Discrete Generalised Polynomial Functors

Marcelo Fiore

Computer Laboratory
University of Cambridge

ICALP 2012
12.VII.2012

This Talk

1. Motivate and explain a general notion of polynomial.
2. Introduce generalised polynomials and present their basic theory.
3. Sketch an application to type theory.

The Two Aspects of Polynomials

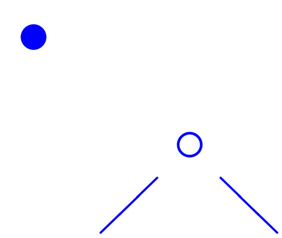
Intensional	Extensional

- Algebra

$p(x) = ax^2 + bx + c$	$f : \mathbb{R} \longrightarrow \mathbb{R}$ $r \mapsto ar^2 + br + c$

- Programming

$$\begin{array}{rcl} \mathtt{T}\,\alpha & = & \mathtt{Nil} \\ & \mid & \mathtt{Cons}(\alpha, \alpha) \end{array}$$



Polynomial Constructions
(Sums of Products)

- Mono-sorted.

$$X \mapsto \sum_{\gamma \in C} X^{|\gamma|}$$

$C =$ set of constructors

$|\gamma| =$ arity of γ

Polynomial Constructions
(Sums of Products)

- Mono-sorted.

  $X \mapsto \sum_{\gamma \in C} X^{|\gamma|}$

  $C =$ set of constructors
  $|\gamma| =$ arity of γ

- Multi-sorted:

  $(X_i)_{i \in S} \mapsto \sum_{\gamma \in C} \prod_{a \in |\gamma|} X_{\sigma(a)}$

  $C =$ set of constructors
  $|\gamma| =$ arity *positions* of γ
  $\sigma : |\gamma| \to S =$ sorting function

Polynomial Constructions
(Sums of Products)

- Mono-sorted.

  $$X \mapsto \sum_{\gamma \in C} X^{|\gamma|}$$

  $C =$ set of constructors

  $|\,\gamma\,| =$ arity of γ

- Multi-sorted:

  $$(X_i)_{i \in S} \mapsto \sum_{\gamma \in C} \prod_{a \in |\gamma|} X_{\sigma(a)}$$

  $C =$ set of constructors

  $|\,\gamma\,| =$ arity *positions* of γ

  $\sigma : |\,\gamma\,| \to S =$ sorting function

Polynomials $\rightsquigarrow$ $\sum_{\gamma \in C} |\,\gamma\,| \longrightarrow C$, $\quad \sigma : \sum_{\gamma \in C} |\,\gamma\,| \to S$

$$\begin{array}{ccc} \sum_{\gamma \in C} |\,\gamma\,| & \longrightarrow & C \\ \downarrow \sigma & & \\ S & & \end{array}$$

Polynomials

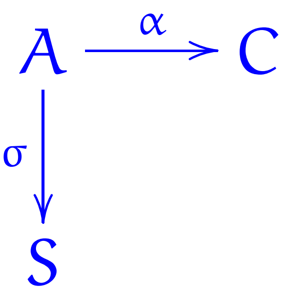
- Single valued.

$$
\begin{array}{ccc}
A & \xrightarrow{\alpha} & C \\
{\scriptstyle\sigma}\downarrow & & \\
S & &
\end{array}
$$

$$(X_i)_{i \in S} \mapsto \sum_{\gamma \in C} \prod_{a \in \alpha^{-1}(\gamma)} X_{\sigma(a)}$$

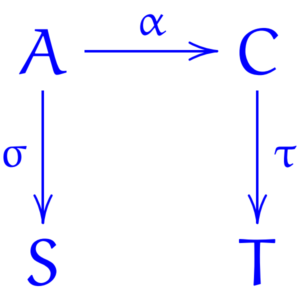
Polynomials

- Single valued.



$$(X_i)_{i\in S} \mapsto \sum_{\gamma\in C} \prod_{a\in\alpha^{-1}(\gamma)} X_{\sigma(a)}$$

- Multi-valued.



$$(X_i)_{i\in S} \mapsto \Big(\sum_{\gamma\in\tau^{-1}(j)} \prod_{a\in\alpha^{-1}(\gamma)} X_{\sigma(a)} \Big)_{j\in T}$$

Polynomials

- Single valued.

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & C \\ {\scriptstyle\sigma}\big\downarrow & & \\ S & & \end{array}$$

$$(X_i)_{i \in S} \mapsto \sum_{\gamma \in C} \prod_{a \in \alpha^{-1}(\gamma)} X_{\sigma(a)}$$

- Multi-valued.

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & C \\ {\scriptstyle\sigma}\big\downarrow & & \big\downarrow{\scriptstyle\tau} \\ S & & T \end{array}$$

$$(X_i)_{i \in S} \mapsto \Big(\sum_{\gamma \in \tau^{-1}(j)} \prod_{a \in \alpha^{-1}(\gamma)} X_{\sigma(a)} \Big)_{j \in T}$$

CONSTRUCTIONS		STRUCTURE
		additive
Polynomial	$\leftsquigarrow$	$\dashv$ reindexing
		$\dashv$ multiplicative

Additive and Multiplicative Transfer Structure

- Logic.

$$
\begin{array}{c} A \\ \big\downarrow \\ B \end{array}
\quad \longmapsto \quad
\begin{array}{c} 2^A \\ \exists \big\downarrow \dashv \big\uparrow \dashv \big\downarrow \forall \\ 2^B \end{array}
$$

Additive and Multiplicative Transfer Structure

- Logic.

$$\begin{array}{c} A \\ \big\downarrow \\ B \end{array} \quad \longmapsto \quad \begin{array}{c} 2^A \\ \exists \big\downarrow \dashv \big\uparrow \dashv \big\downarrow \forall \\ 2^B \end{array}$$

- Type theory.

$$\begin{array}{c} A \\ \big\downarrow \\ B \end{array} \quad \longmapsto \quad \begin{array}{c} \mathcal{S}_{/A} \\ \Sigma \big\downarrow \dashv \big\uparrow \dashv \big\downarrow \Pi \\ \mathcal{S}_{/B} \end{array}$$

Additive and Multiplicative Transfer Structure

- Logic.

$$\begin{array}{c} A \\ \big\downarrow \\ B \end{array} \quad \mapsto \quad \begin{array}{c} 2^{A} \\ \exists \big\downarrow \dashv \big\uparrow \dashv \big\downarrow \forall \\ 2^{B} \end{array}$$

- Type theory.

$$\begin{array}{c} A \\ \big\downarrow \\ B \end{array} \quad \mapsto \quad \begin{array}{c} \mathcal{S}_{/A} \\ \Sigma \big\downarrow \dashv \big\uparrow \dashv \big\downarrow \Pi \\ \mathcal{S}_{/B} \end{array}$$

- Generalised logic.

$$\begin{array}{c} \mathbb{A} \\ \big\downarrow \\ \mathbb{B} \end{array} \quad \mapsto \quad \begin{array}{c} \mathcal{S}et^{\mathbb{A}} \\ \mathrm{Lan} \big\downarrow \dashv \big\uparrow \dashv \big\downarrow \mathrm{Ran} \\ \mathcal{S}et^{\mathbb{B}} \end{array}$$

Kan Extensions

Every

$$f : \mathbb{A} \longrightarrow \mathbb{B}$$

induces

$$\begin{array}{ccc} & \xrightarrow{\; f_* \;} & \\ & \top & \\ \mathcal{P}\mathbb{A} & \xleftarrow{\; f^* \;} & \mathcal{P}\mathbb{B} \\ & \top & \\ & \xrightarrow[\; f_! \;]{} & \end{array}$$

where

$$\mathcal{P}\mathbb{C} = \boldsymbol{\mathcal{S}et}^{\mathbb{C}}$$

and

$$\begin{aligned} (f_* P)_b &= (\mathrm{Ran}_f P)_b = \int_{a \in \mathbb{A}} [\, \mathbb{B}(b, fa) \Rightarrow P_a \,] \\ (f^* Q)_a &= Q_{fa} \\ (f_! P)_b &= (\mathrm{Lan}_f P)_b = \int^{a \in \mathbb{A}} \mathbb{B}(fa, b) \times P_a \end{aligned}$$

Generalised Polynomial Functors

- Polynomials in $\mathcal{C}at$.

$$\mathrm{P} = (\mathbb{A} \xleftarrow{\mathrm{s}} \mathbb{I} \xrightarrow{\mathrm{f}} \mathbb{J} \xrightarrow{\mathrm{t}} \mathbb{B})$$

Generalised Polynomial Functors

- Polynomials in $\mathcal{C}at$.

$$P = (\mathbb{A} \xleftarrow{s} \mathbb{I} \xrightarrow{f} \mathbb{J} \xrightarrow{t} \mathbb{B})$$

- Generalised polynomial functors
  between presheaf categories.

$$F_P = (\ \mathcal{P}\mathbb{A} \xrightarrow{s^*} \mathcal{P}\mathbb{I} \xrightarrow{f_*} \mathcal{P}\mathbb{J} \xrightarrow{t_!} \mathcal{P}\mathbb{B}\ )$$

$$(F_P\,A)_b = \int^{j\in\mathbb{J}} \mathbb{B}(tj, b) \times \int_{i\in\mathbb{I}} \big[\,\mathbb{J}(j, fi) \Rightarrow A_{si}\,\big]$$

Generalised Polynomial Functors

- Polynomials in $\mathcal{C}at$.

$$P = (\mathbb{A} \xleftarrow{s} \mathbb{I} \xrightarrow{f} \mathbb{J} \xrightarrow{t} \mathbb{B})$$

- Generalised polynomial functors between presheaf categories.

$$F_P = (\ \mathcal{P}\mathbb{A} \xrightarrow{s^*} \mathcal{P}\mathbb{I} \xrightarrow{f_*} \mathcal{P}\mathbb{J} \xrightarrow{t_!} \mathcal{P}\mathbb{B}\ )$$

$$\boxed{(F_P\, A)_b = \int^{j \in \mathbb{J}} \mathbb{B}(tj, b) \times \int_{i \in \mathbb{I}} \left[\, \mathbb{J}(j, fi) \Rightarrow A_{si} \,\right]}$$

- Generalised polynomial functors are continuous, and hence admit final coalgebras.

Examples:

- For every presheaf P, the product endofunctor $(-) \times P$ and the exponential endofunctor $(-)^P$ are generalised polynomial.
- The class of generalised polynomial functors:
  1. contains the constant, cocontinuous, and projections functors, and
  2. is closed under sums and finite products.

Discrete Generalised Polynomial Functors

- The class of *discrete* generalised polynomial functors is represented by sums of polynomial diagrams of the form

$$M = (\mathbb{A} \xleftarrow{s} L \cdot \mathbb{J} \xrightarrow{\nabla_L} \mathbb{J} \xrightarrow{t} \mathbb{B})$$

where L is finite.

$$(F_M A)_b = \int^{j \in \mathbb{J}} \mathbb{B}(tj, b) \times \prod_{\ell \in L} A_{s(\ell \cdot j)}$$

Discrete Generalised Polynomial Functors

- The class of *discrete* generalised polynomial functors is represented by sums of polynomial diagrams of the form

$$M = (\mathbb{A} \xleftarrow{s} L \cdot \mathbb{J} \xrightarrow{\nabla_L} \mathbb{J} \xrightarrow{t} \mathbb{B})$$

where L is finite.

$$\boxed{(F_M A)_b = \int^{j \in \mathbb{J}} \mathbb{B}(tj, b) \times \prod_{\ell \in L} A_{s(\ell \cdot j)}}$$

- Discrete generalised polynomial functors are finitary and preserve epimorphisms. Hence they admit inductively constructed free algebras.

Examples:

- **Convolution monoidal closed structure**
  1. *Day's monoidal-convolution tensor product* is [isomorphic to] a discrete generalised polynomial functor.
  2. *Monoidal-convolution exponentiation to a representable* is a discrete generalised polynomial functor.

Examples:

- **Convolution monoidal closed structure**
  1. *Day's monoidal-convolution tensor product* is [isomorphic to] a discrete generalised polynomial functor.
  2. *Monoidal-convolution exponentiation to a representable* is a discrete generalised polynomial functor.
- The class of discrete generalised polynomial functors:
  1. contains the constant, cocontinuous, and projections functors, and
  2. is closed under sums, finite products, *composition*, and *differentiation*.

Discrete Generalised Polynomial Functors in Type Theory

- Vernacular syntactic rules in simple, polymorphic, and dependent type theories are discrete generalised polynomials.
- Their associated functors describe the algebraic structure of type theories.
- Models of type theories are algebras. Initial algebras universally characterise the syntax.

Simply-Typed λ-Calculus

- Let S be a set of sorts closed under an $\Rrightarrow$ type constructor; that is,

$$S \times S \xrightarrow{\Rrightarrow} S \quad .$$

Simply-Typed λ-Calculus

- Let S be a set of sorts closed under an ⇛ type constructor; that is,

$$\mathcal{S} \times \mathcal{S} \xrightarrow{\Rrightarrow} \mathcal{S}\ .$$

- Let $\mathbf{C} = \mathfrak{FinSet}_{/\mathsf{S}}$ be the category of S-sorted contexts, and write

$$\begin{array}{ccc} \mathbf{C} \times \mathsf{S} & \stackrel{\cdot}{\longrightarrow} & \mathbf{C} \\ \Gamma, \sigma & \mapsto & \Gamma \cdot \sigma \end{array}$$

for the operation of context extension.

1. The *application* rule

$$(@)\ \frac{\vdash t : \sigma_1 \Rightarrow \sigma_2 \qquad \vdash t' : \sigma_1}{\vdash t(t') : \sigma_2}$$

1. The *application* rule

$$(@)\quad \frac{\vdash t : \sigma_1 \Rightarrow \sigma_2 \qquad\qquad \vdash t' : \sigma_1}{\vdash t(t') : \sigma_2}$$

corresponds to the discrete polynomial

$$\begin{array}{ccc} 2\cdot(\mathbf{C}\times\mathsf{S}\times\mathsf{S}) & \xrightarrow{\nabla_2} & \mathbf{C}\times\mathsf{S}\times\mathsf{S} \\ {\scriptstyle [\mathrm{id}\times\Rightarrow,\,\mathrm{id}\times\pi_1]}\downarrow & & \downarrow{\scriptstyle \mathrm{id}\times\pi_2} \\ \mathbf{C}\times\mathsf{S} & & \mathbf{C}\times\mathsf{S} \end{array}$$

1. The *application* rule

$$(@)\quad \frac{\vdash t : \sigma_1 \Rightarrow \sigma_2 \qquad \vdash t' : \sigma_1}{\vdash t(t') : \sigma_2}$$

corresponds to the discrete polynomial

$$\begin{array}{ccc} 2 \cdot (\mathbf{C} \times \mathsf{S} \times \mathsf{S}) & \xrightarrow{\nabla_2} & \mathbf{C} \times \mathsf{S} \times \mathsf{S} \\ {\scriptstyle [\mathrm{id}\times\Rightarrow,\mathrm{id}\times\pi_1]}\downarrow & & \downarrow{\scriptstyle \mathrm{id}\times\pi_2} \\ \mathbf{C} \times \mathsf{S} & & \mathbf{C} \times \mathsf{S} \end{array}$$

An $F_{(@)}$-algebra

$$F_{(@)}(T) \to T : \mathbf{C} \times \mathsf{S} \to \mathcal{S}et$$

is a natural transformation

$$\{\, T(\Gamma, \sigma_1 \Rightarrow \sigma_2) \times T(\Gamma, \sigma_1) \to T(\Gamma, \sigma_2) \,\}_{\Gamma, \sigma_1, \sigma_2}$$

2. The *abstraction* rule

$$(\lambda) \quad \frac{x : \tau_1 \vdash t : \tau_2}{\vdash \lambda x.\, t : \tau_1 \Rightarrow \tau_2}$$

2. The *abstraction* rule

$$(\lambda)\quad \frac{x : \tau_1 \vdash t : \tau_2}{\vdash \lambda x.\, t : \tau_1 \Rightarrow \tau_2}$$

corresponds to the discrete polynomial

$$\begin{array}{ccc} 1 \cdot (\mathbf{C} \times \mathsf{S} \times \mathsf{S}) & \xrightarrow{\nabla_1} & \mathbf{C} \times \mathsf{S} \times \mathsf{S} \\ {\scriptstyle [\cdot \times \mathrm{id}]}\downarrow & & \downarrow{\scriptstyle \mathrm{id} \times \Rightarrow} \\ \mathbf{C} \times \mathsf{S} & & \mathbf{C} \times \mathsf{S} \end{array}$$

2. The *abstraction* rule

$$(\lambda)\ \frac{x : \tau_1 \vdash t : \tau_2}{\vdash \lambda x.\, t : \tau_1 \Rightarrow \tau_2}$$

corresponds to the discrete polynomial

$$\begin{array}{ccc} 1 \cdot (\mathbf{C} \times \mathsf{S} \times \mathsf{S}) & \xrightarrow{\nabla_1} & \mathbf{C} \times \mathsf{S} \times \mathsf{S} \\ {\scriptstyle [\cdot \times \mathrm{id}]}\downarrow & & \downarrow{\scriptstyle \mathrm{id} \times \Rightarrow} \\ \mathbf{C} \times \mathsf{S} & & \mathbf{C} \times \mathsf{S} \end{array}$$

An $F_{(\lambda)}$-algebra

$$F_{(\lambda)}(T) \to T : \mathbf{C} \times \mathsf{S} \to \mathcal{S}et$$

is a natural transformation

$$\{\, T(\Gamma \cdot \sigma_1, \sigma_2) \to T(\Gamma, \sigma_1 \Rightarrow \sigma_2) \,\}_{\Gamma, \sigma_1, \sigma_2}$$

Appendix

Dependent Context Structures

1. Let $\mathbb{C}$ be a category (of contexts) with a terminal object ϵ (the empty context).

2. Let

$$\begin{array}{c} \mathsf{T} \\ \downarrow \\ \mathsf{S} \end{array} \quad \text{in } \widehat{\mathbb{C}} \triangleq \boldsymbol{\mathcal{S}et}^{\mathbb{C}^\circ}$$

   be a bundle consisting of a presheaf (of terms) T over a presheaf (of sorts or types) S.

Conventions:

- $\Gamma \vdash \sigma$ denotes an object of the category of elements $\int S$; that is $\Gamma \in \mathbb{C}$ and $\sigma \in S(\Gamma)$.
- For $\Delta \xrightarrow{f} \Gamma \vdash \sigma$, one has $\Delta \vdash \sigma[f]$ where $\sigma[f] \triangleq Sf(\sigma)$.
- The bundle of terms over types is regarded as a presheaf $T \in \widehat{\int S}$.

  $\Gamma \vdash t : \sigma$ denotes an object in the category of elements $\int T$; that is, $t \in T(\Gamma \vdash \sigma)$.
- For $\Delta \xrightarrow{f} \Gamma \vdash t : \sigma$, one has $\Delta \vdash t[f] : \sigma[f]$ where $t[f] \triangleq Tf(t)$.

3. For every $\Gamma \vdash \sigma$, let

$$\pi_\sigma : (\Gamma \cdot \sigma) \to \Gamma \text{ in } \mathbb{C} \quad \text{and} \quad \Gamma \cdot \sigma \vdash \nu_\sigma : \sigma[\pi_\sigma]$$

be such that

$$\begin{array}{ccc} \boldsymbol{y}(\Gamma \cdot \sigma) & \xrightarrow{\nu_\sigma} & T \\ {\scriptstyle \pi_\sigma}\downarrow & & \downarrow \\ \boldsymbol{y}(\Gamma) & \xrightarrow[\sigma]{} & S \end{array}$$

is a pullback.

NB: This data and condition state that pullbacks of the bundle $\begin{array}{c} T \\ \downarrow \\ S \end{array}$ along representable generalised elements $\boldsymbol{y}(\Gamma) \to S$ are themselves representable in $\widehat{\mathbb{C}_{/\Gamma}}$.

This is in fact equivalent to the *context comprehension* axiom of Dybjer.

That is, that for every $\Delta \xrightarrow{f} \Gamma \vdash t : \sigma$ there exists a unique $\Delta \xrightarrow{\langle f,t \rangle} (\Gamma \cdot \sigma)$ in $\mathbb{C}$ such that $\pi_\sigma \langle f, t \rangle = f$ and $\nu_\sigma\big[\langle f, t \rangle\big] = t[f]$.

Identity Types

1. The *identity type* rule

$$(\mathsf{Id})\ \frac{\vdash \sigma}{x : \sigma, y : \sigma \vdash \mathsf{Id}_\sigma(x, y)}$$

corresponds to the discrete polynomial

$$\begin{array}{ccc} 0 & \longrightarrow & \int S \\ \big\downarrow & & \big\downarrow{\scriptstyle \delta} \\ \mathbb{C} & & \mathbb{C} \end{array}$$

where

$$\delta(\Gamma \vdash \sigma) \triangleq (\Gamma \cdot \sigma \cdot \sigma[\pi_\sigma]) \ .$$

An $F_{(\mathsf{Id})}$-algebra

$$F_{(\mathsf{Id})}(S) \longrightarrow S$$

is a family

$$\{\ \Gamma \cdot \sigma \cdot \sigma[\pi_\sigma]\ \vdash\ \mathrm{Id}(\sigma)\ \}_{\Gamma \vdash \sigma}$$

such that, for all $f : \Delta \longrightarrow \Gamma$,

$$\mathrm{Id}(\sigma)\big[\delta(f)\big] = \mathrm{Id}\big(\sigma[f]\big)$$

where

$$\delta(f) = f \cdot \sigma \cdot \sigma[\pi_\sigma]$$

with

$$g \cdot \tau \triangleq \langle g\,\pi_{\tau[g]}, \nu_{\tau[g]} \rangle \ .$$

2. The *reflexivity* rule

$$(\mathsf{r})\ \frac{\vdash \mathrm{t} : \sigma}{\vdash \mathsf{r}_\sigma(\mathrm{t}) : \mathrm{Id}_\sigma(\mathrm{t}, \mathrm{t})}$$

corresponds to the discrete polynomial

$$\begin{array}{ccc} 0 & \longrightarrow & \int \mathsf{T} \\ \big\downarrow & & \big\downarrow \gamma \\ \int \mathsf{S} & & \int \mathsf{S} \end{array}$$

where

$$\gamma(\Gamma \vdash \mathrm{t} : \sigma) \ = \ \big(\ \Gamma \ \vdash \ \mathrm{Id}(\sigma)\big[\langle \mathrm{id}_\Gamma, \mathrm{t}, \mathrm{t}\rangle\big]\ \big) \ \ .$$

An $F_{(r)}$-algebra

$$F_{(r)}(T) \longrightarrow T$$

is a family

$$\left\{ \ \Gamma \ \vdash \ r(t) : \mathrm{Id}(\sigma)\left[\langle \mathrm{id}_\Gamma, t, t\rangle\right] \ \right\}_{\Gamma \vdash t : \sigma}$$

such that

$$r(t)\left[f\right] \ = \ r\big(t[f]\big)$$

for all $f : \Delta \longrightarrow \Gamma$.

NB: For all $\Gamma \vdash t : \sigma$,

$$r(t) = r(\nu_\sigma)\left[\langle \mathrm{id}_\Gamma, t\rangle\right] \ .$$

3. The *elimination* rule

$$(\mathsf{J})\ \frac{x:\sigma, y:\sigma, p:\mathsf{Id}_\sigma(x,y) \vdash E(x,y,p) \qquad z:\sigma \vdash e[z] : E(z,z,\mathsf{r}(z))}{x:\sigma, y:\sigma, p:\mathsf{Id}_\sigma(x,y) \vdash \mathsf{J}(z.e[z],x,y,p) : E(x,y,p)}$$

(**NB:** J is a binding operator.)

corresponds to the discrete polynomial

$$\begin{array}{ccc} \int \mathsf{K} & \xrightarrow{\mathrm{id}} & \int \mathsf{K} \\ {\scriptstyle \lambda}\downarrow & & \downarrow{\scriptstyle \kappa} \\ \int \mathsf{S} & & \int \mathsf{S} \end{array}$$

where, for $\Gamma \ltimes \sigma \triangleq \big(\, \Gamma \cdot \sigma \cdot \sigma[\pi_\sigma] \cdot \mathrm{Id}(\sigma) \,\big)$,

$$\mathsf{K}(\Gamma \vdash \sigma) \triangleq \mathsf{S}(\Gamma \ltimes \sigma)$$

$$\kappa(\, \Gamma \vdash \sigma \,,\, E \,) \triangleq (\, \Gamma \ltimes \sigma \vdash E \,)$$

$$\lambda\big(\, \Gamma \vdash \sigma \,,\, E \,\big) \triangleq (\, \Gamma \cdot \sigma \vdash E[u_\sigma] \,)$$

with $u_\sigma \triangleq \langle \pi_\sigma, \nu_\sigma, \nu_\sigma, \mathrm{r}(\nu_\sigma) \rangle : (\Gamma \cdot \sigma) \to (\Gamma \ltimes \sigma)$.

An $F_{(J)}$-algebra

$$F_{(J)}(T) \longrightarrow T$$

is a family of maps

$$\{\ J(E) : T(\ \Gamma \cdot \sigma \vdash E[u_\sigma]\ ) \longrightarrow T(\ \Gamma \ltimes \sigma \vdash E\ )\ \}_{\Gamma \ltimes \sigma \vdash E}$$

such that, for all $\Gamma \cdot \sigma \vdash e : E[u_\sigma]$ and $f : \Delta \longrightarrow \Gamma$,

$$\big(\ J(E)(e)\ \big)[f \ltimes \sigma] \ = \ J(E[f \ltimes \sigma])\big(\ e[f \cdot \sigma]\ \big)$$

where

$$f \ltimes \sigma \triangleq f \cdot \sigma \cdot \sigma[\pi_\sigma] \cdot \mathrm{Id}(\sigma) : (\Delta \ltimes \sigma[f]) \longrightarrow (\Gamma \ltimes \sigma) \ .$$