Concrete categories and higher-order recursion

With applications including probability, differentiability, and full abstraction

Cristina Matache[†‡], Sean Moss[†] and Sam Staton[†]

LICS 2022

[†]University of Oxford and [‡]University of Edinburgh

Modelling higher-order programs with recursion

Model

- Cartesian closed category (CCC) — higher-order functions
- Partiality monad, L — recursion
- Interpretation:
  Type $\longleftrightarrow$ Object
  Program $\longleftrightarrow$ Partial morphism with admissible domain

Examples:

(1) Probabilistic programming [Heunen et al.'17, Vákár et al.'19]

(2) Differentiable programming [Huot et al.'20, Vákár'20]

(3) Full abstraction for a sequential language

[O'Hearn & Riecke'95], [Matache, Moss, Staton, FSCD'21]

Goal of this work

The examples all model higher-order recursion using the same recipe

(1) Probabilistic programming
(2) Differentiable programming
(3) Full abstraction for a sequential language

Main Theorem (Adequacy)

We build an **adequate** model of **higher-order recursion** as a category of **concrete sheaves**.

Each example is a special case + some domain specific work.

Concreteness: types = sets with structure, terms = structure preserving functions.

Categories of concrete sheaves $\mathsf{ConcSh}(\mathbb{C}, J)$

[*Concrete quasitopoi*, Dubuc'77]
[*Convenient categories of smooth spaces*, Baez & Hoffnung'11]

$(\mathbb{C}, J)$ = **site** of the sheaf category

$\mathbb{C} =$ small (well-pointed) category; models first-order computation

- concrete presheaves on $\mathbb{C}$ model higher-order computation
- restricting to concrete **sheaves** for a coverage J on $\mathbb{C}$ changes the colimits, e.g. $[\![\mathsf{nat}]\!]$ is the coproduct $\sum_{\mathbb{N}} 1$

Concrete sheaf X = set $|X|$ + sets of functions into $|X|$ + some conditions

(1) Probability: sets of random elements $\mathbb{R} \to |X|$
(2) Differentiability: sets of smooth plots $\mathbb{R}^n \to |X|$
(3) Sequentiality: logical relations on $|X|$.

Partiality monad L on ConcSh$(\mathbb{C}, J)$

Theorem

Starting with a **class of admissible monos** $\mathcal{M}$ in the site $(\mathbb{C}, J)$ we can construct a lifting monad L on ConcSh$(\mathbb{C}, J)$.

Proof sketch:

- From $\mathcal{M}$ we obtain a dominance Δ in Sh$(\mathbb{C}, J)$ (in the sense of synthetic domain theory e.g. [Rosolini'86])
- Δ classifies the admissible domains of partial maps
- From the dominance Δ we construct L [Mulry'94, Fiore&Plotkin'97].

Main theorem

$\mathsf{ConcSh}(\mathbb{C}, J)$ with L will not in general admit a fixed point theorem.

Consider the partial order $\mathrm{V} = [0 \leq 1 \leq \ldots \leq \infty]$ and combine with $\mathbb{C}$

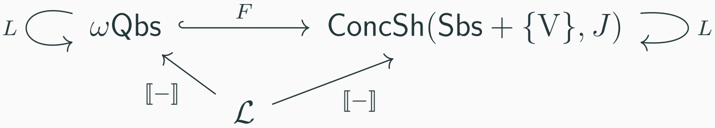
X in $\mathsf{ConcSh}(\mathbb{C} + \{\mathrm{V}\}, J)$ has a set of completed chains $X(\mathrm{V}) \subseteq [\mathrm{V} \to |X|]$

$\implies$ FP theorem in $\mathsf{ConcSh}(\mathbb{C} + \{\mathrm{V}\}, J)$ see also [Fiore & Rosolini'97, '01], [Fiore & Plotkin'97]

Main Theorem (Adequacy)

$\mathsf{ConcSh}(\mathbb{C} + \{\mathrm{V}\}, J)$ with L is an adequate model for call by value PCF.

Example: the ωQbs model of probabilistic computation



Summary

We built an **adequate concrete sheaf** model of **higher-order recursion**.

Examples that are an instance of this construction:

(1) Probabilistic programming
(2) Differentiable programming
(3) Full abstraction for a sequential language

Expect more examples in the future
e.g. piecewise differentiability [Lew et al.'21]