Extensional Universal Types for Call-by-Value

Kazuyuki Asada

RIMS

1 July 2008

Overview

We give:

1. the $\lambda_c 2_\eta$-calculus (and $\lambda_c 2$-calculus):
   a second-order polymorphic call-by-value calculus with *extensional* universal types
2. - $\lambda_c 2_\eta$-models:
     categorical semantics for $\lambda_c 2_\eta$-calculus
   - monadic $\lambda_c 2_\eta$-models:
     categorical semantics for $\lambda_c 2_\eta$-calculus with the focus on monadic metalanguages like Haskell
3. relevant parametric models:
   domain theoretic concrete models of monadic $\lambda_c 2_\eta$-models with a reasonable class of monads including:
   Lifting, Exception, Global State, Input/Output, and List monad

Introduction

SystemF

Types $\sigma ::= \alpha \mid b \mid \sigma \to \sigma \mid \forall\alpha.\sigma$ (b: base types)

Terms $M ::= x \mid c^{\sigma} \mid \lambda x^{\sigma}.M \mid MM \mid \Lambda\alpha.M \mid M\sigma$ (c^{σ}: constans)

Typing Rules: All the rules in the λ-calculus, and:

$$\frac{\Xi, \alpha \mid \Gamma \vdash M : \sigma}{\Xi \mid \Gamma \vdash \Lambda\alpha.M : \forall\alpha.\sigma} \ (\alpha \notin \mathrm{FTV}(\Gamma)) \qquad \frac{\Xi \mid \Gamma \vdash M : \forall\alpha.\sigma}{\Xi \mid \Gamma \vdash M\tau : \sigma[\tau/\alpha]}$$

Axioms: All the axioms in the λ-calculus, and:

$$(\Lambda\alpha.M)\sigma = M[\sigma/\alpha] \qquad \Lambda\alpha.M\alpha = M \quad (\alpha \notin \mathrm{FTV}(M))$$

(Girard 1972, Reynolds 1974)

Polymorphic Encodings

$$1 := \forall\alpha.\alpha \to \alpha$$

$$\sigma \times \tau := \forall\alpha.\,(\sigma \to \tau \to \alpha) \to \alpha \qquad (\alpha \notin \mathrm{FTV}\,(\sigma,\tau))$$

$$0 := \forall\alpha.\alpha$$

$$\sigma + \tau := \forall\alpha.\,(\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha \qquad (\alpha \notin \mathrm{FTV}\,(\sigma,\tau))$$

$$\exists\alpha.sigma := \forall\beta.\,(\forall\alpha.\,(\sigma \to \beta)) \to \beta \qquad (\beta \neq \alpha, \beta \notin \mathrm{FTV}\,(\sigma))$$

$$\mu\gamma.\sigma := \forall\gamma.\,((\sigma \to \gamma) \to \gamma)$$

$$\nu\gamma.\sigma := \exists\gamma.\,((\gamma \to \sigma) \times \gamma)$$

Parametricity

relational parametricity is: (Reynolds 1983)

- (binary) relations $R \subseteq \tau \times \tau'$ between types
- relational interpretation $\sigma\left[\vec{R}/\vec{\alpha}\right]$ of types σ with relations $\vec{R}$
- the parametricity schema:
  $\forall\vec{\beta}.\,\forall u : \left(\forall\alpha.\sigma\left[\alpha,\vec{\beta}\right]\right).\,\forall\tau.\,\forall\tau'.\,\forall R \subseteq \tau\times\tau'.\ \left(u\tau\left(\sigma\left[R/\alpha,\overrightarrow{eq_{\beta}}/\vec{\beta}\right]\right)u\tau'\right)$

linear parametricity is: (Plotkin 1993)

- relations $R \subseteq \tau \times \tau'$ are restricted to *admissible relations*
- admissible relations are intuitively *subalgebras* of product of the two types

Example of Parametric Reasoning

For any $u : \forall\alpha.\alpha \to \alpha$,

$$
\begin{aligned}
& \Lambda\alpha.\lambda a^{\alpha}.a = u \\
\iff\ & \forall\alpha.\ (\lambda a^{\alpha}.a = u\alpha) && \text{(type } \eta\text{-equality)} \\
\iff\ & \forall\alpha.\forall a{:}\,\alpha.\ (a = u\alpha a) && (\eta\text{-equality})
\end{aligned}
$$

For α and $a{:}\,\alpha$, let

$$R := \langle \lambda x^{\alpha}.a \rangle \subseteq \alpha \times \alpha$$

Then by parametricity,

$$u\alpha\,(R \to R)\,u\alpha$$

Clearly,

$$aRa \qquad\qquad \therefore\ \ (u\alpha a)\,R\,(u\alpha a)$$

Hence

$$a = u\alpha a$$

λ_{c}-calculus and η-equality

$$\begin{array}{rrl} \text{Types} & \sigma ::= & b \mid \sigma \to \sigma \mid \top \mid \sigma \times \sigma \\ \text{Terms} & M ::= & x \mid c^{\sigma} \mid \lambda x^{\sigma}.M \mid MM \mid * \mid \langle M, M\rangle \mid \pi_1 M \mid \pi_2 M \\ \text{Values} & V ::= & x \mid c^{\sigma} \mid \lambda x^{\sigma}.M \mid * \mid \langle V, V\rangle \mid \pi_1 V \mid \pi_2 V \\ \text{Evaluation Contexts} & E ::= & [-] \mid EM \mid VE \mid \langle E, M\rangle \mid \langle V, E\rangle \mid \pi_1 E \mid \pi_2 E \end{array}$$

$$(\lambda x^{\sigma}.M)\,V = M\,[V/x] \qquad \pi_i\,\langle V_1, V_2\rangle = V_i \quad (i = 1, 2)$$

$$\lambda x^{\sigma}.Vx = V \quad (x \notin \mathrm{FV}\,(V)) \qquad \langle \pi_1 V, \pi_2 V\rangle = V$$

$$V = * \quad (V : \top) \qquad (\lambda m^{\sigma}.E\,[m])\,M = E\,[M] \quad (m \notin \mathrm{FV}\,(E))$$

(Moggi 1988)

$\lambda_{\mathrm{c}}2_{\eta}$-calculus:

$$(\Lambda\alpha.M)\,\sigma = M\,[\sigma/\alpha] \qquad \Lambda\alpha.M\alpha = M \quad (\alpha \notin \mathrm{FTV}\,(M))$$

Calculi

$\lambda_{\mathbf{c}}$-calculus

$$\text{Types}\quad \sigma ::= b \mid \sigma \to \sigma \mid \top \mid \sigma \times \sigma$$

$$\text{Terms}\quad M ::= x \mid c^{\sigma} \mid \lambda x^{\sigma}.M \mid MM \mid * \mid \langle M, M\rangle \mid \pi_1 M \mid \pi_2 M$$

$$\text{Values}\quad V ::= x \mid c^{\sigma} \mid \lambda x^{\sigma}.M \mid * \mid \langle V, V\rangle \mid \pi_1 V \mid \pi_2 V$$

$$\text{Evaluation Contexts}\quad E ::= [-] \mid EM \mid VE \mid \langle E, M\rangle \mid \langle V, E\rangle \mid \pi_1 E \mid \pi_2 E$$

$$\frac{x{:}\sigma \in \Gamma}{\Gamma \vdash x : \sigma} \qquad \frac{}{\Gamma \vdash c^{\sigma} : \sigma} \qquad \frac{\Gamma, x{:}\sigma \vdash M : \tau}{\Gamma \vdash \lambda x^{\sigma}.M : \sigma \to \tau} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

$$\frac{}{\Gamma \vdash * : \top} \qquad \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau}{\Gamma \vdash \langle M, N\rangle : \sigma \times \tau} \qquad \frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \pi_1 M : \sigma} \qquad \frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \pi_2 M : \tau}$$

$$(\lambda x^{\sigma}.M)\, V = M\,[V/x] \qquad \pi_i\,\langle V_1, V_2\rangle = V_i \quad (i = 1, 2)$$

$$\lambda x^{\sigma}.Vx = V \quad (x \notin \mathrm{FV}\,(V)) \qquad \langle \pi_1 V, \pi_2 V\rangle = V$$

$$V = * \quad (V : \top) \qquad (\lambda m^{\sigma}.E\,[m])\, M = E\,[M] \quad (m \notin \mathrm{FV}\,(E))$$

(Moggi 1988)

$\lambda_c 2$-calculus

Types $\sigma ::= \ldots \mid \alpha \mid \forall\alpha.\sigma$ Values $V ::= \ldots \mid \Lambda\alpha.M$

Terms $M ::= \ldots \mid \Lambda\alpha.M \mid M\sigma$ Evaluation Contexts $E ::= \ldots \mid E\sigma$

Typing Rules: All the rules in the λ_c-calculus, and:

$$\frac{\Xi, \alpha \mid \Gamma \vdash M : \sigma}{\Xi \mid \Gamma \vdash \Lambda\alpha.M : \forall\alpha.\sigma}\ (\alpha \notin \mathrm{FTV}(\Gamma)) \qquad \frac{\Xi \mid \Gamma \vdash M : \forall\alpha.\sigma}{\Xi \mid \Gamma \vdash M\tau : \sigma[\tau/\alpha]}$$

Axioms: All the axioms in the λ_c-calculus (with E extended as above), and:

$$(\Lambda\alpha.M)\,\sigma = M[\sigma/\alpha] \qquad \Lambda\alpha.V\alpha = V \quad (\alpha \notin \mathrm{FTV}(V))$$

$\lambda_{\mathbf{c}}2_{\eta}$-calculus

Types $\sigma ::= \ldots \mid \alpha \mid \forall\alpha.\sigma$ Values $V ::= \ldots \mid \Lambda\alpha.V \mid V\sigma$

Terms $M ::= \ldots \mid \Lambda\alpha.M \mid M\sigma$ Evaluation Contexts $E ::= \ldots \mid E\sigma$

Typing Rules: the same as those in the $\lambda_{\mathrm{c}}2$-calculus.

Axioms: All the axioms in the λ_{c}-calculus (with E extended as above), and:

$$(\Lambda\alpha.M)\,\sigma = M\,[\sigma/\alpha] \qquad \Lambda\alpha.M\alpha = M \quad (\alpha \notin \mathrm{FTV}\,(M))$$

$\lambda_{\mathbf{c}}2$-calculus vs. $\lambda_{\mathbf{c}}2_\eta$-calculus

$\lambda_c 2$-calculus:

Types $\sigma ::= \ldots \mid \alpha \mid \forall\alpha.\sigma$ Values $V ::= \ldots \mid \Lambda\alpha.M$

Terms $M ::= \ldots \mid \Lambda\alpha.M \mid M\sigma$ Evaluation Contexts $E ::= \ldots \mid E\sigma$

$$(\Lambda\alpha.M)\,\sigma = M\,[\sigma/\alpha] \qquad \Lambda\alpha.V\alpha = V \quad (\alpha \notin \mathrm{FTV}\,(V))$$

$\lambda_c 2_\eta$-calculus:

Types $\sigma ::= \ldots \mid \alpha \mid \forall\alpha.\sigma$ Values $V ::= \ldots \mid \Lambda\alpha.V \mid V\sigma$

Terms $M ::= \ldots \mid \Lambda\alpha.M \mid M\sigma$ Evaluation Contexts $E ::= \ldots \mid E\sigma$

$$(\Lambda\alpha.M)\,\sigma = M\,[\sigma/\alpha] \qquad \Lambda\alpha.M\alpha = M \quad (\alpha \notin \mathrm{FTV}\,(M))$$

(cf. ML)

The Class of Monads

$$T ::= \gamma \mid 1 \mid T \times T \mid T^C \mid \nu\gamma.T \mid C \mid T \oplus T \mid \mu\gamma.T$$

Lifting:	$(-) \oplus 1$
Exception:	$(-) \oplus E$
Global State:	$((-) \times S)^S$
Input:	$\mu\beta.(-) \oplus \beta^U$
Output:	$\mu\beta.(-) \oplus (U \times \beta)$
List:	$\mu\beta.1 \oplus (-) \times \beta$

Categorical Semantics

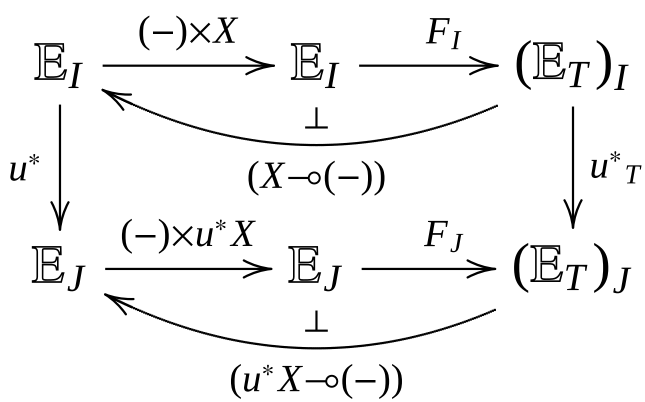
Polymorphic λ_c-models

Definition 1. *A* polymorphic λ_c-model *consists of*

(i) *a cartesian polymorphic fibration, i.e.* *a fibration $p : \mathbb{E} \longrightarrow \mathbb{B}$ which has products in the base category, generic object Ω and fibred products,*

(ii) *a fibred strong monad T on p satisfying the equalizing requirement fiberwise, and*

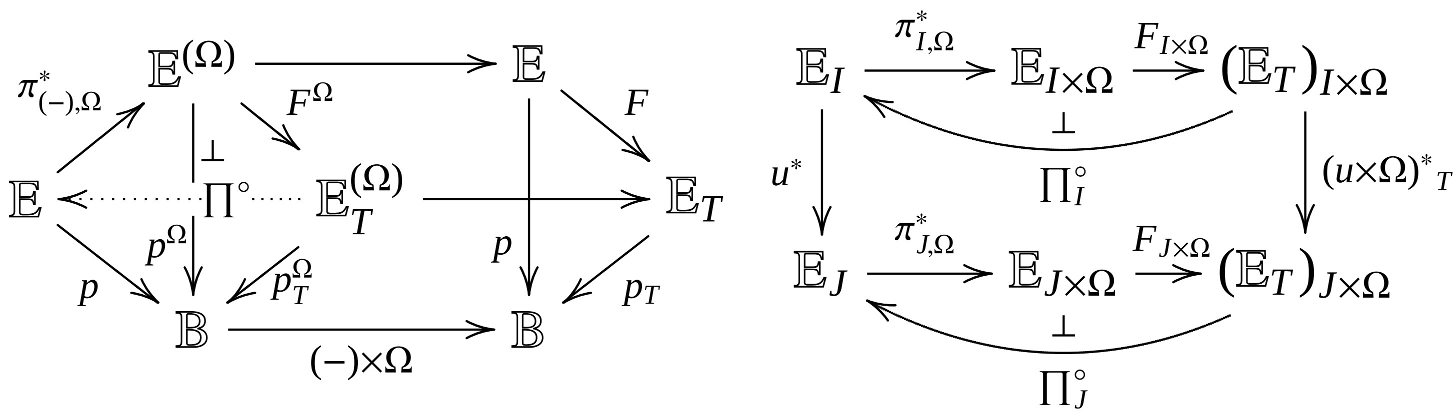
(iii) *fibred Kleisli exponentials:*



λ_{c}2-model

Definition 2. *Let p be a polymorphic λ_c-model,
F^{Ω} be the change-of-base of the Kleisli embedding F along $(-)\times\Omega$,
and $\pi^*_{(-),\Omega}$ be the "weakening" fibred functor.*

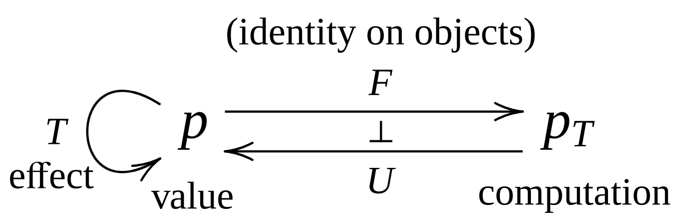
- Kleisli simple Ω-products $\prod^{\circ}$ *is a fibred right adjoint functor to the composite of F^{Ω} and $\pi^*_{(-),\Omega}$, and*
- *a* λ_{c}2-model *is a polymorphic λ_c-model p which has Kleisli simple Ω-products.*



$\lambda_{\mathbf{c}}2_\eta$-models and Monadic $\lambda_{\mathbf{c}}2_\eta$-models

Definition 3. *A* $\lambda_c 2_\eta$-model *is a polymorphic* λ_c*-model* p *whose Kleisli fibration* p_T *has simple* Ω*-products.*

Definition 4. *A* monadic $\lambda_c 2_\eta$-model *is a polymorphic* λ_c*-model such that* p *and* p_T *has simple* Ω*-products and the Kleisli embedding* $F : p \longrightarrow p_T$ *preserves them.*



A Characterization of Monadic $\lambda_{\mathbf{c}}2_\eta$-model

Lemma 5.

- *Let p be a polymorphic λ_c-model s.t. p has simple Ω-products.*
  p is a monadic $\lambda_c 2_\eta$-model,
  (i.e., p_T has simple Ω-products and $F : p \longrightarrow p_T$ preserves simple Ω-products,) if and only if
  the underlying fibred endofunctor of the monad T preserves simple Ω-products.

- *For a $\lambda 2$-fibration and*
  a fibred strong monad on it satisfying the equalizing requirement,
  if the underlying endofunctor preserves simple Ω-products,
  then they form a monadic $\lambda_c 2_\eta$-model.

Concrete Models

Classes of Monads for Monadic $\lambda_{\mathbf{c}}2_{\eta}$-models

$$T ::= \gamma \,|\, 1 \,|\, T \times T \,|\, T^C \,|\, \nu\gamma.T \tag{1}$$

$$T ::= \gamma \,|\, 1 \,|\, T \times T \,|\, T^C \,|\, \nu\gamma.T \,|\, C \tag{2}$$

$$T ::= \gamma \,|\, 1 \,|\, T \times T \,|\, T^C \,|\, \nu\gamma.T \,|\, C \,|\, T \oplus T \,|\, \mu\gamma.T \tag{3}$$

(1) universality

(2) universality + parametricity

(3) universality + parametricity + additional terms (fixed-point operator)

Products and Powers

Lemma 6. *Let $\mathbb{B}$ be a cartesian category, and K be any object of $\mathbb{B}$.*

(1) The subcategory of $\mathbf{Fib}_{\mathbb{B}}$ consisting of all fibrations having simple K-products, and all fibred functors preserving simple K-products, is cartesian subcategory.

(2) Let p be a fibration having simple K-products and fibred finite products. Then the fibred functors $\times : p \times p \longrightarrow p$, and $1 : 1 \longrightarrow p$ preserve simple K-products.

(3) Moreover assume that p is a fibred CCC. Then for any X over 1 in p, the "power" fibred functor $X \Rightarrow (-) : p \longrightarrow p$ preserves simple K-products.

Final Coalgebras

Definition 7. *For a fibred functor $F : q \longrightarrow p{\Rightarrow}p$,
if F has pointwise final coalgebras and the reindexings preserve them,
then we can define the "final coalgebras" fibred functor $fcF : q \longrightarrow p$.*

Lemma 8. *Let $\mathbb{B}$ be a cartesian category, K be an object of $\mathbb{B}$,
p, q be fibrations having simple K-products, and
$F : q \times p \longrightarrow p$ be a fibred functor such that
its transposition $\lambda F : q \longrightarrow p{\Rightarrow}p$ has final coalgebras as above.*

*Then, if F preserves simple K-products,
the fibred functor $fc\,(\lambda F) : q \longrightarrow p$ also preserves simple K-products.*

(Typically $q = p^n$.)

Constants

For a $\lambda 2$-fibration p and any object X over 1,
the “constant” fibred functor $X : 1 \longrightarrow p$ preserves simple Ω-products
if we assume parametricity.

This is possible by all already known types of parametricity including
relational parametricity, linear parametricity, and focal parametricity.

At this point, we can construct global state monads $((-) \times S)^S$
and output monads $\mu\beta.\,(-) + (U \times \beta) \cong (-) \times (\mu\beta.1 + (U \times \beta))$.

Coproducts

$$
\begin{aligned}
&\lambda u\colon (\forall\alpha.\,(\sigma+\tau))\,.\\
&\quad \text{case } (u1) \text{ of } \mathrm{in}_0\, a' \to \mathrm{in}_0\,(\Lambda\alpha.\text{case } (u\alpha) \text{ of } \mathrm{in}_0\, a \to a\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad |\mathrm{in}_1\, b \to \text{“this case nothing”})\\
&\qquad\qquad\quad |\mathrm{in}_1\, b' \to \mathrm{in}_1\,(\Lambda\alpha.\text{case } (u\alpha) \text{ of } \mathrm{in}_0\, a \to \text{“this case nothing”}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad |\mathrm{in}_1\, b \to b \qquad\qquad\qquad\qquad)
\end{aligned}
$$

$$: \forall\alpha.\,(\sigma+\tau) \to \forall\alpha.\sigma + \forall\alpha.\tau$$

$\longrightarrow$ We use $\perp$ for “this case nothing”.

Initial Algebras

$$\text{let } f : ((\forall\alpha.\mu\beta.1 + \sigma \times \beta) \longrightarrow (\mu\beta.1 + \forall\alpha.\sigma \times \beta)) =$$

$$\lambda u : (\forall\alpha.\mu\beta.1 + \sigma \times \beta).\text{case } (u1) \text{ of Nil} \rightarrow \text{Nil}$$

$$|\text{Cons}(a', as') \rightarrow \text{Cons}(\Lambda\alpha.\text{case } (u\alpha) \text{ of Nil} \rightarrow \text{“this case nothing”}$$

$$|\text{Cons}(a, as) \rightarrow a\ ,$$

$$f(\Lambda\alpha.\text{case } (u\alpha) \text{ of Nil} \rightarrow \text{“this case nothing”}$$

$$|\text{Cons}(a, as) \rightarrow as\ ))$$

$\longrightarrow$ We use a fixed-point operator.

Linear Parametric Models

$$p :=\ l_! \underset{U}{\overset{L}{\underset{\perp}{\rightleftarrows}}} l \circlearrowright ! \qquad\qquad T \circlearrowright p \underset{U_T}{\overset{F_T}{\underset{\perp}{\rightleftarrows}}} p_T$$

l :a PILL_Y model (Birkedal et al. 2008),

i.e., $\lambda 2$-version of linear categories

which can also model fixed-point operators

Assumption. *L preserves simple Ω-products.*

Relevant Parametricity

Contraction term formation rule:

$$\frac{\Xi \mid \Gamma; \Delta, x : \sigma, y : \sigma \vdash M : \tau}{\Xi \mid \Gamma; \Delta, x : \sigma \vdash M\,[x/y] : \tau}$$

and, moreover

$$\frac{\rho \subseteq_{\mathrm{adm}} \sigma \times \tau \qquad\qquad M : \sigma' \times \tau' \multimap \sigma \times \tau}{(\text{“}M^{-1}\rho\text{”} =)\,(x : \sigma', y : \tau')\,.\rho\,(\pi_1 M\,\langle x, y\rangle\,, \pi_2 M\,\langle x, y\rangle) \subseteq_{\mathrm{adm}} \sigma' \times \tau'}$$

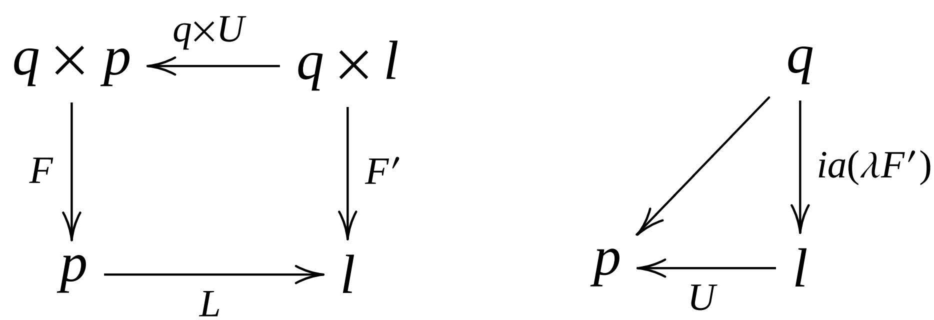
Separated Sums

Proposition 9. *For a model satisfying the assumption thus far, its separated sums preserve simple* Ω*-products.*

$$\begin{array}{ccc} p^2 & \xrightarrow{L^2} & l^2 \\ {\scriptstyle \oplus}\downarrow & & \downarrow{\scriptstyle +} \\ p & \xleftarrow[U]{} & l \end{array}$$

Linearly Initial Algebras

Proposition 10. *For a model* l *satisfying the assumption thus far, let* q *be a fibration which has the same base category and generic object* Ω *as those of* l *and* p*, and has simple* Ω*-products.*
Then for a fibred functor $F : q \times p \longrightarrow p$ *preserving simple* Ω*-products, if the transposition* $\lambda F'$ *has initial algebras and also final coalgebras, then the* linearly initial algebras *fibred functor* $U \circ ia\,(\lambda F') : q \longrightarrow p$ *also preserves simple* Ω*-products.*



Summary and Example

$$T ::= \gamma \,|\, 1 \,|\, T \times T \,|\, T^{C} \,|\, \nu\gamma.T \,|\, C \,|\, T \oplus T \,|\, \mu\gamma.T$$

Theorem 11. *For a model l satisfying the assumption thus far, let T be a fibred strong monad on p satisfying the equalizing requirement. If T is constructed inductively as above, then T and the $\lambda 2$-fibration p form a monadic $\lambda_c 2_\eta$-model.*

Example 12. *($PFam(AP(D)_\perp)$, $PFam(AP(D))$, ...) : the models of linear Plotkin-Abadi logic constructed from domain theoretic PERs (L. Birkedal, R. E. Møgelberg, and R. L. Petersen, 2007)*

Future Work

- Dependent type theoretical approach is also available.
- powerset monads?
  continuations monads?
- all lifting monads?
  all commutative relevant monads?