Mixed Linear and Non-linear Recursive Types

Vladimir Zamdzhiev

Université de Lorraine, CNRS, Inria, LORIA, F 54000 Nancy, France

Joint work with Michael Mislove and Bert Lindenhovius

Applied Category Theory 2019
University of Oxford
19 July 2019

Introduction

- Mixed linear/non-linear type systems have recently found applications in:
  - concurrency (session types for π-calculus);
  - quantum programming (substructural limitations imposed by quantum information);
  - circuit description languages (dealing with wires of string diagrams);
  - programming resource-sensitive data (file handlers, etc.).
- This talk: add recursive types to a mixed linear/non-linear type system.
- Very detailed denotational (and categorical) treatment:
  - a new technique for solving recursive domain equations within **CPO**;
  - coherence theorems for (parameterised) initial algebras;
  - we describe the canonical comonoid structure of recursive types;
  - sound and adequate categorical models.
- Paper to appear in ICFP'19, arxiv:1906.09503.

Long story short

- Syntax and operational semantics is mostly straightforward and is based on prior work[1].
- Main difficulty is on the denotational and categorical side.
- How can we copy/discard non-linear recursive types *implicitly*?
  - A list of qubits (or file handlers) should be *linear* – cannot copy/discard.
  - A list of natural numbers should be *non-linear* – can copy/discard at will (and implicitly).
- For the rest of the talk we focus on the linear/non-linear type structure.
- How do we design a linear/non-linear fixpoint calculus (LNL-FPC)?

[1] Rios and Selinger, QPL'17; Lindenhovius, Mislove and Zamdzhiev LICS'18

Syntax

Type variables	X, Y, Z		
Term variables	x, y, z		
Types	A, B, C	$::=$	$X \mid A + B \mid A \otimes B \mid A \multimap B \mid !A \mid \mu X.A$
Non-linear types	P, R	$::=$	$X \mid P + R \mid P \otimes R \mid !A \mid \mu X.P$
Type contexts	Θ	$::=$	$X_1, X_2, \ldots, X_n$
Term contexts	Γ, Σ	$::=$	$x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n$
Non-linear term contexts	Φ	$::=$	$x_1 : P_1, x_2 : P_2, \ldots, x_n : P_n$
Terms	m, n, p	$::=$	$x \mid \mathrm{left}_{A,B}\, m \mid \mathrm{right}_{A,B}\, m$
			$\mid \mathrm{case}\ m\ \mathrm{of}\ \{\mathrm{left}\ x \to n\ \mathrm{right}\ y \to p\}$
			$\mid \langle m, n \rangle \mid \mathrm{let}\ \langle x, y \rangle = m\ \mathtt{in}\ n \mid \lambda x^A.m \mid mn$
			$\mid \mathrm{lift}\ m \mid \mathrm{force}\ m \mid \mathtt{fold}_{\mu X.A} m \mid \mathtt{unfold}\ m$
Values	v, w	$::=$	$x \mid \mathrm{left}_{A,B}\, v \mid \mathrm{right}_{A,B}\, v \mid \langle v, w \rangle \mid \lambda x^A.m$
			$\mid \mathrm{lift}\ m \mid \mathtt{fold}_{\mu X.A} v$

Operational Semantics

$$\frac{}{x \Downarrow x} \qquad \frac{m \Downarrow v}{\mathsf{left}\ m \Downarrow \mathsf{left}\ v} \qquad \frac{m \Downarrow v}{\mathsf{right}\ m \Downarrow \mathsf{right}\ v}$$

$$\frac{m \Downarrow \mathsf{left}\ v \qquad n[v/x] \Downarrow w}{\mathsf{case}\ m\ \mathtt{of}\ \{\mathsf{left}\ x \to n \mid \mathsf{right}\ y \to p\} \Downarrow w} \qquad \frac{m \Downarrow \mathsf{right}\ v \qquad p[v/y] \Downarrow w}{\mathsf{case}\ m\ \mathtt{of}\ \{\mathsf{left}\ x \to n \mid \mathsf{right}\ y \to p\} \Downarrow w}$$

$$\frac{m \Downarrow v \qquad n \Downarrow w}{\langle m, n\rangle \Downarrow \langle v, w\rangle} \qquad \frac{m \Downarrow \langle v, v'\rangle \qquad n[v/x, v'/y] \Downarrow w}{\mathsf{let}\ \langle x, y\rangle = m\ \mathsf{in}\ n \Downarrow w}$$

$$\frac{}{\lambda x.m \Downarrow \lambda x.m} \qquad \frac{m \Downarrow \lambda x.m' \qquad n \Downarrow v \qquad m'[v/x] \Downarrow w}{mn \Downarrow w}$$

$$\frac{}{\mathsf{lift}\ m \Downarrow \mathsf{lift}\ m} \qquad \frac{m \Downarrow \mathsf{lift}\ m' \qquad m' \Downarrow v}{\mathsf{force}\ m \Downarrow v} \qquad \frac{m \Downarrow v}{\mathtt{fold}\ m \Downarrow \mathtt{fold}\ v} \qquad \frac{m \Downarrow \mathtt{fold}\ v}{\mathtt{unfold}\ m \Downarrow v}$$

Some derived types and terms

- $0 \equiv \mu X.X$ is the empty type (non-linear).
- $I \equiv\ !(0 \multimap 0)$ is the unit type (non-linear).
- $* \equiv \texttt{lift}\ \lambda x^0.x : I$ is the canonical value of unit type (non-linear).
- $\mathrm{Nat} \equiv \mu X.I + X$ is the type of natural numbers (non-linear).
- $\mathrm{zero} \equiv \texttt{fold left}\ * : \mathrm{Nat}$ is the zero natural number, which is a non-linear value.
- $\texttt{succ} \equiv \lambda n.\texttt{fold right}\ n : \mathrm{Nat} \multimap \mathrm{Nat}$ is the successor function.
- $\mathrm{List\ Nat} \equiv \mu X.I + \mathrm{Nat} \otimes X$ is the type of lists of natural numbers (non-linear).
- $\mathrm{List\ Qubit} \equiv \mu X.I + \mathrm{Qubit} \otimes X$ is the type of lists of qubits (linear).
- $\mathrm{Stream\ Qubit} \equiv \mu X.\mathrm{Qubit} \otimes\ !X$ is the type of streams of qubits (linear).

Term level recursion

In FPC, a term-level recursion operator may be defined using fold/unfold terms. The same is true for LNL-FPC.

Theorem

The term-level recursion operator from[2] *is now a derived rule. For a given term* $\Phi, z :!A \vdash m : A$*, define:*

$$\alpha_m^z \equiv \mathit{lift}\ \mathit{fold}\ \lambda x^{!\mu X.(!X \multimap A)}.(\lambda z^{!A}.m)(\mathit{lift}\ (\mathit{unfold}\ \mathit{force}\ x)x)$$

$$\mathit{rec}\ z^{!A}.m \equiv (\mathit{unfold}\ \mathit{force}\ \alpha_m^z)\alpha_m^z$$

[2] Lindenhovius, Mislove, Zamdzhiev: Enriching a Linear/Non-linear Lambda Calculus: A Programming Language for String Diagrams. LICS 2018

Example: functorial function

```
rec fact. λ n.
 case unfold n of
    left u -> succ zero
    right n’ -> mult(n, (force fact) n’)
```

Remark

The above program is written in the formal syntax without syntactic sugar. Note: implicit rules for copying and discarding.

A recap on ω-categories[3].

- A functor $F : \mathbf{A} \to \mathbf{C}$ is a (strict) ω-*functor* if it preserves ω-colimits (and the initial object).
- ω-functors are closed under composition and pairing, that is, if F and G are ω-functors, then so are $F \circ G$ and $\langle F, G \rangle$.
- A category $\mathbf{C}$ is an ω-*category* if it has an initial object and all ω-colimits.
- ω-categories are perfectly suited for computing *parameterised initial algebras*.

[3] Lehmann and Smyth 1981

Baby's first parameterised initial algebra definition

Definition

Let $\mathbf{B}$ be an ω-category and let $T : \mathbf{A} \times \mathbf{B} \to \mathbf{B}$ be an ω-functor. A parameterised initial algebra $(T^\dagger, \phi^T)$ consists of:

- An ω-functor $T^\dagger : \mathbf{A} \to \mathbf{B}$;
- A natural isomorphism $\phi^T : T \circ \langle \mathrm{Id}, T^\dagger \rangle \Rightarrow T^\dagger : \mathbf{A} \to \mathbf{B}$.
- characterised by the property that $(T^\dagger A, \phi^T_A)$ is the initial $T(A, -)$-algebra.

Remark

Parameterised initial algebras are necessary to interpret recursive types defined by nested recursion (also known as mutual recursion).

Coherence Properties for Parameterised Initial Algebras

Theorem

Let $\mathbf{A}$ *and* $\mathbf{C}$ *be categories and let* $\mathbf{B}$ *and* $\mathbf{D}$ *be* ω*-categories. Let*

$$\alpha : T \circ (N \times M) \Rightarrow M \circ H$$

be a natural isomorphism, where H *and* T *are* ω*-functors and where* M *is a strict* ω*-functor. Then, the natural isomorphism* α *induces a natural isomorphism*

$$\alpha^{\dagger} : T^{\dagger} \circ N \Rightarrow M \circ H^{\dagger} : \mathbf{A} \to \mathbf{D},$$

which satisfies some important coherence conditions (omitted here).

How to see a mixed-variance functor as a covariant one

Definition

Given a **CPO**-category **C**, its *subcategory of embeddings*, denoted $\mathbf{C}_e$, is the full-on-objects subcategory of **C** whose morphisms are exactly the embeddings of **C**.

Theorem (Smyth and Plotkin'82)

Let $\mathbf{A}, \mathbf{B}$ *and* **C** *be* **CPO**-*categories where* **A** *and* **B** *have* ω-*colimits over embeddings. If* $T : \mathbf{A}^{\mathrm{op}} \times \mathbf{B} \to \mathbf{C}$ *is a* **CPO**-*functor, then the covariant functor* $T_e : \mathbf{A}_e \times \mathbf{B}_e \to \mathbf{C}_e$

$$T_e(A, B) = T(A, B) \qquad \text{and} \qquad T_e(e_1, e_2) = T((e_1^{\bullet})^{\mathrm{op}}, e_2)$$

is an ω-*functor.*

Remark

Even though this has been known for a while, I found no papers which use this for denotational semantics as a basis for type interpretation.

Models of Intuitionistic Linear Logic

A model of ILL[4] is given by the following data:

- A cartesian closed category **C** with finite coproducts.
- A symmetric monoidal closed category **L** with finite coproducts.
- A symmetric monoidal adjunction:

$$
\begin{tikzcd}
\mathbf{C} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathbf{L} \arrow[ll, bend left, "G"]
\end{tikzcd}
$$

[4]Nick Benton. *A mixed linear and non-linear logic: Proofs, terms and models*. CSL'94

Models of LNL-FPC

Definition

A **CPO**-*LNL model* is given by the following data:

1. A **CPO**-symmetric monoidal closed category $(\mathbf{L}, \otimes, \multimap, I)$, such that:
   - 1a. $\mathbf{L}$ has an initial object 0, such that the initial morphisms $e : 0 \to A$ are embeddings;
   - 1b. $\mathbf{L}$ has ω-colimits over embeddings;
   - 1c. $\mathbf{L}$ has finite **CPO**-coproducts, where $(- + -) : \mathbf{L} \times \mathbf{L} \to \mathbf{L}$ is the coproduct functor.
2. A **CPO**-symmetric monoidal adjunction $\begin{tikzcd} \mathbf{CPO} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{L} \arrow[l, bend left, "G"] \end{tikzcd}$.

Theorem

In every **CPO**-*LNL model:*

1. *The initial object* 0 *is a zero object and each zero morphism* $\perp_{A,B}$ *is least in* $\mathbf{L}(A, B)$*;*
2. $\mathbf{L}$ *is* **CPO**-*algebraically compact.*

A new technique for solving recursive domain equations

Problem

How to interpret the non-linear recursive types within **CPO***.*

Definition

Let $T : \mathbf{A} \to \mathbf{B}$ be a **CPO**-functor between **CPO**-categories $\mathbf{A}$ and $\mathbf{B}$. A morphism f in $\mathbf{A}$ is called a *pre-embedding with respect to* T if Tf is an embedding in $\mathbf{B}$.

Definition

Let $\mathbf{CPO}_{pe}$ be the full-on-objects subcategory of **CPO** of all cpo's with pre-embeddings with respect to the functor $F : \mathbf{CPO} \to L$.

Example

Every embedding in **CPO** is a pre-embedding, but not vice versa. The empty map $\iota : \varnothing \to X$ is a pre-embedding (w.r.t to F in our model), but not an embedding.

A new technique for solving recursive domain equations (contd.)

Theorem

In every $\mathbf{CPO}$*-LNL model:*

(1) $\mathbf{L}_e$ *is an* ω*-category, and the subcategory inclusion* $\mathbf{L}_e \hookrightarrow \mathbf{L}$ *is a strict* ω*-functor which also reflects* ω*-colimits.*

(2) $\mathbf{CPO}_{pe}$ *is an* ω*-category and the subcategory inclusion* $\mathbf{CPO}_{pe} \hookrightarrow \mathbf{CPO}$ *is a strict* ω*-functor which also reflects* ω*-colimits.*

(3) *The subcategory inclusion* $\mathbf{CPO}_e \hookrightarrow \mathbf{CPO}_{pe}$ *preserves and reflects* ω*-colimits (*$\mathbf{CPO}_e$ *has no initial object).*

Remark

We have a few more theorems showing all relevant functors (even mixed-variance ones) from the categorical data become ω*-functors when considered as covariant functors on* $\mathbf{CPO}_{pe}$ *and* $\mathbf{L}_e$*. So, we interpret our types in* $\mathbf{CPO}_{pe}$ *and* $\mathbf{L}_e$*.*

Concrete Models

Theorem

The adjunction $\mathbf{CPO} \underset{U}{\overset{(-)_\perp}{\underset{\perp}{\rightleftarrows}}} \mathbf{CPO}_{\perp!}$ *, where the left adjoint is given by (domain-theoretic) lifting and the right adjoint* U *is the forgetful functor, is a* **CPO**-*LNL model.*

Concrete Models (Presheaves)

For $\mathbf{M}$ a small symmetric monoidal category, let $\mathbf{M}_*$ indicate the free $\mathbf{CPO}_{\perp!}$-enrichment of $\mathbf{M}$ and let $\widehat{\mathbf{M}}$ be the category of $\mathbf{CPO}_{\perp!}$-presheafs and $\mathbf{CPO}_{\perp!}$-natural transformations from $\mathbf{M}_*$ to $\mathbf{CPO}_{\perp!}$.

Theorem

Composing the two adjunctions
$$\begin{tikzcd} \mathbf{CPO} \arrow[r, shift left, "(-)_\perp"] \arrow[r, phantom, "\perp"] & \mathbf{CPO}_{\perp!} \arrow[l, shift left, "U"] \arrow[r, shift left, "- \circledcirc I"] \arrow[r, phantom, "\perp"] & \widehat{\mathbf{M}} \arrow[l, shift left, "\widehat{\mathbf{M}}(I, -)"] \end{tikzcd}$$
yields a $\mathbf{CPO}$-*LNL model.*

Concrete Models (Presheaves contd.)

Example

If the category **M** is:

- the PROP with morphisms $n \times n$ complex matrices, then we get a model for quantum programming.
- the free category of ZX-calculus diagrams, then we get a model for a ZX-diagram description language.
- the free category of string diagrams generated by some signature, then we get a string diagram description language.
- the category of Petri Nets with Boundary[5] then we get a model for a petri net description language.

[5] Owen Stephens (2015): Compositional specification and reachability checking of net systems.

Concrete Models (Kegelspitzen)

Conjecture

We suspect a model based on Kegelspitzen[6] also satisfies our requirements and is a CPO-LNL model.

[6] Keimel and Plotkin 2016, Mixed powerdomains for probability and nondeterminism.

Denotational Semantics (Types)

Main idea:

- Provide a standard interpretation for all types $[\![\Theta \vdash A]\!] : \ \mathbf{L_e}^{|\Theta|} \to \mathbf{L}_e$.
- A closed type is interpreted as $[\![A]\!] \in \mathrm{Ob}(\mathbf{L}_e) = \mathrm{Ob}(\mathbf{L})$.
- Provide a non-linear interpretation for non-linear types $(\!|\Theta \vdash P|\!) : \mathbf{CPO}_{pe}^{|\Theta|} \to \mathbf{CPO}_{pe}$.
- A closed non-linear type admits an interpretation as $(\!|P|\!) \in \mathrm{Ob}(\mathbf{CPO}_{pe}) = \mathrm{Ob}(\mathbf{CPO})$.
- Show that there exists a *coherent* family of isomorphisms $[\![P]\!] \cong F(\!|P|\!)$, which are then used to carry the comonoid structure from **CPO** to **L**.

Denotational Semantics (Types)

$$
\begin{aligned}
[\![\Theta \vdash A]\!] &: \ \mathbf{L_e}^{|\Theta|} \to \mathbf{L}_e \\
[\![\Theta \vdash \Theta_i]\!] &:= \Pi_i \\
[\![\Theta \vdash !A]\!] &:= !_e \circ [\![\Theta \vdash A]\!] \\
[\![\Theta \vdash A + B]\!] &:= +_e \circ \langle [\![\Theta \vdash A]\!], [\![\Theta \vdash B]\!] \rangle \\
[\![\Theta \vdash A \otimes B]\!] &:= \otimes_e \circ \langle [\![\Theta \vdash A]\!], [\![\Theta \vdash B]\!] \rangle \\
[\![\Theta \vdash A \multimap B]\!] &:= \multimap_e \circ \langle [\![\Theta \vdash A]\!], [\![\Theta \vdash B]\!] \rangle \\
[\![\Theta \vdash \mu X.A]\!] &:= [\![\Theta, X \vdash A]\!]^\dagger
\end{aligned}
$$

$$
\begin{aligned}
(\!|\Theta \vdash P|\!) &: \mathbf{CPO}_{pe}^{|\Theta|} \to \mathbf{CPO}_{pe} \\
(\!|\Theta \vdash \Theta_i|\!) &:= \Pi_i \\
(\!|\Theta \vdash !A|\!) &:= G_{pe} \circ [\![\Theta \vdash A]\!] \circ F_{pe}^{\times|\Theta|} \\
(\!|\Theta \vdash P + Q|\!) &:= \amalg_{pe} \circ \langle (\!|\Theta \vdash P|\!), (\!|\Theta \vdash Q|\!) \rangle \\
(\!|\Theta \vdash P \otimes Q|\!) &:= \times_{pe} \circ \langle (\!|\Theta \vdash P|\!), (\!|\Theta \vdash Q|\!) \rangle \\
(\!|\Theta \vdash \mu X.P|\!) &:= (\!|\Theta, X \vdash P|\!)^\dagger
\end{aligned}
$$

Coherence of the interpretations

Theorem

For any non-linear type $\Theta \vdash P$, there exists a natural isomorphism

$$\alpha^{\Theta \vdash P} : [\![\Theta \vdash P]\!] \circ F_{pe}^{\times|\Theta|} \Rightarrow F_{pe} \circ (\!|\Theta \vdash P|\!) : \mathbf{CPO}_{pe}^{|\Theta|} \to \mathbf{L}_e$$

defined by induction on $\Theta \vdash P$ which satisfies some important coherence conditions.

Corollary

For any closed non-linear type P, there exists an isomorphism

$$\alpha^{P} : [\![P]\!] \cong F(\!|P|\!)$$

which satisfies some important coherence conditions.

Coherence for folding/unfolding

Theorem

Let $\Theta \vdash \mu X.P$ be a non-linear type. Then the diagram of natural isomorphisms

$$\begin{tikzcd}
\llbracket \Theta \vdash P[\mu X.P/X] \rrbracket \circ F_{pe}^{\times|\Theta|} \arrow[r, Rightarrow, "\alpha"] \arrow[d, Rightarrow, "\mathrm{fold} F_{pe}^{\times|\Theta|}"'] & F_{pe} \circ (\!| \Theta \vdash P[\mu X.P/X] |\!) \arrow[d, Rightarrow, "F_{pe} \mathit{fold}"] \\
\llbracket \Theta \vdash \mu X.P \rrbracket \circ F_{pe}^{\times|\Theta|} \arrow[r, Rightarrow, "\alpha"'] & F_{pe} \circ (\!| \Theta \vdash \mu X.P |\!)
\end{tikzcd}$$

commutes (note: one has to first formulate 3 substitution lemmas and define 2 fold/unfold maps).

Copying and discarding

Definition

We define morphisms, called discarding ($\diamond$), copying ($\triangle$) and promotion ($\Box$):

$$\diamond^{\Psi} := \llbracket\Psi\rrbracket \xrightarrow{\alpha} F(\!|\Psi|\!) \xrightarrow{F1} F1 \xrightarrow{u^{-1}} I;$$

$$\triangle^{\Psi} := \llbracket\Psi\rrbracket \xrightarrow{\alpha} F(\!|\Psi|\!) \xrightarrow{F\langle \mathrm{id},\mathrm{id}\rangle} F\left((\!|\Psi|\!) \times (\!|\Psi|\!)\right) \xrightarrow{m^{-1}} F(\!|\Psi|\!) \otimes F(\!|\Psi|\!) \xrightarrow{\alpha^{-1}\otimes\alpha^{-1}} \llbracket\Psi\rrbracket \otimes \llbracket\Psi\rrbracket;$$

$$\Box^{\Psi} := \llbracket\Psi\rrbracket \xrightarrow{\alpha} F(\!|\Psi|\!) \xrightarrow{F\eta} !F(\!|\Psi|\!) \xrightarrow{!\alpha^{-1}} !\llbracket\Psi\rrbracket,$$

where Ψ is a closed non-linear type or non-linear term context.

Proposition

The triple $\left(\llbracket\Psi\rrbracket, \triangle^{\Psi}, \diamond^{\Psi}\right)$ *forms a cocommutative comonoid in* $\mathbf{L}$.

Denotational Semantics (Terms)

- A term $\Gamma \vdash m : A$ is interpreted as a morphism $[\![\Gamma \vdash m : A]\!] : [\![\Gamma]\!] \to [\![A]\!]$ in $\mathbf{L}$ in the standard way.
- The interpretation of a non-linear value $[\![\Phi \vdash v : P]\!]$ commutes with the substructural operations of ILL (shown by providing a non-linear interpretation $(\!|\Phi \vdash v : P|\!)$ within **CPO**).
- Soundness: If $m \Downarrow v$, then $[\![m]\!] = [\![v]\!]$.
- Adequacy: For models that satisfy some additional axioms, the following is true: for any $\cdot \vdash m : P$ with P non-linear, then $m \Downarrow$ iff $[\![m]\!] \neq \perp$.

Conclusion

- Introduced LNL-FPC: the linear/non-linear fixpoint calculus;
- Implicit weakening and contraction rules (copying and deletion of non-linear variables);
- New results about parameterised initial algebras;
- New technique for solving recursive domain equations in **CPO**;
- Detailed semantic treatment of mixed linear/non-linear recursive types;
- Sound and adequate models;
- How to axiomatise **CPO** away?
- More concrete models?

Thank you for your attention!