Recursion and Sequentiality in Categories of Sheaves

Cristina Matache (joint work with Sean Moss and Sam Staton)

University of Oxford

Fully abstract models of programming languages

Model

- Cartesian closed category
- Partiality monad, L
- Interpretation: Type $\longleftrightarrow$ Object
  Program $\longleftrightarrow$ Partial morphism

A model is **fully abstract** if:

Contextual equivalence = Equality in the model

$$t_1 \cong t_2 \iff [\![t_1]\!] = [\![t_2]\!]$$

The $\Longrightarrow$ is hard to get.

$\mathrm{PCF_v}$: A call-by-value language

$$\begin{aligned}
\text{Types:}\quad & \tau ::= 0 \mid 1 \mid \mathsf{nat} \mid \tau + \tau \mid \tau \times \tau \mid \tau \to \tau \\
\text{Values:}\quad & v, w ::= \ldots \mid \lambda x.\, t \mid \mathsf{rec}\, f x.\, t \\
\text{Computations:}\quad & t ::= \ldots \mid v\, w \mid \mathsf{let}\, x = t \,\mathsf{in}\, t'
\end{aligned}$$

Typing judgements: $\Gamma \vdash^{\mathsf{v}} v : \tau$ and $\Gamma \vdash^{\mathsf{c}} t : \tau$.

An interpretation looks like:

$$\begin{aligned}
[\![\mathsf{nat}]\!] = \textstyle\sum_0^\infty 1 = 1 + 1 + \ldots \qquad & [\![\tau \to \tau']\!] = [\![\tau]\!] \Rightarrow L[\![\tau']\!] \\
[\![\Gamma \vdash^{\mathsf{v}} v : \tau]\!] : [\![\Gamma]\!] \to [\![\tau]\!] \qquad & [\![\Gamma \vdash^{\mathsf{c}} t : \tau]\!] : [\![\Gamma]\!] \to L[\![\tau]\!]
\end{aligned}$$

Related Work

The ωcpo model of PCF_v:

Types $\longleftrightarrow$ posets with sups of ω-chains.

Terms $\longleftrightarrow$ continuous functions.

! **Not fully abstract.** E.g. parallel-or not definable.

Need to capture sequentiality

O'Hearn and Riecke's idea [OHR'95, Riecke&Sandholm'02]

Use logical relations to cut down to sequential functions.

[Plotkin'80], [Jung & Tiuryn'93]: logical relations for λ-definability.

[Sieber'92]: definability for PCF up to order 2.

Related Work

The ωcpo model of PCF_v:

Types $\longleftrightarrow$ posets with sups of ω-chains.

Terms $\longleftrightarrow$ continuous functions.

! **Not fully abstract.** E.g. parallel-or not definable.

Need to capture sequentiality

O'Hearn and Riecke's idea [OHR'95, Riecke&Sandholm'02]

Use logical relations to cut down to sequential functions.

What we did [MMS'21]

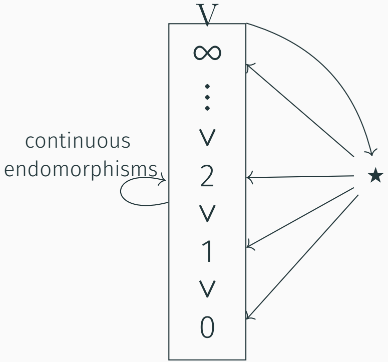
Describe the OHR model as a sheaf category.

Outline

Concrete presheaves on the vertical natural numbers

$\mathrm{V} = \{0 < 1 < 2 < \ldots < \infty\}$ = poset of vertical natural numbers

$\mathbb{V}$ = two-object category:



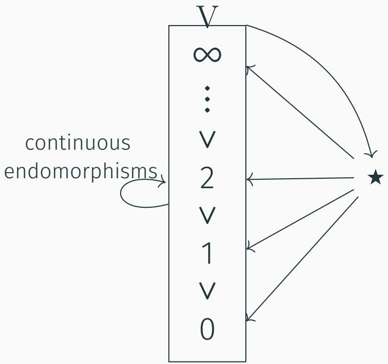
vSet = $[\mathbb{V}^{op}, Set]$= presheaves on $\mathbb{V}$

Concrete presheaf on $\mathbb{V}$

- a set $X(\star)$
- a set of functions $X(\mathrm{V}) \subseteq [\mathrm{V} \to X(\star)]$

$X(\mathrm{V})$ is a **relation** with arity V on $X(\star)$.

Concrete presheaves on the vertical natural numbers



vSet = $[\mathbb{V}^{op}, Set]$

Concrete presheaf on $\mathbb{V}$

- a set $X(\star)$
- a set of functions $X(\mathbb{V}) \subseteq [\mathbb{V} \to X(\star)]$

$X(\mathbb{V})$ is a **relation** with arity $\mathbb{V}$ on $X(\star)$.

A map between concrete presheaves X and Y is:

- a function $f : X(\star) \to Y(\star)$
- acting by postcomposition: $g \in X(\mathbb{V}) \mapsto f \circ g \in Y(\mathbb{V})$ i.e. f **preserves the relation**.

Exponentials in vSet

If X and Y are **concrete presheaves**, the exponential is also a concrete presheaf:

$(X \Rightarrow Y)(\star) = \{f : X(\star) \to Y(\star) \mid f \text{ preserves the relation}\}$

$(X \Rightarrow Y)(\mathrm{V}) \subseteq [\mathrm{V} \to (X \Rightarrow Y)(\star)]$ such that (among other conditions)

if $(f_0, f_1, \ldots) \in (X \Rightarrow Y)(\mathrm{V})$
then $(x_0, x_1, \ldots) \in X(\mathrm{V})$ implies $(f_0(x_0), f_1(x_1), \ldots) \in Y(\mathrm{V})$.

So $(X \Rightarrow Y)(\mathrm{V})$ is a **“logical” relation**.

Partiality monad L on vSet

For a **concrete presheaf** X:

$(LX)(\star) = X(\star) + \{\bot\}$

$(LX)(\mathrm{V}) = \{\bot\} + \sum_{n\in\mathbb{N}}(X(\mathrm{V}))_n$

$(X(\mathrm{V}))_n$ contains each chain from $X(\mathrm{V})$ with n $\bot$-elements added at the beginning.

Modelling $\mathsf{PCF_v}$ in vSet

Claim

We can model $\mathsf{PCF_v}$ using the concrete presheaves in vSet, starting from:
$[\![\mathrm{nat}]\!](\star) = \mathbb{N}$
$[\![\mathrm{nat}]\!](V) = \{\text{constant functions } V \to \mathbb{N}\}$.

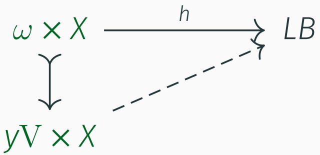
! The vSet model is actually the ωcpo model.

Modelling fixed points in vSet

```
type vertical = Succ of (unit -> vertical);;

let rec top : vertical = Succ (fun () -> top);;

let lub ((fs, ax) : (vertical * 'a -> 'b) * 'a) : 'b =
  fs (top, ax);;
```

```
let rec approx : (vertical * (('a -> 'b) * 'a -> 'b) * 'a) -> 'b

let tarski : ((('a -> 'b) * 'a -> 'b) * 'a) -> 'b
```

Similarly we can define a fixed point of $f : (A \Rightarrow LB) \times A \to LB$ in vSet if LB is **orthogonal** to $\omega \times X \to y\mathrm{V} \times X$ for any X in vSet:



ω = greatest subobject of $y\mathrm{V}$ without ∞

Outline

Building a fully abstract *syntactic* model of $\mathrm{PCF_v}$

Semidecidable subset of a type τ = represented by a program $s : \tau \to 1$.

Category *Syn*:

- Objects: (τ, s) type + semidecidable subset
- Morphisms: $f : (\tau, s) \to (\tau', s')$ is a(n equivalence class of) program(s) $x : \tau \vdash f : \tau'$ with domain s and image in s'.

Building a fully abstract *syntactic* model of $\mathrm{PCF_v}$

$[Syn^{op}, Set]$ almost a model with **full definability** $\Longrightarrow$ **full abstraction**.

Problems:

1. y(nat) is not $\sum_0^\infty 1$ in presheaves.
2. Recursion.
3. We'd like a non-syntactic model.

Solving 1 and 2: nat as $\sum_0^\infty 1$, and recursion

Use a **sheaf condition** on *Syn* to make $y(\text{nat})$ a coproduct.

! There are uncountably many maps $\sum_0^\infty 1 \to \sum_0^\infty 1$.
We can't get full definability.

For each n, consider Syn_n such that natural numbers $> n$ trigger divergence [Milner'77].

Combine the **truncated** sites Syn_n and impose a sheaf condition on them.

Solving 2, recursion: add $\mathbb{V}$ as one of the sites.

Something like $\text{Sh}(\mathbb{V} + \bigvee_n Syn_n)$
has full definability for truncated types.

Solving 3: Non-syntactic model

Instead of Syn_n use a bigger class of sites.

Given a finite set w:

A system of partitions S^w [Streicher'06, Marz'00]

Contains **partial partitions** (=partial equivalence relations) of w s.t.:

1. $\{w\}, \varnothing \in S^w$
2. P, $Q \in S^w$ and $U \in P$ imply that: $(P \setminus \{U\}) \cup (\{U \cap U' \mid U' \in Q\} \setminus \{\varnothing\}) \in S^w$.
3. $U, U' \in P \in S^w$ implies that $(P \setminus \{U, U'\}) \cup \{U \cup U'\} \in S^w$.

Systems of partitions

w =finite set $S^w \subseteq$ {partial partitions of w} +axioms

(w, S^w): w is a finite type
$P \in S^w$ is (roughly) a computable function $w \rightharpoonup \mathbb{N}$

The axioms of S^w imply that the system of functions:

- includes all constant functions
- is closed under postcomposition with any $f : \mathbb{N} \rightharpoonup \mathbb{N}$
- is closed under sequencing of functions from S^w.

! For $P \in S^w$, think of $\bigcup P$ as the **semidecidable subset** s from $(\tau_n, s : \tau_n \to 1)$ from Syn_n.

SSP: A category of systems of partitions

w =finite set $S^w \subseteq$ {partial partitions of w} +axioms
$P \in S^w$, $\bigcup P$ =semidecidable subset of w

The systems of partitions form a category SSP:

- Objects: (w, S^w)
- Morphisms: $f : (v, S^v) \to (w, S^w)$ is a function $f : v \to w$ s.t. if $P \in S^w$ then $f^{-1}(P) \in S^v$.

Partiality monad $L_{\mathrm{SSP}}(w, S^w) = (w \sqcup \{\bot\}, \ldots)$.

! A map in Syn_n is a partial function $(\tau_n, s) \to (\tau_n', s')$ with domain s and image in s'.

Defining sites via systems of partitions

w =finite set $\quad S^w \subseteq$ {partial partitions of w} +axioms
$P \in S^w$, $\bigcup P$ =semidecidable subset of w
$\mathrm{SSP}_\perp$ has Kleisli maps $(v, S^v) \to L_{\mathrm{SSP}}(w, S^w)$

For a faithful functor $F : \mathcal{C} \to \mathrm{SSP}_\perp$ define a category $\mathcal{I}_{\mathcal{C},F}$ similar to Syn_n:

- Objects: (c, U), $c \in \mathcal{C}$ and $U = \bigcup P$ for some $P \in S^{F(c)}$, (and a terminal object).
- Morphisms: $f : (c, U) \to (d, W)$ is a function $f : U \to W$
  - either constant
  - or s.t. there is $F(\phi) : F(c) \to L_{\mathrm{SSP}}(F(d))$ with domain U and image in W.

Each $\mathcal{I}_{\mathcal{C},F}$ is a **guess** at Syn_n.

First attempt at a model using guesses

Candidate model: $[(\mathbb{V} + \bigvee_{F:\mathcal{C}\to\mathrm{SSP}_\perp} \mathcal{I}_{\mathcal{C},F})^{op}, Set]$

If (c, U) is a type, $S^{F(c)}$ encodes the maps $U \to \mathrm{nat}$,

nat needs to be interpreted as the **concrete presheaf**:

$$\llbracket \mathrm{nat} \rrbracket(\star) = \mathbb{N}$$
$$\llbracket \mathrm{nat} \rrbracket(c, U) = \{g : U \to \mathbb{N} \mid \{g^{-1}(i) \mid i \in \mathbb{N}\} \in S^{F(c)}\}$$

But this is not the coproduct $\sum_0^\infty 1$:

$$(\textstyle\sum_0^\infty 1)(\star) = \mathbb{N} \qquad (\textstyle\sum_0^\infty 1)(c, U) = \{f : U \to \mathbb{N} \mid f \text{ constant}\}$$

From presheaves to sheaves

Final model: $\mathcal{G} = \mathrm{Sh}(\mathbb{V} + \bigvee_{F:\mathcal{C}\to\mathrm{SSP}_\perp} \mathcal{I}_{\mathcal{C},F})$

In $\mathcal{G}$ the same $[\![\mathrm{nat}]\!]$ becomes a coproduct.

Sheaf condition:

- (c, U) covered by $\{(c, U_i) \to (c, U)\}_{1\leq i\leq n}$ where $P = \{U_1, \ldots, U_n\} \in S^{F(c)}$ and $\bigcup U_i = U$.
- A concrete presheaf X is a sheaf if given a tuple of functions $\big(f_i : U_i \to X(\star) \in X(c, U_i)\big)_{U_i\in P}$ then $(f_1 + f_2 + \ldots + f_n) : U \to X(\star) \in X(c, U)$.
- Ensures sum types are interpreted as coproducts.

$\mathcal{G}$ is a model of PCF_v

Partiality monad on $\mathcal{G} = \mathrm{Sh}(\mathbb{V} + \bigvee_{F:\mathcal{C}\to\mathrm{SSP}_\perp} \mathcal{I}_{\mathcal{C},F})$:

$$(L_{\mathcal{G}}X)(\star) = X(\star) + \{\perp\}$$
$$(L_{\mathcal{G}}X)(c, U) = \sum_{W\subseteq U} X(c, W) \text{ s.t. exists } P \in S^{F(c)}, \bigcup P = W.$$

Theorem

$\mathcal{G}$, with $L_{\mathcal{G}}$, gives a fully abstract model of PCF_v such that:

1. nat is interpreted as $\sum_0^\infty 1$
2. we interpret recursion
3. the model is non-syntactic.

The connection between $\mathcal{G}$ and logical relations

All types τ are interpreted as **concrete sheaves** $[\![\tau]\!]$.

The interpretation of $\mathrm{PCF_v}$ can be thought of as a:

Kripke logical relation of varying arity

- $[\![\tau]\!](c, U)$ is a **relation** with arity U (like in vSet).
- **logical**: at function types $\tau_1 \to \tau_2$, a tuple of related functions maps related arguments to related results.
- **Kripke**: the relation $[\![\tau]\!](c, U)$ is compatible with $[\![\tau]\!](d, W)$ according to the maps $(d, W) \to (c, U)$.
- **varying arity**: $[\![\tau]\!](d, W)$ has arity $W \neq U$.

Outline

Summary

Fully abstract model of PCF_v:

- Recursion: presheaves on $\mathbb{V}$
- Definability/Sequentiality: guess the truncated types
- Take sheaves on these guesses to model nat and sum types as coproducts.

- Each partiality monad comes from a dominance, like in synthetic domain theory.

Future work

- Recursive types [Riecke & Sandholm'02]
- Other computational effects
- Non-well-pointed models [Levy'07, Amb breaks well-pointedness...]