Solving Guarded Domain Equations in Presheaves Over Ordinals and Mechanizing It

Sergei Stepanenko [1]   Amin Timany [1]

[1]Aarhus University

June 13, 2025

Context

Semantics of higher-order languages

$$\begin{array}{lll} \text{expressions} & \mathrm{Expr} \ni e, e_i & ::= \; e_1\ e_2 \mid v \mid \dots \\ \text{values} & \mathrm{Val} \ni v & ::= \; x \mid \mathrm{fix}\ f\ x.\ e \mid \dots \end{array}$$

$$(\mathrm{fix}\ f\ x.\ e)\ v \mapsto e[v/x][\mathrm{fix}\ f\ x.\ e/f]$$

Problem (from denotational semantics field)

Can we represent expressions as mathematical objects, such that reductions become equalities?

Semantics of higher-order languages

$$
\begin{array}{lll}
\text{expressions} & \mathrm{Expr} \ni e, e_i & ::= \; e_1\ e_2 \mid v \mid \ldots \\
\text{values} & \mathrm{Val} \ni v & ::= \; x \mid \mathrm{fix}\ f\ x.\ e \mid \ldots
\end{array}
$$

$$\mathbb{D} \cong T(\mathbb{D} \to \mathbb{D}) + \ldots$$

Quintessential example of recursive domain equations.

$$F(X, Y) \triangleq T(X \to Y) + \ldots$$

Semantics of higher-order languages

$$\mathbb{D} \cong T(\mathbb{D} \to \mathbb{D}) + \dots$$

Negative occurrence, need to guard ($\blacktriangleright$) negative occurrences.

- Intuitively, we have implicit 'fuel'.
- If we have some fuel left, $\blacktriangleright X$ consumes a unit of fuel, and becomes X.
- If there is no fuel, $\blacktriangleright X$ is a trivial (singleton) set.

Semantics of higher-order languages

$$\mathbb{D} \cong \blacktriangleright(\mathbb{D} \to \mathbb{D}) + \blacktriangleright\mathbb{D} + \dots$$

- Intuitively, we have implicit 'fuel'.
- If we have some fuel left, $\blacktriangleright X$ consumes a unit of fuel, and becomes X.
- If there is no fuel, $\blacktriangleright X$ is a trivial (singleton) set.

$$\begin{aligned}
\mathbb{D}_0 &= 1 + 1 + \dots \\
\mathbb{D}_1 &= (\mathbb{D}_0 \to \mathbb{D}_0) + \mathbb{D}_0 + \dots \\
\dots
\end{aligned}$$

Semantics of higher-order languages

$$\mathbb{D} \cong \blacktriangleright(\mathbb{D} \to \mathbb{D}) + \blacktriangleright\mathbb{D} + \dots$$

- $\blacktriangleright$ is an applicative pointed endofunctor ($\mathsf{next} \colon \mathbf{Id} \to \blacktriangleright$).
- There is a fixpoint combinator $\mu \colon (\blacktriangleright\mathbb{D} \to \mathbb{D}) \to \mathbb{D}$, such that $\mu f \equiv f(\mathsf{next}(\mu f))$.

$$\begin{aligned}
[\![-]\!]_{-} &: \mathrm{Expr} \to (\mathrm{Var} \to \mathbb{D}) \to \mathbb{D} \\
[\![x]\!]_{\gamma} &\triangleq \gamma(x) \\
[\![\mathrm{fix}\ f\ x.\ e]\!]_{\gamma} &\triangleq \mu(\lambda F \colon \blacktriangleright\mathbb{D}.\mathsf{next}(\lambda X.[\![e]\!]_{\gamma, f \mapsto F, x \mapsto X})) \\
\dots &\triangleq \dots
\end{aligned}$$

An interface to solve recursive domain equations, and to find fixpoints.

Complete Ordered Families of Equivalences (COFEs)

Sets with step-indexed down-closed equivalences ($\mathcal{OFE}$), and a way to compute 'limits' wrt equivalences ($\mathcal{COFE}$).

Solver for certain subset of functors $\mathcal{COFE}^{\mathsf{op}} \times \mathcal{COFE} \to \mathcal{COFE}$.

- Implementation of one particular model in a host proof assistant (Rocq).
- Shown to be practical (Iris, Iris-based frameworks).
- Allows to extract proofs into Rocq propositions.

Gap

Transfinite Indexing

In some cases we need to consider indexing beyond ω (e.g., ω_1 for applicative bisimulation for $\mathbb{D}$).

If $\vDash \exists x : \mathbb{N}.\, \Phi\ x$, then $\vDash \Phi\ x$ for some $x : \mathbb{N}$, if the indexing is over ω_1.

Transfinite COFEs

Sets with step-indexed (indexing goes beyond ω) down-closed equivalences, and a way to compute 'limits'.

Solver for certain subset of functors $\mathcal{OFE}^{\mathsf{op}} \times \mathcal{OFE} \to \mathcal{COFE}$.

Some recursive domain equations can be solved, but not the one for $\mathbb{D}$.

$$\mathbb{D} \cong \blacktriangleright(\mathbb{D} \to \mathbb{D}) + \blacktriangleright\mathbb{D} + \dots$$

$$F(X, Y) \triangleq \blacktriangleright(X \to Y) + \blacktriangleright Y + \dots$$

Sheaves over ordinals

- Use sheaves over ordinals.[a]
- Use presheaves over ordinals.
- Presheaves are easier to work with than sheaves.

[a] First steps in synthetic guarded domain theory: step-indexing in the topos of trees, L. Birkedal, R. E. Møgelberg, J. Schwinghammer, K. Støvring

Our solution

- Guarded domain theory within existing Rocq ecosystem.
- Presheaves instead of sheaves.

Details

Intuition (sheaves)

- Presheaves over ordinals are ordinal-indexed families of sets.

$$F(0) \xleftarrow[F(0\leq 1)]{} F(1) \longleftarrow \dots \longleftarrow F(\omega) \longleftarrow \dots \longleftarrow F(\omega+\omega) \longleftarrow \dots$$

- Sheaves over ordinals are presheaves, where sets at limit ordinals are determined by elements below, the lowest set is trivial.

$$\begin{tikzcd}
\lim\limits_{i \prec \omega} F(i) \cong F(\omega) \arrow[d] \arrow[dr] \arrow[drr] & & & \\
1 \cong F(0) & F(1) \arrow[l] & F(2) \arrow[l] & \dots \arrow[l]
\end{tikzcd}$$

$$F(0) \cong 1$$
$$F(\kappa) \cong \lim_{i \prec \kappa} F(i)$$

Later modality

- $\blacktriangleright$ makes presheaves trivial at 0, shifts the rest, and ...
- transforms presheaves into sheaves!
- There is a left adjoint to $\blacktriangleright$, and it is an equivalence.

$$
\begin{tikzcd}
\lim_{d \prec \omega} F(d) = \blacktriangleright F(\omega) \arrow[d] \arrow[dr] \arrow[drr] & & & \\
1 = \blacktriangleright F(0) & F(0) = \blacktriangleright F(1) \arrow[l] & F(1) = \blacktriangleright F(2) \arrow[l] & \dots \arrow[l]
\end{tikzcd}
$$

$$\blacktriangleright F(c) \triangleq \lim_{d \prec c} F(d) \qquad \blacktriangleright F(c \leq d) \triangleq \lim_{e \prec c} \Pi_e^{\blacktriangleright F(d)}$$

Internal fixpoints

$\eta \colon A \to A$ is contractive if $\eta \equiv \eta' \circ \mathsf{next}$ for some η'.

Internal fixpoints.

Let $\eta \colon A \to A$ be contractive. Then there exists a unique $\mu\eta \colon 1 \to A$, such that $\eta \circ \mu\eta \equiv \eta$.

- Sheaf properties are used in fixpoint construction.
- But $\blacktriangleright$ always gives us a sheaf.

Internal fixpoint

$$
\begin{aligned}
\mu \colon (\blacktriangleright X \to X) &\to X \\
\mu_0(f) &\triangleq f_0(*) \\
\mu_{i+1}(f) &\triangleq f_{i+1}(\mu_i(f)) \\
\mu_\kappa(f) &\triangleq f_\kappa(\text{`glue' all known } \mu_{i \prec \kappa}(f)) \quad \text{(sheaf properties)}
\end{aligned}
$$

Insights

- **Insight 1**: There is an adjoint equivalence between sheaves and presheaves.
- **Insight 2**: Sheaf condition is used only when working with $\blacktriangleright F$, but $\blacktriangleright F$ is always a sheaf.

Recursive domain equations

- Presheaves form a CCC category.
- Arrows of presheaves are represented by exponentials, ‘hom-sets’ (self-enrichment).
- F is locally contractive, if its action on hom-sets is contractive.
- And the witness is functorial.
- The proof mostly follows the original work, except for a different tower construction.

Theorem

The locally contractive functor F has a solution.

Recursive domain equations

- Sheaves: $X_0 ::= 1 \quad X_1 ::= F(1) \quad \ldots \quad X_\omega ::= \lim_{i \prec \omega} X_i \quad \ldots$.
- Presheaves: $X_0 ::= F(1) \quad X_1 ::= F(F(1)) \quad \ldots \quad X_\omega ::= F(\lim_{i \prec \omega} X_i) \quad \ldots$

The sequence above is a diagram (also called a tower), and its limit is a solution to F.

Insights

- **Insight 1**: There is an adjoint equivalence between sheaves and presheaves.
- **Insight 2**: Sheaf condition is needed only when working with $\blacktriangleright F$, but $\blacktriangleright F$ is always a sheaf.
- **Insight 3**: Different tower construction.

Example

```
*Some instances*
Example simpl_gitree_dom
  := (functor_compose exp_func later)
     + (Discr nat)
     + (lift later).
Lemma simpl_gitree_dom_lc
  : LocallyContractiveFunctor simpl_gitree_dom.
Proof.
  *8 lines*
Qed.
Lemma simpl_gitree_dom_sol : bifunc_solution simpl_gitree_dom.
Proof.
  *5 lines*
Qed.
```

Big picture

$$
\begin{tikzcd}
\mathcal{OFE} \arrow[r, shift left] \arrow[r, phantom, "\perp"] \arrow[d, shift left] \arrow[d, phantom, "\vdash"] & \color{green}{\hat{\lambda}} \arrow[l, shift left] \arrow[d, shift right, "a"'] \arrow[d, phantom, "\vdash"] \arrow[d, bend left, color=green, "\dashv \; \equiv"] \\
\color{red}{?} \arrow[u, hook, shift left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathrm{Sh}(\lambda) \arrow[l, shift left] \arrow[u, hook, bend left, "i"] \arrow[u, shift right, color=green] \\
\mathcal{COFE} \arrow[u, hook] \arrow[uu, hook, bend left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathcal{E} \arrow[l, shift left] \arrow[u, hook]
\end{tikzcd}
$$

Implementation-oriented ideal solution

A reflective subcategory of $\mathcal{OFE}$ ($\color{red}{?}$), such that $\mathcal{COFE}$ embeds into it, and it is connected to $\mathrm{Sh}(\lambda)$ via an adjunction.

Theoretical ideal solution

Working with $\mathrm{Sh}(\lambda)$.

Our solution

Instead of working with sheaves, use presheaves.

Summary

Implemented:

- Internal fixpoints;
- Recursive domain equations;
- Internal logic.

Our assumptions:

- Axiom of choice;
- Propositional extensionality;
- Functional extensionality.

Placeholder before backup slides

Bonus: setoid hell

- Essentially, we construct a functor from ordinals to algebras over F.
- We construct the solution functor by induction.
- Given a functor $T : \{\beta \mid \beta \prec \alpha\} \to \mathbf{Alg}(F)$, we construct a functor $\{\beta \mid \beta \preceq \alpha\} \to \mathbf{Alg}(F)$ by assigning a F applied to the limit of T (the extension) at stage α.
- Canonical partial solution is a functor from some down-set that is constructed like that at all stages.
- Two canonical partial solutions should be equal.
- We need to show that limits of setoid equivalent functors (induction hypothesis) are equivalent.
    - Need to lower the level of abstraction, and look at the limits.
    - The resulting algebras have equal carriers, but only equivalent (modulo casts) morphisms.