
Parametric Domain-theoretic Models of Linear/Intuitionistic Polymorphic Lambda Calculus

Lars Birkedal

The IT University of Copenhagen

Joint work with Rasmus Møgelberg and Rasmus Lerchedahl Petersen



Second-order LNL

- ◆ Types: $\sigma ::= \alpha \mid 1 \mid \sigma \otimes \tau \mid \sigma \multimap \tau \mid !\sigma \mid \prod \alpha. \sigma$
- ◆ Expressions of form

$$\alpha_1, \dots, \alpha_l \mid x_1 : \sigma_1, \dots, x_n : \sigma_n; a_1 : \sigma'_1, \dots, a_m : \sigma'_m \vdash t : \tau$$

- ◆ Terms: LNL + abstraction over types, type application, and fixed point combinator Y

$$\frac{}{\vec{\alpha} \mid \Gamma; - \vdash Y : \prod \alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha}$$

where $\sigma \rightarrow \tau = !\sigma \multimap \tau$.



Parametricity and Domain Theory

- ◆ Important for data abstraction
- ◆ Useful metalanguage for domain-theory [Plotkin 1994]
 - In particular: Not only inductive and coinductive datatypes, but also *recursive* types



Encoding

◆ Inductive:

$$\mu\alpha. \sigma(\alpha) = \prod\alpha. (\sigma(\alpha) \multimap \alpha) \rightarrow \alpha.$$

◆ Coinductive:

$$\nu\alpha. \sigma(\alpha) = \prod\beta. \prod\alpha. (!(\alpha \multimap \sigma(\alpha)) \otimes \alpha \multimap \beta) \multimap \beta$$

◆ Recursive: given $\sigma(\alpha, \beta)$

- $\omega(\alpha) = \mu\beta. \sigma(\alpha, \beta)$
- $\tau' = \nu\alpha. \sigma(\omega(\alpha), \alpha)$
- $\tau = \omega(\tau')$
- $\tau \cong \sigma(\tau, \tau)$



Linear Abadi & Plotkin logic, I

- ◆ Abadi & Plotkin (1993) proposed a logic for reasoning about parametricity.
- ◆ Plotkin (1994) suggested it could be extended to reason about second-order LNL
- ◆ Allows to formulate parametricity as a schema and prove expected consequences, including recursive types for definable functors.
- ◆ We will define parametric models to be models of Linear Abadi & Plotkin logic satisfying parametricity.



Linear Abadi & Plotkin logic, II

◆ Expressions:

$$\vec{\alpha} \mid \vec{x} : \vec{\sigma} \mid \vec{R} \subset \vec{\tau} \times \vec{\tau}' \mid \vec{S} \subset^{\mathbf{Adm}} \vec{\omega} \times \vec{\omega}' \vdash \phi : \text{Prop}$$

where $\vec{\alpha} \mid \vec{x} : \vec{\sigma}; -$ is a context of second-order LNL

◆ Propositions:

$$\begin{aligned} \phi ::= & (t =_{\sigma} u) \mid \rho(t, u) \mid \phi \supset \psi \mid \perp \mid \top \mid \phi \wedge \psi \mid \phi \vee \psi \mid \\ & \forall \alpha : \text{Type}. \phi \mid \forall x : \sigma. \phi \mid \forall R \subset \sigma \times \tau. \phi \mid \\ & \forall S \subset^{\mathbf{Adm}} \sigma \times \tau. \phi \mid \exists \alpha : \text{Type}. \phi \mid \exists x : \sigma. \phi \mid \\ & \exists R \subset \sigma \times \tau. \phi \mid \exists S \subset^{\mathbf{Adm}} \sigma \times \tau. \phi \end{aligned}$$

for ρ a definable relation.



Definable relations, I

- ◆ Definable relations

$$\rho ::= R \mid (x : \sigma, y : \tau). \phi$$

- ◆ Formation rules

$$\frac{}{\Xi \mid \Gamma \mid \Theta, R \subset \sigma \times \tau \vdash R \subset \sigma \times \tau}$$

$$\frac{\Xi \mid \Gamma, x : \sigma, y : \tau \mid \Theta \vdash \phi : \text{Prop}}{\Xi \mid \Gamma \mid \Theta \vdash (x : \sigma, y : \tau). \phi \subset \sigma \times \tau}$$

$$\frac{\Xi \mid \Gamma \mid \Theta \vdash \rho \subset^{\text{Adm}} \sigma \times \tau}{\Xi \mid \Gamma \mid \Theta \vdash \rho \subset \sigma \times \tau}$$



Definable relations, III

$$\frac{\begin{array}{c} \alpha_1, \dots, \alpha_n \vdash \sigma(\vec{\alpha}) : \text{Type} \\ \Xi \mid \Gamma \mid \Theta \vdash \rho_1 \subset^{\text{Adm}} \tau_1 \times \tau'_1, \dots, \rho_n \subset^{\text{Adm}} \tau_n \times \tau'_n \end{array}}{\Xi \mid \Gamma \mid \Theta \vdash \sigma[\vec{\rho}] \subset^{\text{Adm}} \sigma(\vec{\tau}) \times \sigma(\vec{\tau}')}$$



Definable relations, IV

- ◆ Instantiating definable relations

$$\frac{\Xi \mid \Gamma \mid \Theta \vdash \rho \subset \sigma \times \tau \quad \Xi \mid \Gamma; - \vdash t : \sigma, s : \tau}{\Xi \mid \Gamma \mid \Theta \vdash \rho(t, s) : \text{Prop}}$$

- ◆ Also write $t\rho s$ for $\rho(t, s)$.



Definable relations, examples

- ◆ The graph of a function $f: \sigma \multimap \tau$

$$\langle f \rangle = (x: \sigma, y: \tau). f(x) =_{\tau} y$$

- ◆ The equality relation eq_{σ} defined as the graph of the identity map.



Remarks

- ◆ In Abadi and Plotkins original article, $\sigma[\vec{\rho}]$ was defined inductively on the structure of σ .
- ◆ We would like to consider $\sigma[\vec{\rho}]$ as a construction in the internal language of the model (also work for types that are not inductively defined).
- ◆ In our version, we capture the inductive definition by axioms of the logic, e.g.,

$$\frac{\vec{\alpha} \vdash \sigma \multimap \sigma' : \text{Type} \quad \Xi \mid \Gamma \mid \Theta \vdash \vec{\rho} \subset^{\mathbf{Adm}} \vec{\tau} \times \vec{\tau}'}{\Xi \mid \Gamma \mid \Theta \mid \top \vdash (\sigma \multimap \sigma')[\vec{\rho}] \equiv (\sigma[\vec{\rho}] \multimap \sigma'[\vec{\rho}])},$$

where $\rho \multimap \rho'$ is

$$(f : \sigma \multimap \sigma', g : \tau \multimap \tau'). \forall x : \sigma \forall y : \tau. \rho(x, y) \supset \rho'(fx, gy)$$



Admissible Relations

$$\frac{}{\Xi \mid \Gamma \mid \Theta \vdash \mathit{eq}_\sigma \subset^{\mathbf{Adm}} \sigma \times \sigma}$$

$$\frac{\Xi \mid \Gamma \mid \Theta \vdash \rho \subset^{\mathbf{Adm}} \sigma \times \tau \quad \Xi \mid \Gamma \mid \Theta \vdash t: \sigma' \multimap \sigma, u: \tau' \multimap \tau}{\Xi \mid \Gamma \mid \Theta \vdash (x: \sigma', y: \tau'). \rho(t x, u y) \subset^{\mathbf{Adm}} \sigma' \times \tau'}$$

$$\frac{\Xi \mid \Gamma \mid \Theta \vdash \rho \subset^{\mathbf{Adm}} \sigma \times \tau \quad \Xi \mid \Gamma \mid \Theta \vdash \phi: \mathbf{Prop} \quad x, y \notin \mathit{Dom}(\Gamma)}{\Xi \mid \Gamma \mid \Theta \vdash (x: \sigma, y: \tau). \phi \supset \rho(x, y) \subset^{\mathbf{Adm}} \sigma \times \tau}$$

$$\frac{\Xi, \alpha \mid \Gamma \mid \Theta \vdash \rho \subset^{\mathbf{Adm}} \sigma \times \tau \quad \Xi \mid \Gamma \mid \Theta \quad \Xi \vdash \sigma: \mathbf{Type} \quad \Xi \vdash \tau: \mathbf{Type}}{\Xi \mid \Gamma \mid \Theta \vdash (x: \sigma, y: \tau). \forall \alpha: \mathbf{Type}. \rho(x, y) \subset^{\mathbf{Adm}} \sigma \times \tau}$$



Parametricity

- ◆ Identity extension:

$$\forall \vec{\alpha}: \text{Type}. \sigma[\mathbf{eq}_{\vec{\alpha}}] \equiv \mathbf{eq}_{\sigma(\vec{\alpha})}$$

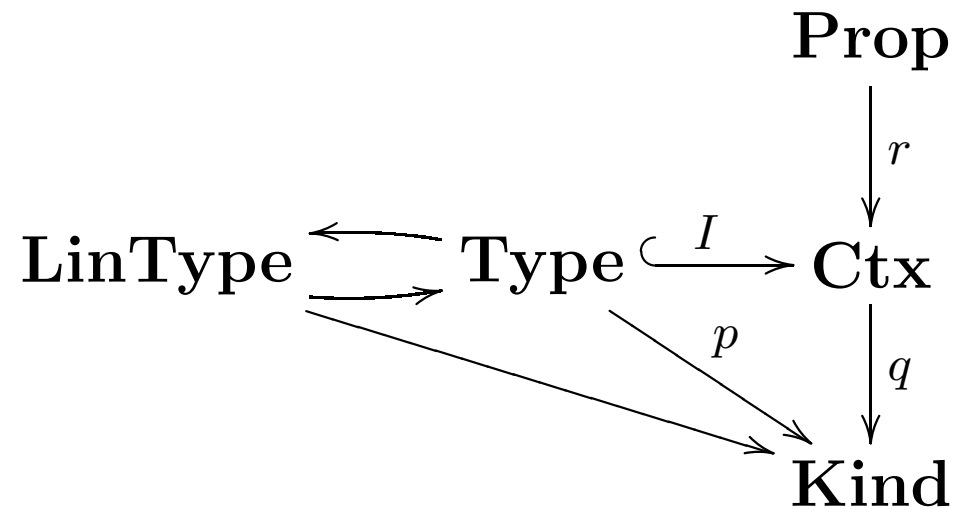
- ◆ Extensionality:

$$\begin{aligned} (\forall x: \sigma. t x =_{\tau} u x) &\supset t =_{\sigma \rightarrow \tau} u \\ (\forall \alpha: \text{Type}. t \alpha =_{\tau} u \alpha) &\supset t =_{\prod \alpha: \text{Type}. \tau} u. \end{aligned}$$



LAPL-structures

A LAPL-structure consists of categories and functors:

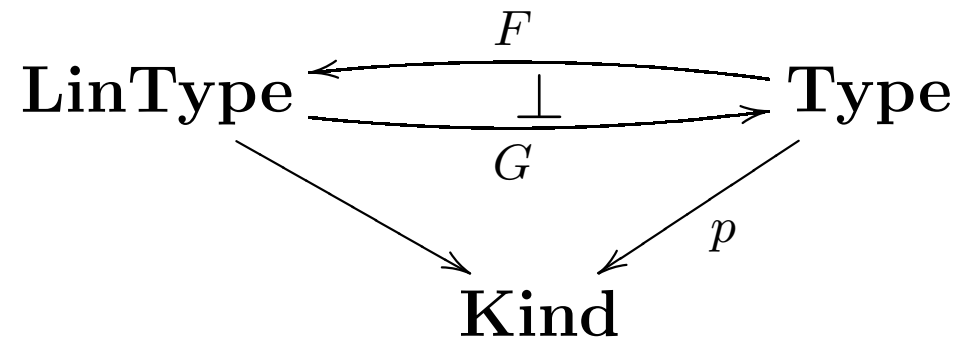


such that



LAPL-structures, I

- ◆ the diagram



is a model of second-order LNL + Y .



LAPL-structures, II

- ◆ (r, q) is an indexed first-order logic fibration which has products and coproducts with respect to projections $\Xi \times \Omega \rightarrow \Xi$ in \mathbf{Kind} , where Ω is the generic object of p .
- ◆ I is a faithful product-preserving map of fibrations.
- ◆ q is a fibration with fibred products



LAPL-structures, III

a contravariant morphism of fibrations:

$$\begin{array}{ccc} \mathbf{LinType} \times_{\mathbf{Kind}} \mathbf{LinType} & \xrightarrow{U} & \mathbf{Ctx} \\ & \searrow & \swarrow \\ & \mathbf{Kind} & \end{array}$$

such that

- ◆ $U(\sigma, \tau)$ is powerset of σ and τ in $\mathbf{LinType}_{\Xi}$, i.e.,

$$\Psi_{\Xi} : \mathbf{Hom}_{\mathbf{Ctx}_{\Xi}}(\xi, U(\sigma, \tau)) \rightarrow \mathbf{Obj}(\mathbf{Prop}_{\xi \times I(G(\sigma) \times G(\tau))})$$



LAPL-structures, IV

To interpret admissible relations, we require subfunctor V of U , which we think of as the subset of all admissible relations, closed under all the rules for creating admissible relations.



LAPL-structures, V

A context

$$\Xi \mid x_1 : \sigma_1, \dots, x_n : \sigma_n \mid R_1 \subset \tau_1 \times \tau'_1, \dots, R_m \subset \tau_m \times \tau'_m \mid \\ S_1 \subset^{\mathbf{Adm}} \omega_1 \times \omega'_1, \dots, S_k \subset^{\mathbf{Adm}} \omega_k \times \omega'_k$$

is interpreted as

$$\prod_i IG(\llbracket \sigma_i \rrbracket) \times \prod_j U(\llbracket \tau_j \rrbracket, \llbracket \tau'_j \rrbracket) \times \prod_l V(\llbracket \omega_l \rrbracket, \llbracket \omega'_l \rrbracket)$$

where the interpretations of the types is the usual interpretation of types in $\mathbf{LinType} \rightarrow \mathbf{Kind}$.



Fibration of relations, I

- ◆ Given the above data, one can construct a fibration

$$\begin{array}{c} \text{AdmRelations} \\ \downarrow \\ \text{AdmRelCtx.} \end{array}$$

- ◆ AdmRelCtx is given as the pullback

$$\begin{array}{ccc} \text{AdmRelCtx} & \longrightarrow & \text{Ctx} \\ \langle \partial_0, \partial_1 \rangle \downarrow & \lrcorner & \downarrow \\ \text{Kind} \times \text{Kind} & \xrightarrow{\times} & \text{Kind} \end{array}$$

We write an object Θ in AdmRelCtx over (Ξ, Ξ') as $\Xi, \Xi' \mid \Theta$.



Fibrations of relations, II

- ◆ Objects of `AdmRelations` over an object $\Xi, \Xi' \mid \Theta$ are

$$\Xi; \Xi' \mid \Theta \vdash \rho \subset^{\text{Adm}} \sigma \times \tau.$$

- ◆ A vertical morphisms in `AdmRelations` from $\rho \subset^{\text{Adm}} \sigma \times \tau$ to $\rho' \subset^{\text{Adm}} \sigma' \times \tau'$ is a pair of morphisms $f: \sigma \multimap \tau$, $g: \sigma' \multimap \tau'$ in `LinType` such that in the internal language the formula

$$\forall x: \sigma, y: \tau. \rho(x, y) \supset \rho'(f\ x, g\ y)$$

holds.



Relational interpretation of types

To interpret $\sigma[\rho]$ we will require that the graph

$$\left(\begin{array}{c} \text{AdmRelations} \\ \downarrow \\ \text{AdmRelCtx} \end{array} \right) \begin{array}{c} \xrightarrow{\partial_0} \\ \xrightarrow{\partial_1} \end{array} \left(\begin{array}{c} \text{LinType} \\ \downarrow \\ \text{Kind} \end{array} \right) .$$

has an extension to a reflexive graph of linear λ_2 -fibrations.



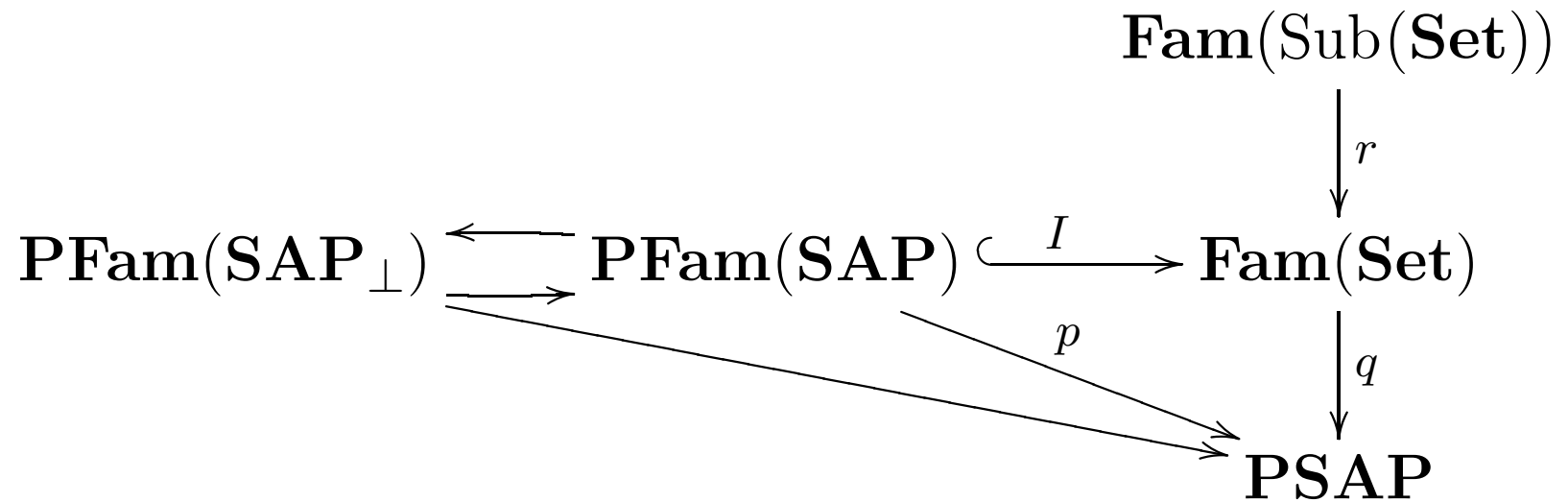
Soundness and completeness

- ◆ The presented model is sound.
- ◆ Completeness is proved by constructing a syntactical model.
- ◆ A *parametric LAPL-structure* is an LAPL-structure, with very strong equality in which identity extension and extensionality holds in the internal logic.
- ◆ **Theorem:** In a parametric LAPL-structure, for any polymorphically strong fibred functor $F : \mathbf{LinType}^{\text{op}} \times \mathbf{LinType} \rightarrow \mathbf{LinType}$, there exists a closed type τ such that $F(\tau, \tau) \cong \tau$.



A concrete model, I

- ◆ The expected model does indeed work:



A concrete model, II

- ◆ PSAP has natural numbers as objects, and a morphism $n \rightarrow m$ is an m -vector of objects in $\mathbf{PFam}(\mathbf{SAP}_\perp)$.
- ◆ Over 1, objects of $\mathbf{PFam}(\mathbf{SAP})$ are functions
 - $f^p: \mathbf{SAP}_\perp \rightarrow \mathbf{SAP}_\perp$
 - $f^r: \prod_{\sigma, \tau \in \mathbf{SAP}_\perp} (\text{RegSub}(\sigma \times \tau) \rightarrow \text{RegSub}(f^p(\sigma) \times f^p(\tau)))$satisfying $f^r(\text{eq}_\sigma) = \text{eq}_{(f^p(\sigma))}$, for all $\sigma \in \mathbf{SAP}_\perp$.
- ◆ Morphisms in $\mathbf{PFam}(\mathbf{SAP})$ have continuous realizers, preserving relations
- ◆ Morphisms in $\mathbf{PFam}(\mathbf{SAP}_\perp)$ have strict continuous realizers, preserving relations



Other concrete models

(back-of-the-envelope calculations only)

- ◆ Rosolini and Simpson model (IST)
- ◆ Model based on Lily (Pitts et. al.)



Internal models

- ◆ the definition of LAPL-structure is fairly long
- ◆ for an internal LNL-model in a quasi-topos with a Lawvere-Tierney topology j , much of the LAPL-structure can be derived



Summary

- ◆ Precise definition of LAPL and proofs of expected properties
- ◆ Definition of parametric LAPL-structure, sound and complete
- ◆ Concrete model
- ◆ Ongoing and future work:
 - Adequacy of Lily-like language a la Rosolini and Simpson
 - Completion process
 - References and other computational effects

