarrow & & \downarrow \mu \otimes A \\
 A^{\otimes 2} & \xrightarrow{\mu} & A & & A & & A & & A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 2}
 \end{array}$$

$$\begin{array}{ccccc}
 A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
 & \searrow & \downarrow \varsigma & & A \otimes \varsigma \downarrow & & \downarrow \varsigma \otimes A \\
 & & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3}
 \end{array}$$

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

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

The Relevant Case

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

$$\begin{array}{ccccc}
 A^{\otimes 3} & \xrightarrow{\mu \otimes A} & A^{\otimes 2} & & A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
 A^{\otimes \mu} \downarrow & & \downarrow \mu & & \searrow \mu & & \downarrow \mu & & A^{\otimes \mu} \downarrow & & \downarrow \mu \otimes A \\
 A^{\otimes 2} & \xrightarrow{\mu} & A & & A & & A & & A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 2}
 \end{array}$$

$$\begin{array}{ccccc}
 A^{\otimes 2} & \xrightarrow{\varsigma} & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3} \\
 & \searrow & \downarrow \varsigma & & A^{\otimes \varsigma} \downarrow & & \downarrow \varsigma \otimes A \\
 & & A^{\otimes 2} & & A^{\otimes 3} & \xrightarrow{\varsigma \otimes A} & A^{\otimes 3} & \xrightarrow{A \otimes \varsigma} & A^{\otimes 3}
 \end{array}$$

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

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

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:

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

Relevant Substitution Algebra

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

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

$$\begin{array}{ccc}
 \delta(P) \otimes \delta(P) \otimes P \xrightarrow{\text{str}' \otimes P} \delta(\delta(P) \otimes P) \otimes P & & \\
 \delta(P) \otimes \sigma \downarrow & & \downarrow \delta(\sigma) \otimes P \\
 \delta(P) \otimes P \xrightarrow{\sigma} P \xleftarrow{\sigma} \delta(P) \otimes P & &
 \end{array}$$

Relevant Substitution Algebra

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

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

$$\begin{array}{ccc}
 \delta(P) \otimes \delta(P) \otimes P \xrightarrow{\text{str}' \otimes P} \delta(\delta(P) \otimes P) \otimes P & & \\
 \delta(P) \otimes \sigma \downarrow & & \downarrow \delta(\sigma) \otimes P \\
 \delta(P) \otimes P \xrightarrow{\sigma} P \xleftarrow{\sigma} \delta(P) \otimes P & &
 \end{array}$$

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

Relevant Substitution Algebra

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

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

$$\begin{array}{ccc}
 \delta(P) \otimes \delta(P) \otimes P & \xrightarrow{\text{str}' \otimes P} & \delta(\delta(P) \otimes P) \otimes P \\
 \delta(P) \otimes \sigma \downarrow & & \downarrow \delta(\sigma) \otimes P \\
 \delta(P) \otimes P & \xrightarrow{\sigma} & P \xleftarrow{\sigma} \delta(P) \otimes P
 \end{array}$$

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

$$\begin{array}{ccc}
 \delta^2(P) \otimes \delta(P) \otimes P \otimes P & \xrightarrow{\cong} & \delta^2(P) \otimes P \otimes \delta(P) \otimes P & \xrightarrow{\text{str}} & \delta(\delta(P) \otimes P) \otimes \delta(P) \otimes P \\
 \Delta \uparrow & & & & \downarrow \delta(\sigma) \otimes \sigma \\
 \delta^2(P) \otimes \delta(P) \otimes P & & & & \delta(P) \otimes P \\
 \rho \otimes P \downarrow & & & & \downarrow \sigma \\
 \delta(\delta(P) \otimes P) \otimes P & \xrightarrow{\delta(\sigma) \otimes P} & \delta(P) \otimes P & \xrightarrow{\sigma} & P
 \end{array}$$

Relevant Substitution Algebra

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

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

$$\begin{array}{ccc}
 \delta(P) \otimes \delta(P) \otimes P & \xrightarrow{\text{str}' \otimes P} & \delta(\delta(P) \otimes P) \otimes P \\
 \delta(P) \otimes \sigma \downarrow & & \downarrow \delta(\sigma) \otimes P \\
 \delta(P) \otimes P & \xrightarrow{\sigma} & P \xleftarrow{\sigma} \delta(P) \otimes P
 \end{array}$$

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

$$\begin{array}{ccc}
 \delta^2(P) \otimes \delta(P) \otimes P \otimes P & \xrightarrow{\cong} & \delta^2(P) \otimes P \otimes \delta(P) \otimes P \xrightarrow{\text{str}} \delta(\delta(P) \otimes P) \otimes \delta(P) \otimes P \\
 \Delta \uparrow & & \downarrow \delta(\sigma) \otimes \sigma \\
 \delta^2(P) \otimes \delta(P) \otimes P & & \delta(P) \otimes P \\
 \rho \otimes P \downarrow & & \downarrow \sigma \\
 \delta(\delta(P) \otimes P) \otimes P & \xrightarrow{\delta(\sigma) \otimes P} & \delta(P) \otimes P \xrightarrow{\sigma} P
 \end{array}$$

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

Binding Signatures

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

Binding Signatures

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

Ω is a set of **operations**

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

For $\omega \in \Omega$ with $\alpha(\omega) = (n_1, \dots, n_k)$

k is the usual arity \rightsquigarrow how many inputs

Each n_i say how many variables ω binds in the i th input

Binding Signatures

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

Ω is a set of **operations**

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

For $\omega \in \Omega$ with $\alpha(\omega) = (n_1, \dots, n_k)$

k is the usual arity \rightsquigarrow how many inputs

Each n_i say how many variables ω binds in the i th input

eg. **Lambda Calculus**

$\Omega = \{\mathbf{abs}, \mathbf{app}\}$

$\alpha(\mathbf{abs}) = (1) \rightsquigarrow$ 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**
 $\alpha : \Omega \rightarrow \mathbb{N}^*$ is an **arity** map

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

eg. **Lambda Calculus**

$\Omega = \{\mathbf{abs}, \mathbf{app}\}$

$\alpha(\mathbf{abs}) = (1) \rightsquigarrow$ 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-substitution

See [Fiore, Mahmoud, Hur, Szamozvancev](#) for cartesian case

Binding for Cartesian Syntax

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

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

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

$$\tau : TS \rightarrow ST$$

$$\begin{array}{ccccc}
 S & & TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT & & TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\
 \eta S \downarrow & \searrow S\eta & \mu S \downarrow & & & & \downarrow S\mu & & \varsigma S \downarrow & & & & \downarrow S\varsigma \\
 TS & \xrightarrow{\tau} & ST & & TS & \xrightarrow{\tau} & ST & & TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT
 \end{array}$$

Symmetric Distributive Laws

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

$$\tau : TS \rightarrow ST$$

$$\begin{array}{ccccc}
 S & & TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT & & TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\
 \eta S \downarrow & \searrow S\eta & \mu S \downarrow & & & & \downarrow S\mu & & \varsigma S \downarrow & & & & \downarrow S\varsigma \\
 TS & \xrightarrow{\tau} & ST & & TS & \xrightarrow{\tau} & ST & & TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT
 \end{array}$$

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

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

Symmetric Distributive Laws

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

$$\tau : TS \rightarrow ST$$

$$\begin{array}{ccccc}
 S & & TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT & & TTS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT \\
 \eta S \downarrow & \searrow S\eta & \mu S \downarrow & & & & \downarrow S\mu & & \varsigma S \downarrow & & & & \downarrow S\varsigma \\
 TS & \xrightarrow{\tau} & ST & & TS & \xrightarrow{\tau} & ST & & TS & \xrightarrow{T\tau} & TST & \xrightarrow{\tau T} & STT
 \end{array}$$

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

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

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

$- \times Q$ is a symmetric comonad

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

Lifting Symmetric Distributive Laws

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

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

is a symmetric distributive law

Lifting Symmetric Distributive Laws

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

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

is a symmetric distributive law

δ cartesian monoidal and cocartesian monoidal

$$\mathbf{swap} : \delta\Sigma \rightarrow \Sigma\delta$$

Lifting Symmetric Distributive Laws

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

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

is a symmetric distributive law

δ cartesian monoidal and cocartesian monoidal

$$\mathbf{swap} : \delta\Sigma \rightarrow \Sigma\delta$$

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

$$\mathbf{str} : \Sigma(P) \times Q \rightarrow \Sigma(P \times Q)$$

Abstract Syntax of a Binding Signature

Σ -algebra: (P, h) where $h : \Sigma(P) \rightarrow P$

Σ -Alg: Category of Σ -algebras and homomorphisms

Abstract Syntax of a Binding Signature

Σ -algebra: (P, h) where $h : \Sigma(P) \rightarrow P$

Σ -**Alg**: Category of Σ -algebras and homomorphisms

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

Abstract Syntax of a Binding Signature

Σ -algebra: (P, h) where $h : \Sigma(P) \rightarrow P$

Σ -Alg: Category of Σ -algebras and homomorphisms

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

Free functor: $F : \mathcal{F} \rightarrow \Sigma\text{-Alg}$

Left adjoint to U

For P in \mathcal{F} , consider the endofunctor $P + \Sigma : \mathcal{F} \rightarrow \mathcal{F}$

Let $[\eta_P, \varphi_P] : P + \Sigma(TP) \rightarrow TP$ 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 $h : \Sigma(P) \rightarrow P$

Σ -Alg: Category of Σ -algebras and homomorphisms

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

Free functor: $F : \mathcal{F} \rightarrow \Sigma\text{-Alg}$

Left adjoint to U

For P in \mathcal{F} , consider the endofunctor $P + \Sigma : \mathcal{F} \rightarrow \mathcal{F}$

Let $[\eta_P, \varphi_P] : P + \Sigma(TP) \rightarrow TP$ 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 $\rightsquigarrow [\eta_V, \varphi_V] : V + \Sigma(TV) \rightarrow TV$

Abstract Syntax of a Binding Signature

Σ -algebra: (P, h) where $h : \Sigma(P) \rightarrow P$

Σ -Alg: Category of Σ -algebras and homomorphisms

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

Free functor: $F : \mathcal{F} \rightarrow \Sigma\text{-Alg}$

Left adjoint to U

For P in \mathcal{F} , consider the endofunctor $P + \Sigma : \mathcal{F} \rightarrow \mathcal{F}$

Let $[\eta_P, \varphi_P] : P + \Sigma(TP) \rightarrow TP$ 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 $\rightsquigarrow [\eta_V, \varphi_V] : V + \Sigma(TV) \rightarrow TV$

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 $\alpha : S(A) \rightarrow A$ and S' -algebra $\beta : S'(B) \rightarrow B$, and a natural transformation $\psi : FS \rightarrow S'F$ there exists a unique $f : F(A) \rightarrow B$ such that

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

Substitution for Abstract Syntax

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

Substitution for Abstract Syntax

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

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

Substitution for Abstract Syntax

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

Variables: $\nu : 1 \rightarrow \delta(TV)$ 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 $\eta_V : V \rightarrow TV$ and $\varphi_V : \Sigma(TV) \rightarrow TV$

Variables: $\nu : 1 \rightarrow \delta(TV)$ 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:

$$\begin{array}{ccc}
 \Sigma(\delta(TV) \times TV) & \xrightarrow{\Sigma(\sigma)} & \Sigma(TV) \\
 \text{str} \uparrow & & \downarrow \varphi_V \\
 \Sigma\delta(TV) \times TV & & \\
 \text{swap} \times TV \uparrow & & \\
 \delta\Sigma(TV) \times TV & & \\
 \delta(\varphi_V) \times TV \downarrow & & \\
 \delta(TV) \times TV & \xrightarrow{\sigma} & TV \\
 \eta_V \times TV \uparrow & \nearrow \beta & \\
 \delta(V) \times TV & &
 \end{array}$$

Substitution for Abstract Syntax

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

Variables: $\nu : 1 \rightarrow \delta(TV)$ 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:

$$\begin{array}{ccc} \Sigma(\delta(TV) \times TV) & \xrightarrow{\Sigma(\sigma)} & \Sigma(TV) \\ \text{str} \uparrow & & \downarrow \varphi_V \\ \Sigma\delta(TV) \times TV & & \\ \text{swap} \times TV \uparrow & & \\ \delta\Sigma(TV) \times TV & & \\ \delta(\varphi_V) \times TV \downarrow & & \\ \delta(TV) \times TV & \xrightarrow{\sigma} & TV \\ \eta_V \times TV \uparrow & \nearrow \beta & \\ \delta(V) \times TV & & \end{array}$$

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

Substitution for Abstract Syntax

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

Variables: $\nu : 1 \rightarrow \delta(TV)$ 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:

$$\begin{array}{ccc} \Sigma(\delta(TV) \times TV) & \xrightarrow{\Sigma(\sigma)} & \Sigma(TV) \\ \text{str} \uparrow & & \downarrow \varphi_V \\ \Sigma\delta(TV) \times TV & & \\ \text{swap} \times TV \uparrow & & \\ \delta\Sigma(TV) \times TV & & \\ \delta(\varphi_V) \times TV \downarrow & & \\ \delta(TV) \times TV & \xrightarrow{\sigma} & TV \\ \eta_V \times TV \uparrow & \nearrow \beta & \\ \delta(V) \times TV & & \end{array}$$

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:

$$\Sigma\text{-Alg} \begin{array}{c} \xleftarrow{F} \\ \perp \\ \xrightarrow{U} \end{array} \mathcal{B}$$

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:

$$\Sigma\text{-Alg} \begin{array}{c} \xleftarrow{F} \\ \perp \\ \xrightarrow{U} \end{array} \mathcal{B}$$

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

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:

$$\Sigma\text{-Alg} \begin{array}{c} \xleftarrow{F} \\ \perp \\ \xrightarrow{U} \end{array} \mathcal{B}$$

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

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:

$$\Sigma\text{-Alg} \begin{array}{c} \xleftarrow{F} \\ \perp \\ \xrightarrow{U} \end{array} \mathcal{B}$$

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

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

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

Derived Signature Functors

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

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

Derived Signature Functors

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

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

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

Derived Signature Functors

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

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

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

Derived Signature Functors

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

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

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

Derived Signature Functors

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

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

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

Derived Signature Functors

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

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

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

Derived Signature Functors

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

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

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

Derived Signature Functors

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

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

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

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

Derived Signature Functors

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

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

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

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

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

Σ^\dagger linear in Q : $\text{str} : \Sigma^\dagger(P, Q) \otimes R \rightarrow \Sigma^\dagger(P, Q \otimes R)$

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

Substitution for Linear Abstract Syntax

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

Substitution for Linear Abstract Syntax

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

Basic Substitution:

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

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

Substitution for Linear Abstract Syntax

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

Basic Substitution:

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

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

Substitution:

$$\begin{array}{ccc} \Sigma^\dagger(TV, \delta(TV) \otimes TV) & \xrightarrow{\Sigma(TV, \sigma)} & \Sigma^\dagger(TV) \\ \text{str} \uparrow & & \downarrow \varphi^\dagger \\ \Sigma^\dagger(TV, \delta(TV)) \otimes TV & & \\ \text{swap} \otimes TV \uparrow & & \\ \delta\Sigma(TV) \otimes TV & & \\ \delta(\varphi_V) \otimes TV \downarrow & & \\ \delta(TV) \otimes TV & \xrightarrow{\text{---}\sigma\text{---}} & TV \\ \delta(\eta_V) \otimes TV \uparrow & \nearrow \beta & \\ \delta(V) \otimes TV & & \end{array}$$

Substitution for Linear Abstract Syntax

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

Basic Substitution:

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

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

Substitution:

$$\begin{array}{ccc} \Sigma^\dagger(TV, \delta(TV) \otimes TV) & \xrightarrow{\Sigma(TV, \sigma)} & \Sigma^\dagger(TV) \\ \text{str} \uparrow & & \downarrow \varphi^\dagger \\ \Sigma^\dagger(TV, \delta(TV)) \otimes TV & & \\ \text{swap} \otimes TV \uparrow & & \\ \delta \Sigma(TV) \otimes TV & & \\ \delta(\varphi_V) \otimes TV \downarrow & & \\ \delta(TV) \otimes TV & \xrightarrow{\sigma} & TV \\ \delta(\eta_V) \otimes TV \uparrow & \nearrow \beta & \\ \delta(V) \otimes TV & & \end{array}$$

(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: $[\eta_V, \varphi_V] : V + \Sigma(TV) \rightarrow TV$

Affine and Relevant Cases

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

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

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: $[\eta_V, \varphi_V] : V + \Sigma(TV) \rightarrow TV$

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

Affine and Relevant Cases

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

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

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: **swap** : $\delta\Sigma(P) \cong \Sigma^\dagger(P, \delta(P))$

Day convolution structure: **str** : $\Sigma^\dagger(P, Q) \otimes R \rightarrow \Sigma^\dagger(P, Q \otimes R)$

Substitution Algebras: Induced as before

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

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

Towards Uniform Axiomatisation

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

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

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

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

Towards Uniform Axiomatisation

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

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

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

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

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

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

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

Towards Uniform Axiomatisation

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

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

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

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

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

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

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

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