Foundations for Proof Search in Coinductive Horn Clause Theories

Henning Basold

CNRS, ENS Lyon

Joint work with Ekaterina Komendantskaya and Yue Li from Heriot-Watt University

Chocola Seminar
20 September 2018

Coinductive Horn Clause Theories

–

An Introduction

What The Heck Are Coinductive Horn Clause Theories?

- Horn clauses that describe observations
- Canonical Herbrand model is a greatest fixed point
- Why would we want that?
  - Coinductive programming (web-server, control systems, . . .)
  - Coinductive data types (streams, delayed computations, . . .)
  - Mutual type class instances in Haskell
  - Coinductive predicates (bisimilarity, modal logic, . . .)
  - Let your imagination run free!

Let us look at three examples.

What The Heck Are Coinductive Horn Clause Theories?

- Horn clauses that describe observations
- Canonical Herbrand model is a greatest fixed point
- Why would we want that?
  - Coinductive programming (web-server, control systems, ...)
  - Coinductive data types (streams, delayed computations, ...)
  - Mutual type class instances in Haskell
  - Coinductive predicates (bisimilarity, modal logic, ...)
  - Let your imagination run free!

Let us look at three examples.

Example 1 – Coinductive Data Types

Example (Data types of natural numbers and streams)

$$
\begin{aligned}
\kappa_{\text{nat0}} &: \forall x. & & \text{nat } 0 \\
\kappa_{\text{nat1}} &: \forall x. \text{ nat } x & \to & \text{ nat } (s\ x) \\
\kappa_{\text{stream}} &: \forall x. \text{ nat } x \ \wedge \ \text{stream } y & \to & \text{ stream } (\text{scons } x\ y)
\end{aligned}
$$

Our goal

Prove

$$\exists x. \text{ stream } x$$

with x a term that represents stream of zeros:

$$\text{scons } 0 \ (\text{scons } 0 \ \ldots)$$

Example 1 – Coinductive Data Types

Example (Data types of natural numbers and streams)

$$
\begin{aligned}
\kappa_{\mathrm{nat0}} &: \forall x. & & \mathrm{nat}\ 0 \\
\kappa_{\mathrm{nat1}} &: \forall x.\, \mathrm{nat}\ x & \rightarrow\ & \mathrm{nat}\ (s\ x) \\
\kappa_{\mathrm{stream}} &: \forall x.\, \mathrm{nat}\ x\ \wedge\ \mathrm{stream}\ y & \rightarrow\ & \mathrm{stream}\ (\mathrm{scons}\ x\ y)
\end{aligned}
$$

Our goal

Prove

$$\exists x.\, \mathrm{stream}\ x$$

with x a term that represents stream of zeros:

$$\mathrm{scons}\ 0\ (\mathrm{scons}\ 0\ \ldots)$$

Example 2 – Coinductive Programs

Example (Enumerating natural numbers)

$$\kappa_{\text{from}} : \forall x\ y.\, \text{from}\ (s\ x)\ y \to \text{from}\ x\ (\text{scons}\ x\ y)$$

Our goal

Prove

$$\exists x.\, \text{from}\ 0\ x$$

with x a term that represents

$$\text{scons}\ 0\ (\text{scons}\ (s\ 0)\ (\text{scons}\ (s\ (s\ 0))\ \ldots))$$

Example 2 – Coinductive Programs

Example (Enumerating natural numbers)

$$\kappa_{\text{from}} : \forall x\ y.\, \text{from}\ (s\ x)\ y \to \text{from}\ x\ (\text{scons}\ x\ y)$$

Our goal

Prove

$$\exists x.\, \text{from}\ 0\ x$$

with x a term that represents

$$\text{scons}\ 0\ (\text{scons}\ (s\ 0)\ (\text{scons}\ (s\ (s\ 0))\ \ldots))$$

Example 3 – Type Class Inference

Example (As Haskell declaration)

data OddList a = OCons a (EvenList a)
data EvenList a = Nil | ECons a (OddList a)

instance(**Eq** a, **Eq** (EvenList a)) => **Eq** (OddList a) **where** ...
instance(**Eq** a, **Eq** (OddList a)) => **Eq** (EvenList a) **where** ...

Example (As Horn clause theory with some base type $\mathbf{i}$)

$$\kappa_{\mathbf{i}} : \text{eq } \mathbf{i}$$

$$\kappa_{\text{odd}} : \forall x.\, \text{eq } x \wedge \text{eq } (\text{even } x) \rightarrow \text{eq } (\text{odd } x)$$

$$\kappa_{\text{even}} : \forall x.\, \text{eq } x \wedge \text{eq } (\text{odd } x) \rightarrow \text{eq } (\text{even } x)$$

Example 3 – Type Class Inference

Example (As Haskell declaration)

```
data OddList  a  =  OCons a (EvenList  a)
data EvenList a  =  Nil | ECons a (OddList a)

instance(Eq a, Eq (EvenList a)) => Eq (OddList a) where ...
instance(Eq a, Eq (OddList  a)) => Eq (EvenList a) where ...
```

Example (As Horn clause theory with some base type $\mathbf{i}$)

$$
\begin{aligned}
\kappa_{\mathbf{i}} &: \mathrm{eq}\ \mathbf{i} \\
\kappa_{\mathrm{odd}} &: \forall x.\, \mathrm{eq}\ x \wedge \mathrm{eq}\ (\mathrm{even}\ x) \rightarrow \mathrm{eq}\ (\mathrm{odd}\ x) \\
\kappa_{\mathrm{even}} &: \forall x.\, \mathrm{eq}\ x \wedge \mathrm{eq}\ (\mathrm{odd}\ x) \rightarrow \mathrm{eq}\ (\mathrm{even}\ x)
\end{aligned}
$$

Example 3 – Type Class Inference

Example (As Horn clause theory with some base type $\mathbf{i}$)

$$\begin{aligned} \kappa_{\mathbf{i}} &: \mathrm{eq}\ \mathbf{i} \\ \kappa_{\mathrm{odd}} &: \forall x.\, \mathrm{eq}\ x \wedge \mathrm{eq}\ (\mathrm{even}\ x) \to \mathrm{eq}\ (\mathrm{odd}\ x) \\ \kappa_{\mathrm{even}} &: \forall x.\, \mathrm{eq}\ x \wedge \mathrm{eq}\ (\mathrm{odd}\ x)\ \ \to \mathrm{eq}\ (\mathrm{even}\ x) \end{aligned}$$

Our goal

Prove

$$\mathrm{eq}\ (\mathrm{odd}\ \mathbf{i})$$

and provide the proof object for the type checker

$\Longrightarrow$ constructive proofs and discovery of proof objects

Example 3 – Type Class Inference

Example (As Horn clause theory with some base type $\mathbf{i}$)

$$\begin{aligned}\kappa_{\mathbf{i}} &: \mathrm{eq}\ \mathbf{i} \\ \kappa_{\mathrm{odd}} &: \forall x.\, \mathrm{eq}\ x \wedge \mathrm{eq}\ (\mathrm{even}\ x) \to \mathrm{eq}\ (\mathrm{odd}\ x) \\ \kappa_{\mathrm{even}} &: \forall x.\, \mathrm{eq}\ x \wedge \mathrm{eq}\ (\mathrm{odd}\ x) \ \to \mathrm{eq}\ (\mathrm{even}\ x)\end{aligned}$$

Our goal

Prove

$$\mathrm{eq}\ (\mathrm{odd}\ \mathbf{i})$$

and provide the proof object for the type checker
$\Longrightarrow$ constructive proofs and discovery of proof objects

Inductive Horn Clause Theories

- The classical interpretation of logic programs
- Horn clauses that describe constructions
- Canonical Herbrand model is a least fixed point
- Proofs and terms must be finite and non-circular

Example (Natural numbers revisited)

$$\kappa_{\text{nat0}} : \forall x. \quad \text{nat } 0$$
$$\kappa_{\text{nat1}} : \forall x. \text{nat } x \to \text{nat } (s\ x)$$

Interpretation	Possible instances for x in $\text{nat } x$
Inductive	0, $s\ 0$, $s\ (s\ 0)$, $\ldots$
Coinductive	0, $s\ 0$, $s\ (s\ 0)$, $\ldots$ and s^ω with $s^\omega = s\ s^\omega$

Inductive Horn Clause Theories

- The classical interpretation of logic programs
- Horn clauses that describe constructions
- Canonical Herbrand model is a least fixed point
- Proofs and terms must be finite and non-circular

Example (Natural numbers revisited)

$$\begin{aligned} \kappa_{\mathrm{nat0}} &: \forall x. & &\mathrm{nat}\ 0 \\ \kappa_{\mathrm{nat1}} &: \forall x.\, \mathrm{nat}\ x \rightarrow & &\mathrm{nat}\ (s\ x) \end{aligned}$$

Interpretation	Possible instances for x in $\mathrm{nat}\ x$
Inductive	$0,\ s\ 0,\ s\ (s\ 0),\ \ldots$
Coinductive	$0,\ s\ 0,\ s\ (s\ 0),\ \ldots$ and s^ω with $s^\omega = s\ s^\omega$

Goal

A theory of **search** and **models** for **constructive** proofs in **coinductive** Horn clause theories

Other Approaches

- **Circular Unifiers** (Gupta, Simon, et al., 2006/07) – does not cover the examples from and eq
- **CIRC** (Roșu and Lucanu, 2009) – only for bisimilarity but not general Horn clause theories
- **SMT-based** (Reynolds and Kunak, 2015; Blanchette et al., 2018) – classical and no proof objects
- Typically not concerned with algorithmic proof search: lattice theory, game theory, type theory, cyclic proofs
- No constructive approach

Outline

Fixed Point Terms and Circular Unification

Why Fixed Point Terms?

Recall the stream of zeros:

$$\text{scons } 0 \ (\text{scons } 0 \ \cdots)$$

As circular unifier

$x = \text{scons } 0 \ x$

As fixed point term

$\text{fix } x. \, \text{scons } 0 \ x$

Typed λ-Terms With Fixed Points

Types and Signatures

$$\mathbb{T} \ni \sigma, \tau ::= \iota \in \mathbb{B} \mid \sigma \to \tau$$

Signature is a set Σ of pairs $c : \tau$, where $\tau \in \mathbb{T}$.

Terms (Simply typed λ-calculus with fixed points)

$$\frac{c : \tau \in \Sigma}{\Gamma \vdash c : \tau} \qquad \frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash M\ N : \tau}$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.\, M : \sigma \to \tau} \qquad \frac{\Gamma, x : \tau \vdash M : \tau}{\Gamma \vdash \mathrm{fix}\, x.\, M : \tau}$$

Operational Semantics

$$(\lambda x.\, M)N \longrightarrow M[N/x] \qquad (\mathrm{fix}\, x.\, M) \longrightarrow M[\mathrm{fix}\, x.\, /x]$$

Example

Recall the enumeration of natural numbers

$$\kappa_{\text{from}} : \forall x\ y.\, \text{from}\ (s\ x)\ y \to \text{from}\ x\ (\text{scons}\ x\ y)$$

and the term

$$\text{scons}\ 0\ (\text{scons}\ (s\ 0)\ (\text{scons}\ (s\ (s\ 0)) \ldots))$$

Representation as fixed point term

Define

$$s_{\text{fr}} = \text{fix}\ f.\, \lambda x.\, \text{scons}\ x\ (f\ (s\ x)),$$

then $s_{\text{fr}} : \iota \to \iota$.

Example

Recall the enumeration of natural numbers

$$\kappa_{\mathrm{from}} : \forall x\ y.\, \mathrm{from}\ (s\ x)\ y \to \mathrm{from}\ x\ (\mathrm{scons}\ x\ y)$$

and the term

$$\mathrm{scons}\ 0\ (\mathrm{scons}\ (s\ 0)\ (\mathrm{scons}\ (s\ (s\ 0))\ldots))$$

Representation as fixed point term

Define

$$s_{\mathrm{fr}} = \mathrm{fix}\ f.\, \lambda x.\, \mathrm{scons}\ x\ (f\ (s\ x)),$$

then $s_{\mathrm{fr}} : \iota \to \iota$.

Guarded Terms

- Not all fixed point terms are productive:

$$M \longrightarrow\!\!\!\!\!\rightarrow c\, M', \text{ for } c \in \Sigma$$

- Example: $\operatorname{fix} x.\, x$
- Guarded terms are syntactically defined productive terms
- Can be unfolded to elements in Σ^∞, which are potentially infinite trees with nodes in Σ
- NB: Semantics use that Σ^∞ is a final coalgebra
- Circular unifiers give guarded terms

Guarded Terms

- Not all fixed point terms are productive:

$$M \longrightarrow\!\!\!\!\rightarrow c\, M', \text{ for } c \in \Sigma$$

- Example: $\mathrm{fix}\, x.\, x$
- Guarded terms are syntactically defined productive terms
- Can be unfolded to elements in Σ^∞, which are potentially infinite trees with nodes in Σ
- NB: Semantics use that Σ^∞ is a final coalgebra
- Circular unifiers give guarded terms

Guarded Terms

- Not all fixed point terms are productive:

$$M \twoheadrightarrow c\, M', \text{ for } c \in \Sigma$$

- Example: $\mathrm{fix}\, x.\, x$
- Guarded terms are syntactically defined productive terms
- Can be unfolded to elements in Σ^∞, which are potentially infinite trees with nodes in Σ
- NB: Semantics use that Σ^∞ is a final coalgebra
- Circular unifiers give guarded terms

Constructive Coinductive Proofs

Recursive Proofs

- Recursion as first step to proof search
- Eliminates the need to find invariants like in lattices:

$$\frac{x \leq y \leq f(y)}{x \leq \nu f}$$

- Recursion will be controlled by the so-called later modality
- Gives $\mathbf{iFOL}_{\blacktriangleright}$ – an extension of intuitionistic first-order logic

Formulas and Theories

Predicate Signatures

Set Π of pairs $p : \tau_1 \to \cdots \to \tau_n \to o$, where $\tau_k \in \mathbb{T}$ and $o \notin \mathbb{T}$

Formulas

$$
\begin{aligned}
\varphi, \psi ::= {} & p\ M_1 \cdots\ M_n, \quad p \in \Pi \\
& \mid \blacktriangleright \varphi \\
& \mid \top \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \forall x : \tau.\, \varphi \mid \exists x : \tau.\, \varphi
\end{aligned}
$$

Horn Clause

$\forall \vec{x}.\, (A_1 \wedge \cdots \wedge A_n) \to B$, where $A_1, \ldots, A_n$ and B are atoms

Horn clause theory or logic program

Finite set P of Horn clauses

Proof System

Standard First-Order Intuitionistic Logic plus

Rules for the later modality

$$\frac{\Gamma \mid \Delta \vdash \varphi}{\Gamma \mid \Delta \vdash \blacktriangleright \varphi}\ \textbf{(Next)} \qquad \frac{\Gamma \mid \Delta \vdash \blacktriangleright(\varphi \to \psi)}{\Gamma \mid \Delta \vdash \blacktriangleright \varphi \to \blacktriangleright \psi}\ \textbf{(Mon)}$$

$$\frac{\Gamma \mid \Delta, \blacktriangleright \varphi \vdash \varphi}{\Gamma \mid \Delta \vdash \varphi}\ \textbf{(Löb)}$$

Axioms for coinductive Horn clause theories P

$$\frac{\forall \vec{x}.\, (A_1 \wedge \cdots \wedge A_n) \to B \in P}{\Gamma \mid \Delta \vdash \forall \vec{x}.\, (\blacktriangleright A_1 \wedge \cdots \wedge \blacktriangleright A_n) \to B}$$

Example: Type class inference

Horn clause theory

$$
\begin{aligned}
P = \{\kappa_{\mathbf{i}} &: \mathrm{eq}\ \mathbf{i} \\
\kappa_{\mathrm{odd}} &: \forall x.\, \mathrm{eq}\ x \wedge \mathrm{eq}\ (\mathrm{even}\ x) \rightarrow \mathrm{eq}\ (\mathrm{odd}\ x) \\
\kappa_{\mathrm{even}} &: \forall x.\, \mathrm{eq}\ x \wedge \mathrm{eq}\ (\mathrm{odd}\ x) \ \rightarrow \mathrm{eq}\ (\mathrm{even}\ x)\}
\end{aligned}
$$

Resulting axioms

$$
\begin{aligned}
&\Gamma \mid \Delta \vdash \mathrm{eq}\ \mathbf{i} \\
&\Gamma \mid \Delta \vdash \forall x.\ \blacktriangleright(\mathrm{eq}\ x) \wedge \blacktriangleright(\mathrm{eq}\ (\mathrm{even}\ x)) \rightarrow \mathrm{eq}\ (\mathrm{odd}\ x) \\
&\Gamma \mid \Delta \vdash \forall x.\ \blacktriangleright(\mathrm{eq}\ x) \wedge \blacktriangleright(\mathrm{eq}\ (\mathrm{odd}\ x)) \rightarrow \mathrm{eq}\ (\mathrm{even}\ x)
\end{aligned}
$$

Example: Proof for Type Class Instance

$$
\dfrac{\dfrac{\dfrac{\dfrac{\kappa_{\text{odd}}}{\Delta \vdash \forall x.\ \blacktriangleright(\text{eq}\ x) \land \blacktriangleright(\text{eq}\ (\text{even}\ x)) \to \text{eq}\ (\text{odd}\ x)}}{\Delta \vdash \blacktriangleright(\text{eq}\ \mathbf{i}) \land \blacktriangleright(\text{eq}\ (\text{even}\ \mathbf{i})) \to \text{eq}\ (\text{odd}\ \mathbf{i})} \qquad \spadesuit}{\blacktriangleright(\text{eq}\ (\text{odd}\ \mathbf{i})) \vdash \text{eq}\ (\text{odd}\ \mathbf{i})}}{\vdash \text{eq}\ (\text{odd}\ \mathbf{i})}\ \textbf{(Löb)}
$$

$$
\spadesuit : \quad \dfrac{\textbf{(Next)}\ \dfrac{\dfrac{\kappa_{\mathbf{i}}}{\Delta \vdash \text{eq}\ \mathbf{i}}}{\Delta \vdash \blacktriangleright(\text{eq}\ \mathbf{i})} \qquad \dfrac{\dfrac{\diamondsuit}{\Delta \vdash \text{eq}\ (\text{even}\ \mathbf{i})}}{\Delta \vdash \blacktriangleright(\text{eq}\ (\text{even}\ \mathbf{i}))}\ \textbf{(Next)}}{\Delta \vdash \blacktriangleright(\text{eq}\ \mathbf{i}) \land \blacktriangleright(\text{eq}\ (\text{even}\ \mathbf{i}))}
$$

$$
\diamondsuit : \quad \dfrac{\dfrac{\dfrac{\kappa_{\text{even}}}{\vdots}}{\Delta \vdash \blacktriangleright(\text{eq}\ (\text{odd}\ \mathbf{i})) \to \text{eq}\ (\text{even}\ \mathbf{i})} \qquad \blacktriangleright(\text{eq}\ (\text{odd}\ \mathbf{i})) \in \Delta}{\Delta \vdash \text{eq}\ (\text{even}\ \mathbf{i})}
$$

Example: Proof for Type Class Instance

$$\dfrac{\dfrac{\dfrac{\kappa_{\text{odd}}}{\Delta \vdash \forall x.\ \blacktriangleright(\text{eq}\ x) \wedge \blacktriangleright(\text{eq}\ (\text{even}\ x)) \rightarrow \text{eq}\ (\text{odd}\ x)}}{\Delta \vdash \blacktriangleright(\text{eq}\ \mathbf{i}) \wedge \blacktriangleright(\text{eq}\ (\text{even}\ \mathbf{i})) \rightarrow \text{eq}\ (\text{odd}\ \mathbf{i})} \qquad \spadesuit}{\dfrac{\blacktriangleright(\text{eq}\ (\text{odd}\ \mathbf{i})) \vdash \text{eq}\ (\text{odd}\ \mathbf{i})}{\vdash \text{eq}\ (\text{odd}\ \mathbf{i})}\ \textbf{(Löb)}}$$

$$\spadesuit: \quad \textbf{(Next)}\ \dfrac{\dfrac{\dfrac{\kappa_{\mathbf{i}}}{\Delta \vdash \text{eq}\ \mathbf{i}}}{\Delta \vdash \blacktriangleright(\text{eq}\ \mathbf{i})} \qquad \dfrac{\dfrac{\diamondsuit}{\Delta \vdash \text{eq}\ (\text{even}\ \mathbf{i})}}{\Delta \vdash \blacktriangleright(\text{eq}\ (\text{even}\ \mathbf{i}))}\ \textbf{(Next)}}{\Delta \vdash \blacktriangleright(\text{eq}\ \mathbf{i}) \wedge \blacktriangleright(\text{eq}\ (\text{even}\ \mathbf{i}))}$$

$$\diamondsuit: \quad \dfrac{\dfrac{\dfrac{\kappa_{\text{even}}}{\vdots}}{\Delta \vdash \blacktriangleright(\text{eq}\ (\text{odd}\ \mathbf{i})) \rightarrow \text{eq}\ (\text{even}\ \mathbf{i})} \qquad \blacktriangleright(\text{eq}\ (\text{odd}\ \mathbf{i})) \in \Delta}{\Delta \vdash \text{eq}\ (\text{even}\ \mathbf{i})}$$

Example: Proof for Type Class Instance

$$\dfrac{\dfrac{\dfrac{\kappa_{\text{odd}}}{\Delta \vdash \forall x.\ \blacktriangleright(\text{eq } x) \wedge \blacktriangleright(\text{eq } (\text{even } x)) \to \text{eq } (\text{odd } x)}}{\Delta \vdash \blacktriangleright(\text{eq } \mathbf{i}) \wedge \blacktriangleright(\text{eq } (\text{even } \mathbf{i})) \to \text{eq } (\text{odd } \mathbf{i})} \qquad \spadesuit}{\dfrac{\blacktriangleright(\text{eq } (\text{odd } \mathbf{i})) \vdash \text{eq } (\text{odd } \mathbf{i})}{\vdash \text{eq } (\text{odd } \mathbf{i})}\ \textbf{(Löb)}}$$

$$\spadesuit : \quad \dfrac{\textbf{(Next)}\ \dfrac{\dfrac{\kappa_{\mathbf{i}}}{\Delta \vdash \text{eq } \mathbf{i}}}{\Delta \vdash \blacktriangleright(\text{eq } \mathbf{i})} \qquad \dfrac{\dfrac{\diamondsuit}{\Delta \vdash \text{eq } (\text{even } \mathbf{i})}}{\Delta \vdash \blacktriangleright(\text{eq } (\text{even } \mathbf{i}))}\ \textbf{(Next)}}{\Delta \vdash \blacktriangleright(\text{eq } \mathbf{i}) \wedge \blacktriangleright(\text{eq } (\text{even } \mathbf{i}))}$$

$$\diamondsuit : \quad \dfrac{\dfrac{\dfrac{\kappa_{\text{even}}}{\vdots}}{\Delta \vdash \blacktriangleright(\text{eq } (\text{odd } \mathbf{i})) \to \text{eq } (\text{even } \mathbf{i})} \qquad \blacktriangleright(\text{eq } (\text{odd } \mathbf{i})) \in \Delta}{\Delta \vdash \text{eq } (\text{even } \mathbf{i})}$$

Coinductive Uniform Proofs
–
Foundations for Proof Search in Coinductive Horn Clause Theories

What and Why Uniform Proofs?

Issues with $\mathbf{iFOL}_{\blacktriangleright}$

- Recursion can be started anywhere
- Proof system has cut rule (through implication)
- $\implies$ Prevents algorithmic proof search

Towards proof search

- Fix where recursion can start
- Eliminate cut, while preserving implication
- Operational semantics for proofs that correspond to resolution
- $\implies$ Proof search is semi-decidable resolution strategy

What and Why Uniform Proofs?

Issues with $\mathbf{iFOL}_{\blacktriangleright}$

- Recursion can be started anywhere
- Proof system has cut rule (through implication)
- $\Longrightarrow$ Prevents algorithmic proof search

Towards proof search

- Fix where recursion can start
- Eliminate cut, while preserving implication
- Operational semantics for proofs that correspond to resolution
- $\Longrightarrow$ Proof search is semi-decidable resolution strategy

Coinductive Uniform Proofs (CUP)

Definite clauses and goal formulas

The operational semantics for proofs use specific formula shapes:

- **Definite clauses** denoted by D
- **Goal formulas** denoted by G

Proof steps (judgements)

$\Sigma; P \looparrowright \varphi$	φ is proven coinductively from P
$\Sigma; P; \Delta \Longrightarrow \langle \varphi \rangle$	φ in proven uniformly from P and coinduction hypothesis in Δ, while forcing progress
$\Sigma; \Delta \Longrightarrow G$	G is proven uniformly from P
$\Sigma; \Delta \xRightarrow{D} A$	A has to be proven from D

Starting a coinductive uniform proof

$$\frac{\Sigma; P; \varphi \Longrightarrow \langle \varphi \rangle}{\Sigma; P \looparrowright \varphi} \text{ co-fix}$$

Controlling the use of the coinduction hypothesis

$$\frac{\Sigma; P \cup \Delta \xrightarrow{D} A \qquad D \in P}{\Sigma; P; \Delta \Longrightarrow \langle A \rangle} \text{ decide}\langle\rangle$$

$$\frac{\Sigma; P, \varphi_1; \Delta \Longrightarrow \langle \varphi_2 \rangle}{\Sigma; P; \Delta \Longrightarrow \langle \varphi_1 \to \varphi_2 \rangle} \to R\langle\rangle$$

$$\frac{\Sigma; P; \Delta \Longrightarrow \langle \varphi_1 \rangle \qquad \Sigma; P; \Delta \Longrightarrow \langle \varphi_2 \rangle}{\Sigma; P; \Delta \Longrightarrow \langle \varphi_1 \wedge \varphi_2 \rangle} \wedge R\langle\rangle$$

$$\vdots$$

Starting a coinductive uniform proof

$$\frac{\Sigma; P; \varphi \Longrightarrow \langle \varphi \rangle}{\Sigma; P \looparrowright \varphi}\ \textsf{co-fix}$$

Controlling the use of the coinduction hypothesis

$$\frac{\Sigma; P \cup \Delta \overset{D}{\Longrightarrow} A \qquad D \in P}{\Sigma; P; \Delta \Longrightarrow \langle A \rangle}\ \textsf{decide}\langle\rangle$$

$$\frac{\Sigma; P, \varphi_1; \Delta \Longrightarrow \langle \varphi_2 \rangle}{\Sigma; P; \Delta \Longrightarrow \langle \varphi_1 \to \varphi_2 \rangle}\ {\to}R\langle\rangle$$

$$\frac{\Sigma; P; \Delta \Longrightarrow \langle \varphi_1 \rangle \qquad \Sigma; P; \Delta \Longrightarrow \langle \varphi_2 \rangle}{\Sigma; P; \Delta \Longrightarrow \langle \varphi_1 \wedge \varphi_2 \rangle}\ \wedge R\langle\rangle$$

$$\vdots$$

Standard rules of uniform proofs

$$\frac{\Sigma; \Delta \xrightarrow{D} A \qquad D \in P \cup \Delta}{\Sigma; \Delta \Longrightarrow A}\ \textsf{decide} \qquad \frac{A \equiv A'}{\Sigma; \Delta \xrightarrow{A'} A}\ \textsf{initial}$$

$$\frac{\Sigma; \Delta \xrightarrow{D} A \qquad \Sigma; \Delta \Longrightarrow G}{\Sigma; \Delta \xrightarrow{G \to D} A} \to\! L \qquad \frac{\Sigma; P, D \Longrightarrow G}{\Sigma; \Delta \Longrightarrow D \to G} \to\! R$$

$$\frac{\Sigma; \Delta \xrightarrow{D_x} A \qquad x \in \{1, 2\}}{\Sigma; \Delta \xrightarrow{D_1 \wedge D_2} A} \wedge L$$

$$\frac{\Sigma; \Delta \Longrightarrow G_1 \qquad \Sigma; \Delta \Longrightarrow G_2}{\Sigma; \Delta \Longrightarrow G_1 \wedge G_2} \wedge R$$

$$\vdots$$

Example

Define

$$\kappa_{\mathrm{from}} : \forall x\ y.\, \mathrm{from}\ (s\ x)\ y \to \mathrm{from}\ x\ (\mathrm{scons}\ x\ y)$$

$$\varphi = \forall x.\, \mathrm{from}\ x\ (s_{\mathrm{fr}}\ x)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\spadesuit}{c, \Sigma; P, \varphi \xrightarrow{\mathrm{from}\ (s\ c)\ (s_{\mathrm{fr}}\ c) \to \mathrm{from}\ c\ (\mathrm{scons}\ c\ (s_{\mathrm{fr}}\ c))} \mathrm{from}\ c\ (s_{\mathrm{fr}}\ c)}\ \to L}{c, \Sigma; P, \varphi \xrightarrow{\kappa_{\mathrm{from}}} \mathrm{from}\ c\ (s_{\mathrm{fr}}\ c)}\ \forall L\ \text{(2 times)}}{c, \Sigma; P; \varphi \Longrightarrow \langle \mathrm{from}\ c\ (s_{\mathrm{fr}}\ c) \rangle}\ \mathsf{decide}\langle\rangle}{\Sigma; P; \forall x.\, \mathrm{from}\ x\ (s_{\mathrm{fr}}\ x) \Longrightarrow \langle \forall x.\, \mathrm{from}\ x\ (s_{\mathrm{fr}}\ x) \rangle}\ \forall R\langle\rangle}{\Sigma; P \looparrowright \forall x.\, \mathrm{from}\ x\ (s_{\mathrm{fr}}\ x)}\ \text{co-fix}$$

Example

Define

$$\kappa_{\text{from}} : \forall x\ y.\ \text{from}\ (s\ x)\ y \to \text{from}\ x\ (\text{scons}\ x\ y)$$

$$\varphi = \forall x.\ \text{from}\ x\ (s_{\text{fr}}\ x)$$

$$\dfrac{\clubsuit \qquad \dfrac{\dfrac{\dfrac{}{c, \Sigma; P, \varphi \xRightarrow{\text{from}\ (s\ c)\ (s_{\text{fr}}\ (s\ c))} \text{from}\ (s\ c)\ (s_{\text{fr}}\ (s\ c))}\ \text{initial}}{c, \Sigma; P, \varphi \xRightarrow{\varphi} \text{from}\ (s\ c)\ (s_{\text{fr}}\ (s\ c))}\ \forall L}{c, \Sigma; P, \varphi \Longrightarrow \text{from}\ (s\ c)\ (s_{\text{fr}}\ (s\ c))}\ \text{decide}}{\dfrac{c, \Sigma; P, \varphi \xRightarrow{\text{from}\ (s\ c)\ (s_{\text{fr}}\ c) \to \text{from}\ c\ (\text{scons}\ c\ (s_{\text{fr}}\ c))} \text{from}\ c\ (s_{\text{fr}}\ c)}{\spadesuit}}\ \to L$$

Example

Define

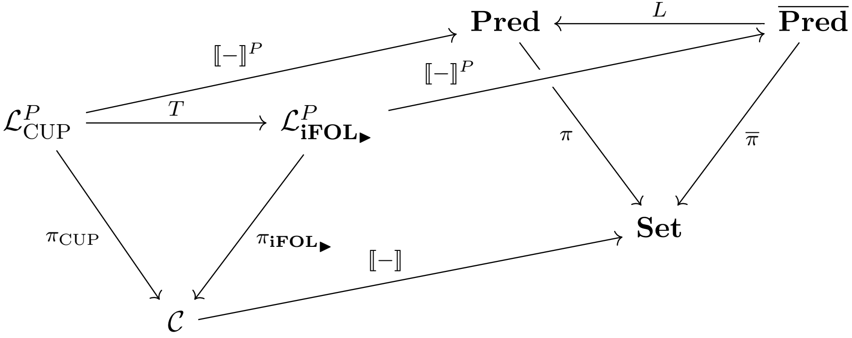
$$\kappa_{\text{from}} : \forall x\ y.\, \text{from}\ (s\ x)\ y \to \text{from}\ x\ (\text{scons}\ x\ y)$$

$$\varphi = \forall x.\, \text{from}\ x\ (s_{\text{fr}}\ x)$$

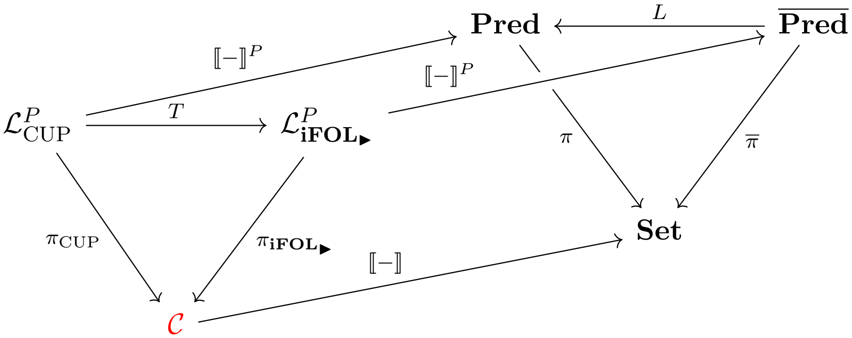
$$\dfrac{\dfrac{\text{scons}\ c\ (s_{\text{fr}}\ (s\ c)) \equiv s_{\text{fr}}\ c}{c, \Sigma; P, \varphi \xrightarrow{\text{from}\ c\ (\text{scons}\ c\ (s_{\text{fr}}\ (s\ c)))} \text{from}\ c\ (s_{\text{fr}}\ c)}\ \text{initial}}{\clubsuit}$$

Relative Soundness and Models

Semantics and Proof Translation

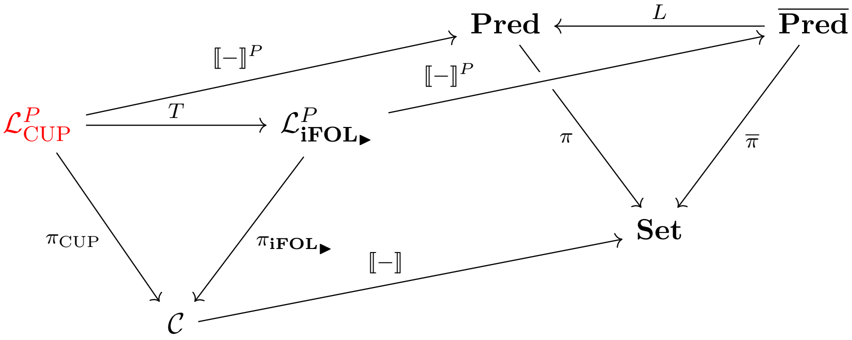


Semantics and Proof Translation



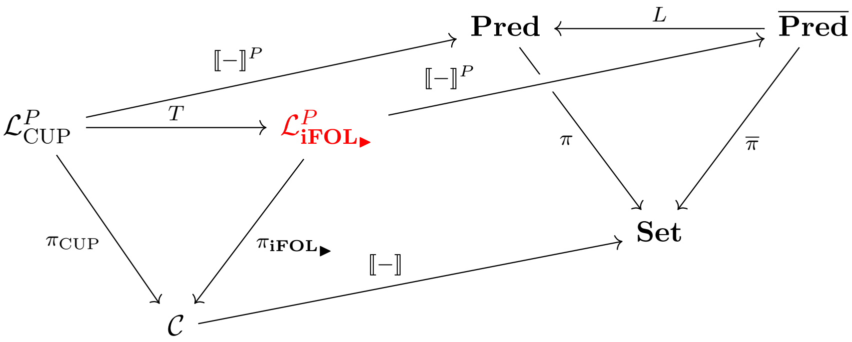
- $\mathcal{C}$ — Contexts and terms
- $\mathcal{L}^P_{\mathrm{CUP}}$ — Formulas and provability in CUP relative to P
- $\mathcal{L}^P_{\mathbf{iFOL}_\blacktriangleright}$ — Formulas and provability in $\mathbf{iFOL}_\blacktriangleright$ relative to P
- $\mathbf{Pred}$ — Set-based predicates
- $\overline{\mathbf{Pred}}$ — Descending chains of predicates (Kripke model)

Semantics and Proof Translation



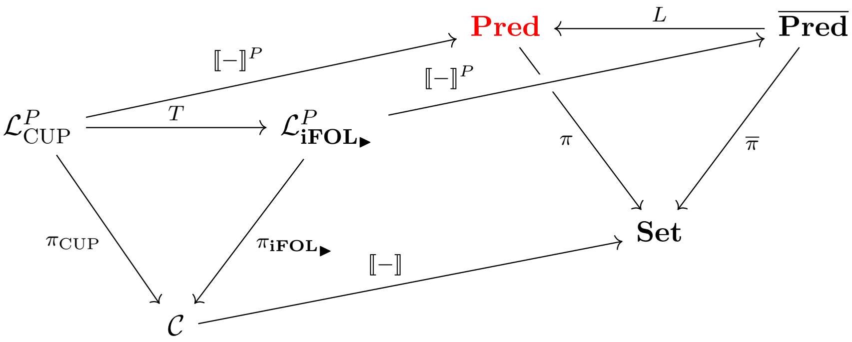
- $\mathcal{C}$ — Contexts and terms
- $\mathcal{L}^P_{\mathrm{CUP}}$ — Formulas and provability in CUP relative to P
- $\mathcal{L}^P_{\mathbf{iFOL}_\blacktriangleright}$ — Formulas and provability in $\mathbf{iFOL}_\blacktriangleright$ relative to P
- $\mathbf{Pred}$ — Set-based predicates
- $\overline{\mathbf{Pred}}$ — Descending chains of predicates (Kripke model)

Semantics and Proof Translation



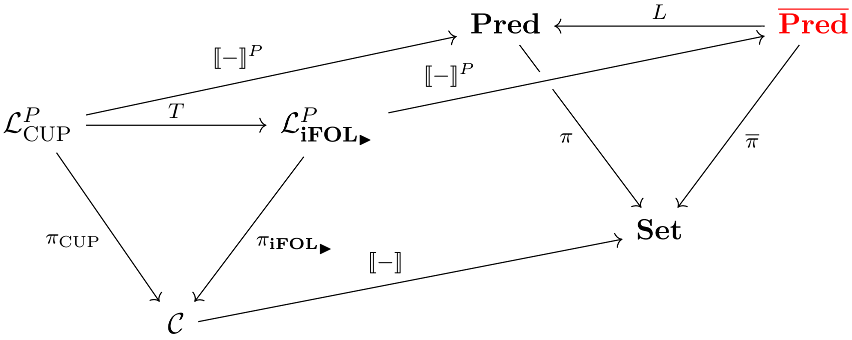
- $\mathcal{C}$ — Contexts and terms
- $\mathcal{L}^P_{\mathrm{CUP}}$ — Formulas and provability in CUP relative to P
- $\mathcal{L}^P_{\mathbf{iFOL}_\blacktriangleright}$ — Formulas and provability in $\mathbf{iFOL}_\blacktriangleright$ relative to P
- $\mathbf{Pred}$ — Set-based predicates
- $\overline{\mathbf{Pred}}$ — Descending chains of predicates (Kripke model)

Semantics and Proof Translation



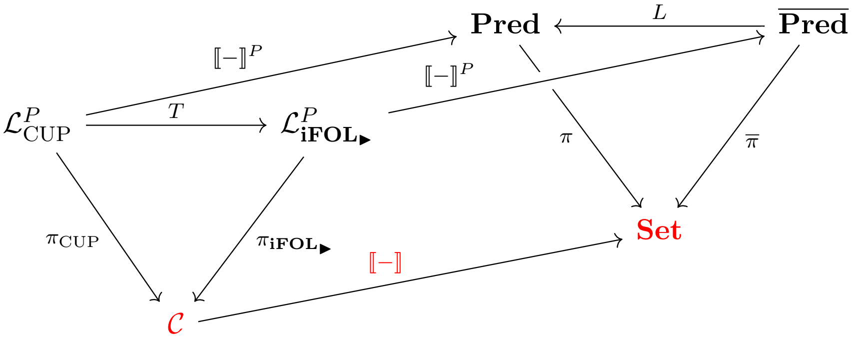
- $\mathcal{C}$ — Contexts and terms
- $\mathcal{L}^P_{\mathrm{CUP}}$ — Formulas and provability in CUP relative to P
- $\mathcal{L}^P_{\mathbf{iFOL}_\blacktriangleright}$ — Formulas and provability in $\mathbf{iFOL}_\blacktriangleright$ relative to P
- $\mathbf{Pred}$ — Set-based predicates
- $\overline{\mathbf{Pred}}$ — Descending chains of predicates (Kripke model)

Semantics and Proof Translation



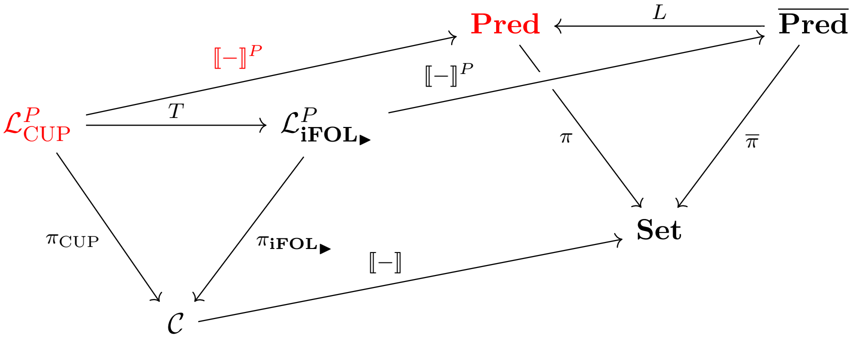
- $\mathcal{C}$ — Contexts and terms
- $\mathcal{L}^P_{\mathrm{CUP}}$ — Formulas and provability in CUP relative to P
- $\mathcal{L}^P_{\mathbf{iFOL}_\blacktriangleright}$ — Formulas and provability in $\mathbf{iFOL}_\blacktriangleright$ relative to P
- $\mathbf{Pred}$ — Set-based predicates
- $\overline{\mathbf{Pred}}$ — Descending chains of predicates (Kripke model)

Semantics and Proof Translation



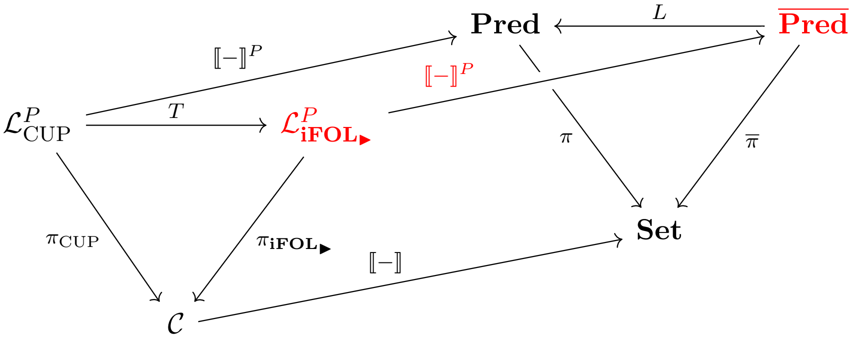
- $[\![-]\!]$ — Semantics of types and terms
- $[\![-]\!]^P$ — Semantics of formulas and soundness
- T — Proof translation
- L — Soundness of Kripke semantics for fixed point model
- NB: All these are maps of first-order fibrations

Semantics and Proof Translation



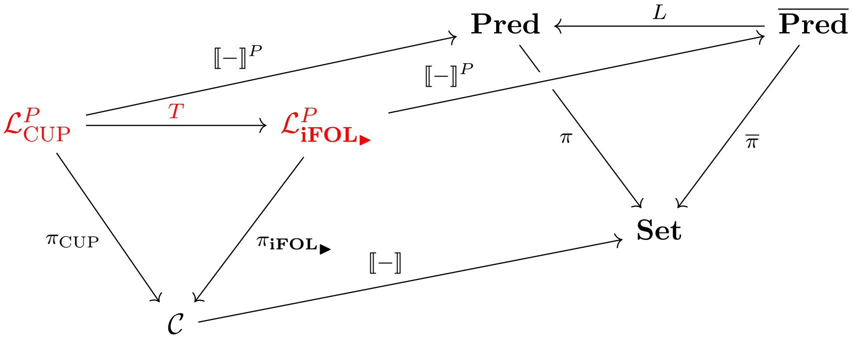
- $[\![-]\!]$ — Semantics of types and terms
- $[\![-]\!]^P$ — Semantics of formulas and soundness
- T — Proof translation
- L — Soundness of Kripke semantics for fixed point model
- NB: All these are maps of first-order fibrations

Semantics and Proof Translation



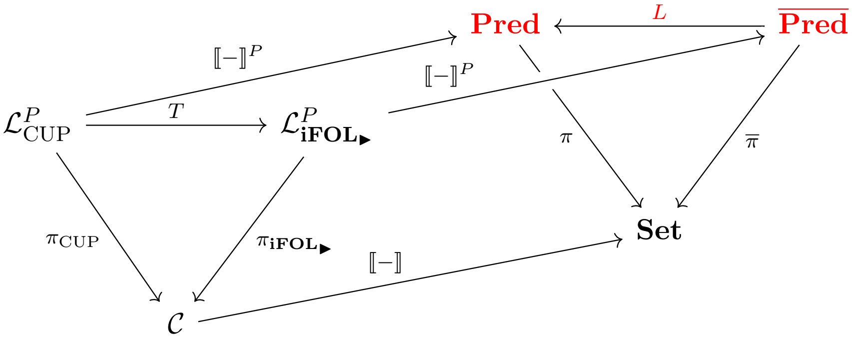
- $[\![-]\!]$ — Semantics of types and terms
- $[\![-]\!]^P$ — Semantics of formulas and soundness
- T — Proof translation
- L — Soundness of Kripke semantics for fixed point model
- NB: All these are maps of first-order fibrations

Semantics and Proof Translation



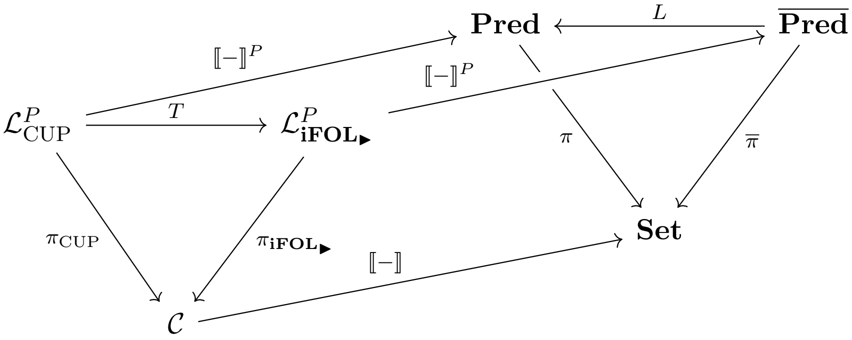
- $[\![-]\!]$ — Semantics of types and terms
- $[\![-]\!]^P$ — Semantics of formulas and soundness
- T — Proof translation
- L — Soundness of Kripke semantics for fixed point model
- NB: All these are maps of first-order fibrations

Semantics and Proof Translation



- $[\![-]\!]$ — Semantics of types and terms
- $[\![-]\!]^P$ — Semantics of formulas and soundness
- T — Proof translation
- L — Soundness of Kripke semantics for fixed point model
- NB: All these are maps of first-order fibrations

Semantics and Proof Translation



- $[\![-]\!]$ — Semantics of types and terms
- $[\![-]\!]^P$ — Semantics of formulas and soundness
- T — Proof translation
- L — Soundness of Kripke semantics for fixed point model
- NB: All these are maps of first-order fibrations

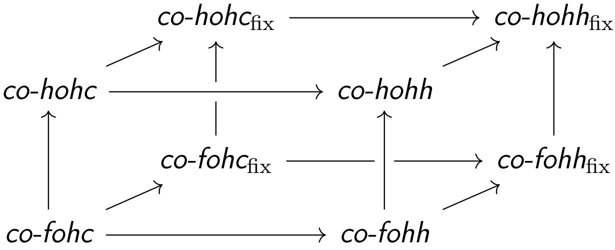
The End

What else is there?

- Heuristics to strengthen goals:

$$\exists t.\, \mathrm{from}\ 0\ t \quad \text{to} \quad \forall x.\, \mathrm{from}\ x\ (s_{\mathrm{fr}}\ x)$$

- Logic classification



What's next?

- Generate proof objects
- Inductive-coinductive Horn clause theories
- Richer types (not just one base type)

Thank you very much for your attention!