Coinduction in Uniform: Foundations for Corecursive Proof Search with Horn Clauses*

Henning Basold[1] [0000−0001−7610−8331], Ekaterina Komendantskaya[2], and Yue Li[2] [0000−0003−0545−0732]

[1] CNRS, ENS Lyon, henning.basold@ens-lyon.fr
[2] Heriot-Watt University, {ek19,yl55}@hw.ac.uk

Abstract. We establish proof-theoretic, constructive and coalgebraic foundations for proof search in coinductive Horn clause theories. Operational semantics of coinductive Horn clause resolution is cast in terms of *coinductive uniform proofs*; its constructive content is exposed via soundness relative to an intuitionistic first-order logic with recursion controlled by the later modality; and soundness of both proof systems is proven relative to a novel coalgebraic description of complete Herbrand models.

Keywords: Horn Clause Logic, Coinduction, Uniform Proofs, Intuitionistic Logic, Coalgebra, Fibrations

1 Introduction

Horn clause logic is a Turing complete and constructive fragment of first-order logic, that plays a central role in verification [20], automated theorem proving [46,50,47] and type inference. Examples of the latter can be traced from the Hindley-Milner type inference algorithm [48,64], to more recent uses of Horn clauses in Haskell type classes [45,24] and in refinement types [38,25]. Its popularity is attributed to well-understood fixed point semantics and an efficient semi-decidable resolution procedure for automated proof search.

According to the standard fixed point semantics [43,46], given a set P of Horn clauses, the *least Herbrand model* for P is the set of all (finite) ground atomic formulae *inductively entailed* by P. For example, the two clauses below define the set of natural numbers in the least Herbrand model.

$$\kappa_{\mathbf{nat}0} : \mathbf{nat}\, 0$$
$$\kappa_{\mathbf{nat}s} : \forall x.\, \mathbf{nat}\, x \to \mathbf{nat}\, (s\, x)$$

Formally, the least Herbrand model for the above two clauses is the set of ground atomic formulae obtained by taking a (forward) closure of the above two clauses. The model for **nat** is given by $\mathcal{N} = \{\mathbf{nat}\, 0,\, \mathbf{nat}\, (s\, 0),\, \mathbf{nat}\, (s\, (s\, 0)), \ldots\}$.

* This work is supported by the European Research Council (ERC) under the EU's Horizon 2020 programme (CoVeCe, grant agreement No. 678157) and by the EPSRC research grant EP/N014758/1.

We can also view Horn clauses coinductively. The *greatest complete Herbrand model* for a set P of Horn clauses is the largest set of finite and infinite ground atomic formulae *coinductively entailed* by P. For example, the greatest complete Herbrand model for the above two clauses is the set

$$\mathcal{N}^{\infty} = \mathcal{N} \cup \{\mathbf{nat}\,(s\,(s\,(\cdots)))\},$$

obtained by taking a backward closure of the above two inference rules on the set of all finite and infinite ground atomic formulae. The *greatest Herbrand model* is the largest set of *finite* ground atomic formulae *coinductively entailed* by P. In our example, it would be given by $\mathcal{N}$ already. Finally, one can also consider the *least complete Hebrand model*, which interprets entailment inductively but over potentially infinite terms. In the case of **nat**, this interpretation does not differ from $\mathcal{N}$. However, finite paths in coinductive structures like transition systems, for example, require such semantics.

The need for coinductive semantics of Horn clauses arises in several scenarios: the Horn clause theory may explicitely define a coinductive data structure or a coinductive relation. However, it may also happen that a Horn clause theory, which is not explicitly intended as coinductive, nevertheless gives rise to infinite inference by resolution and has an interesting coinductive model. This commonly happens in type inference. We will illustrate all these cases by means of examples.

Horn clause theories as coinductive data type declarations The following clause defines the data type of streams of natural numbers.

$$\kappa_{\mathbf{stream}} : \forall x y.\, \mathbf{nat}\, x \,\wedge\, \mathbf{stream}\, y \,\rightarrow \mathbf{stream}\,(\mathrm{scons}\, x\, y)$$

This Horn clause does not have a meaningful inductive, i.e. least fixed point, model. The greatest Herbrand model of this clause is given by

$$\mathcal{S} = \mathcal{N}^{\infty} \cup \{\mathbf{stream}(\mathrm{scons}\, x_0\,(\mathrm{scons}\, x_1\, \cdots)) \mid \mathbf{nat}\, x_0, \mathbf{nat}\, x_1, \ldots \in \mathcal{N}^{\infty}\}$$

In trying to prove for example the goal ($\mathbf{stream}\, x$), a goal-directed proof search may try to find a substitution for x that will make ($\mathbf{stream}\, x$) valid relative to the coinductive model of this set of clauses. This search by resolution may proceed by means of an infinite reduction $\underline{\mathbf{stream}\, x} \overset{\kappa_{\mathbf{stream}}:[\mathrm{scons}\ y\ x'/x]}{\rightsquigarrow}$ $\mathbf{nat}\, y \wedge \mathbf{stream}\, x' \overset{\kappa_{\mathbf{nat0}}:[0/y]}{\rightsquigarrow} \underline{\mathbf{stream}\, x'} \overset{\kappa_{\mathbf{stream}}:[\mathrm{scons}\ y'\ x''/x']}{\rightsquigarrow} \cdots$, thereby generating a stream Z of zeros via composition of the computed substitutions: $Z = (\mathrm{scons}\, 0\, x')[\mathrm{scons}\, 0\, x''/x'] \cdots$. Above, we annotated each resolution step with the label of the clause it resolves against and the computed substitution. A method to compute an answer for this infinite sequence of reductions was given by Gupta et al. [36] and Simon et al. [61]: the underlined loop gives rise to the circular unifier $x = \mathrm{scons}\, 0\, x$ that corresponds to the infinite term Z. It is proven that, if a loop and a corresponding circular unifier are detected, they provide an answer that is sound relative to the greatest complete Herbrand model of the clauses. This approach is known under the name of CoLP.

Horn Clause Theories in Type Inference Below clauses give the typing rules of the simply typed λ-calculus, and may be used for type inference or type checking:

$$\kappa_{t1} : \forall x\, \Gamma\, a.\, \mathbf{var}\, x \wedge \mathbf{find}\, \Gamma\, x\, a \;\to \mathbf{typed}\, \Gamma\, x\, a$$
$$\kappa_{t2} : \forall x\, \Gamma\, a\, m\, b.\, \mathbf{typed}\, [x : a|\Gamma]\, m\, b \to \mathbf{typed}\, \Gamma\, (\lambda\, x\, m)\, (a \to b)$$
$$\kappa_{t3} : \forall \Gamma\, a\, m\, n\, b.\, \mathbf{typed}\, \Gamma\, m\, (a \to b) \wedge \mathbf{typed}\, \Gamma\, n\, a \to \mathbf{typed}\, \Gamma\, (\mathrm{app}\, m\, n)\, b$$

It is well known that the Y-combinator is not typable in the simply-typed λ-calculus and, in particular, self-application $\lambda x.\, x\, x$ is not typable either. However, by switching off the occurs-check in Prolog or by allowing circular unifiers in CoLP [36,61], we can resolve the goal "$\mathbf{typed}\, []\, (\lambda\, x\, (\mathrm{app}\, x\, x))\, a$" and would compute the circular substitution: $a = b \to c, b = b \to c$ suggesting that an infinite, or circular, type may be able to type this λ-term. A similar trick would provide a typing for the Y-combinator. Thus, a coinductive interpretation of the above Horn clauses yields a theory of infinite types, while an inductive interpretation corresponds to the standard type system of the simply typed λ-calculus.

Horn Clause Theories in Type Class Inference Haskell type class inference does not require circular unifiers but may require a cyclic resolution inference [45,32]. Consider, for example, the following mutually defined data structures in Haskell.

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

This type declaration gives rise to the following equality class instance declarations, where we leave the, here irrelevant, body out.

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

The above two type class instance declarations have the shape of Horn clauses. Since the two declarations mutually refer to each other, an instance inference for, e.g., **Eq** (OddList **Int**) will give rise to an infinite resolution that alternates between the subgoals **Eq** (OddList **Int**) and **Eq** (EvenList **Int**). The solution is to terminate the computation as soon as the cycle is detected [45], and this method has been shown sound relative to the greatest Herbrand models in [31]. We will demonstrate this later in the proof systems proposed in this paper.

The diversity of these coinductive examples in the existing literature shows that there is a practical demand for coinductive methods in Horn clause logic, but it also shows that no unifying proof-theoretic approach exists to allow for a generic use of these methods. This causes several problems.

Problem 1. The existing proof-theoretic coinductive interpretations of cycle and loop detection are unclear, incomplete and not uniform.

To see this, consider Tab. 1, that exemplifies three kinds of circular phenomena in Horn clauses: The clause γ_1 is the easiest case – its coinductive models are given by a finite set $\{p\, a\}$. On the other extreme is the clause γ_3 that, just like our example of $\kappa_{\mathbf{stream}}$, admits only an infinite formula in its coinductive model. The intermediate case is γ_2 that could be interpreted by an infinite set of finite

Horn clauses	$\gamma_1 : \forall x.\, p\,x \to p\,x$	$\gamma_2 : \forall x.\, p(f\,x) \to p\,x$	$\gamma_3 : \forall x.\, p\,x \to p(f\,x)$
Greatest Herbrand model:	$\{p\,a\}$	$\{p(a), p(f\,a), p(f(f\,a)), \ldots\}$	$\emptyset$
Greatest complete Herbrand model:	$\{p\,a\}$	$\{p(a), p(f\,a), p(f(f\,a), \ldots, p(f(f \ldots)\}$	$\{p(f(f \ldots)\}$
CoLP substitution for query $p\,a$	*id*	fails	fails
CoLP substitution for query $p\,x$	*id*	$x = f\,x$	$x = f\,x$

Table 1. Examples of greatest (complete) Herbrand models for Horn clauses γ_1, γ_2, γ_3. The signatures are $\{a\}$ for the clause γ_1 and $\{a, f\}$ for the others.

formulae in its greatest Herbrand model, or may admit an infinite formula in its greatest complete Herbrand model. Examples like γ_1 appear in Haskell type class resolution [45], and examples like γ_2 – in its experimental extensions [32]. Cycle detection would only cover computations for γ_1, whereas γ_2, γ_3 require some form of loop detection[3]. However, CoLP's loop detection gives confusing results here. It correctly fails to infer $p\,a$ from γ_3 (no unifier for subgoals $p\,a$ and $p\,(f\,a)$ exists), but incorrectly fails to infer $p\,a$ from γ_2 (also failing to unify $p\,a$ and $p\,(f\,a)$). The latter failure is misleading bearing in mind that $p\,a$ is in fact in the coinductive model of γ_2. And vice versa, if we interpret the CoLP answer $x = f\,x$ as a declaration that an infinite term $(f\,f\, \ldots)$ is in the model, then CoLP's answer for γ_3 and $p\,x$ is exactly correct, however the same answer is badly incomplete for the query involving $p\,x$ and γ_2, because γ_2 in fact admits other, finite, formulae in its models. And in some applications, e.g. in Haskell type class inference, a finite formula would be the only acceptable answer for any query to γ_2.

This set of examples shows that loop detection is too coarse a tool to give an operational semantics to a diversity of coinductive models.

Problem 2. Constructive interpretation of coinductive proofs in Horn clause logic is unclear. Horn clause logic is known to be a constructive fragment of FOL. Some applications of Horn clauses rely on this property in a crucial way. For example, inference in Haskell type class resolution is constructive: when a certain formula F is inferred, the Haskell compiler in fact constructs a proof term that inhabits F seen as type. In our earlier example **Eq** (OddList **Int**) of the Haskell type classs, Haskell in fact captures the cycle by a fixpoint term t and proves that t inhabits the type **Eq** (OddList **Int**). Although we know from [31] that these computations are sound relative to greatest Herbrand models of Horn clauses, the results of [31] do not extend to Horn clauses like γ_3 or $\kappa_{\mathbf{stream}}$, or generally to Horn clauses modelled by the greatest *complete* Herbrand models. This shows that there is not just a need for coinductive proofs in Horn clause logic, but *constructive* coinductive proofs.

Problem 3. Incompleteness of circular unification for irregular coinductive data structures. Table 1 already showed some issues with incomplete-

[3] We follow the standard terminology of [65] and say that two formulae F and G form a cycle if $F = G$, and a loop if $F[\theta] = G[\theta]$ for some (possibly circular) unifier θ.

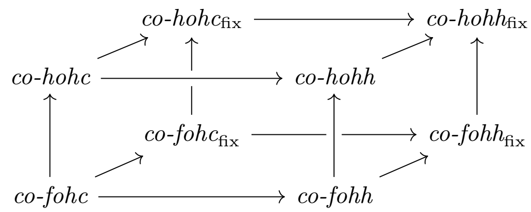


Fig. 1. Cube of logics covered by CUP

ness of circular unification. A more famous consequence of it is the failure of circular unification to capture irregular terms. This is illustrated by the following Horn clause, which defines the infinite stream of successive natural numbers.

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

The reductions for $\mathbf{from}\, 0\, y$ consist only of irregular (non-unifiable) formulae:

$$\mathbf{from}\ 0\ y \overset{\kappa_{\mathbf{from}}:[\mathrm{scons}\ \ 0\ \ y'/y]}{\leadsto} \mathbf{from}\ (s\, 0)\ y' \overset{\kappa_{\mathbf{from}}:[\mathrm{scons}\ \ (s\, 0)\ \ y''/y']}{\leadsto} \ldots$$

The composition of the computed substitutions would suggest an infinite term as answer: $\mathbf{from}\, 0\, (\mathrm{scons}\ 0\ (\mathrm{scons}\ (s\, 0)\ \ \ldots))$. However, circular unification no longer helps to compute this answer, and CoLP fails. Thus, there is a need for more general operational semantics that allows irregular coinductive structures.

A New Theory of Coinductive Proof Search in Horn Clause Logic

In this paper, we aim to give a principled and *general* theory that resolves the three problems above. This theory establishes a *constructive* foundation for coinductive resolution and allows us to give proof-theoretic characterisations of the approaches that have been proposed throughout the literature.

To solve Problem 1, we follow the footsteps of the *uniform proofs* by Miller and Nadathur [47], who gave a general proof-theoretic account of resolution in first-order Horn clause logic (*fohc*) and three extensions: first-order hereditary Harrop clauses (*fohh*), higher-order Horn clauses (*hohc*), and higher-order hereditary Harrop clauses (*hohh*). In Sec. 3, we extend uniform proofs with a general coinduction proof principle. The resulting framework is called *coinductive uniform proofs (CUP)*. We show how the resulting coinductive extensions of the four logics of Miller and Nadathur, which we name *co-fohc*, *co-fohh*, *co-hohc* and *co-hohh*, in fact give a precise proof-theoretic characterisation to the different kinds of coinduction described in the literature. For example, coinductive proofs involving the clauses γ_1 and γ_2 belong to *co-fohc* and *co-fohh*, respectively. However, proofs involving clauses like γ_3 or $\kappa_{\mathbf{stream}}$ require in addition an explicit use of fixed point terms expressing the infinite data structures. These extentions are denoted by *co-fohc*$_{\mathrm{fix}}$, *co-fohh*$_{\mathrm{fix}}$, *co-hohc*$_{\mathrm{fix}}$ and *co-hohh*$_{\mathrm{fix}}$.

Sec. 3 shows that this yields the cube in Fig. 1, where the arrows show the increase in logical strength. The invariant search for regular infinite objects done in CoLP is fully described by the logic *co-fohc*$_{\mathrm{fix}}$, including proofs for clauses like γ_3 and $\kappa_{\mathbf{stream}}$. An important consequence is that CUP is complete for γ_1, γ_2, and γ_3, e.g. $p\,a$ is provable from γ_2 in CUP, but not in CoLP.

In tackling Problem 3, we will find that the irregular proofs, such as those for $\kappa_{\mathbf{from}}$, can be given in *co-hohh*$_{\mathrm{fix}}$. The stream of successive numbers can be defined as a higher-order fixed point term $s_{\mathrm{fr}} = \mathrm{fix}\, f.\, \lambda x.\, \mathrm{scons}\, x\, (f\, (s\, x))$, and the proposition $\forall x.\, \mathbf{from}\, x\, (s_{\mathrm{fr}}\, x)$ is provable in *co-hohh*$_{\mathrm{fix}}$. This requires the use of higher-order syntax, fixed point terms and the goals of universal shape, which become available in the syntax of Hereditary Harrop logic.

In order to solve Problem 2 and to expose the constructive nature of the resulting proof systems, we present in Sec. 4 a coinductive extension of first-order intuitionistic logic and its sequent calculus. This extension ($\mathbf{iFOL}_{\blacktriangleright}$) is based on the so-called later modality (or Löb modality) known from provability logic [15,62], type theory [51,8] and domain theory [18]. However, our way of using the later modality to control recursion in first-order proofs is new and has only been proposed before in [13,14]. In the same section we also show that CUP is sound relative to $\mathbf{iFOL}_{\blacktriangleright}$, which gives us a handle on the constructive content of CUP. This yields, among other consequences, a constructive interpretation of CoLP proofs.

Section 5 is dedicated to showing soundness of both coinductive proof systems relative to *complete Herbrand models* [46]. The construction of these models is carried out by using coalgebras and category theory. This frees us from having to use topological methods and will simplify future extensions of the theory to, e.g., encompass typed logic programming. It also makes it possible to give original and constructive proofs of soundness for both CUP and $\mathbf{iFOL}_{\blacktriangleright}$ in Section 5. We finish the paper with discussion of related and future work.

Originality of the contribution

The results of this paper give a comprehensive characterisation of coinductive Horn clause theories from the point of view of proof search (by expressing coinductive proof search and resolution as coinductive uniform proofs), constructive proof theory (via a translation into a intuitionsitic sequent calculus), and coalgebraic semantics (via coinductive Herbrand models and constructive soundness results). Several of the presented results have never appeared before: the coinductive extension of uniform proofs; characterisation of coinductive properties of Horn clause theories in higher-order logic with and without fixed point operators; coalgebraic and fibrational view on complete Herbrand models; and soundness of an intuitionsitic logic with later modality relative to complete Herbrand models.

2 Preliminaries: Terms and Formulae

In this section, we set up notation and terminology for the rest of the paper. Most of it is standard, and blends together the notation used in [47] and [11].

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

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

Fig. 2. Well-Formed Terms

$$\frac{(p : \tau_1 \to \cdots \to \tau_n \to o) \in \Pi \qquad \Gamma \vdash M_1 : \tau_1 \qquad \cdots \qquad \Gamma \vdash M_n : \tau_n}{\Gamma \Vdash p\, M_1 \cdots\, M_n}$$

$$\frac{}{\Gamma \Vdash \top} \qquad \frac{\Gamma \Vdash \varphi \qquad \Gamma \Vdash \psi \qquad \Box \in \{\wedge, \vee, \to\}}{\Gamma \Vdash \varphi \,\Box\, \psi} \qquad \frac{\Gamma, x : \tau \Vdash \varphi}{\Gamma \Vdash \forall x : \tau.\, \varphi} \qquad \frac{\Gamma, x : \tau \Vdash \varphi}{\Gamma \Vdash \exists x : \tau.\, \varphi}$$

Fig. 3. Well-formed Formulae

Definition 1. We define the set $\mathbb{T}$ of *(simple) types* and the set $\mathbb{P}$ of *proposition types* to be given by the following grammars, where ι is the *base type*.

$$\mathbb{T} \ni \sigma, \tau ::= \iota \mid \sigma \to \tau \qquad\qquad \mathbb{P} \ni \rho ::= o \mid \sigma \to \rho, \quad \sigma \in \mathbb{T}$$

We adapt the usual convention that $\to$ binds to the right.

Definition 2. A *term signature* Σ is a set of pairs $c : \tau$, where $\tau \in \mathbb{T}$, and a *predicate signature* is a set Π of pairs $p : \rho$ with $\rho \in \mathbb{P}$. The elements in Σ and Π are called *term symbols* and *predicate symbols*, respectively. Given term and predicate signatures Σ and Π, we refer to the pair (Σ, Π) as *signature*. Let Var be a countable set of variables, the elements of which we denote by $x, y, \ldots$ We call a list Γ of pairs $x : \tau$ of variables and types a *context*. The set Λ_Σ of *(well-typed) terms* over Σ is the collection of all M with $\Gamma \vdash M : \tau$ for some context Γ and type $\tau \in \mathbb{T}$, where $\Gamma \vdash M : \tau$ is defined inductively in Fig. 2. A term is called *closed* if $\vdash M : \tau$, otherwise it is called *open*. Finally, we let Λ_Σ^- denote the set of all terms M that do not involve fix.

Definition 3. Let (Σ, Π) be a signature. We say that φ is a *(well-formed) formula* in context Γ, if $\Gamma \Vdash \varphi$ is inductively derivable from the rules in Fig. 3.

Definition 4. The *reduction relation* $\longrightarrow$ on terms in Λ_Σ is given as the compatible closure (reduction under applications and binders) of β- and fix-reduction:

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

Two terms M and N are called *convertible*, if $M \equiv N$, where $\equiv$ is the equivalence closure of $\longrightarrow$. Conversion of terms extends from terms to formulae by taking the compatible closure of conversion under predicate symbols: if $M_k \equiv M'_k$ for $k = 1, \ldots, n$, then $p\, M_1 \cdots M_n \equiv p\, M'_1 \cdots\, M'_n$.

We will use in the following that the above calculus features subject reduction and confluence, cf. [54]: if $\Gamma \vdash M : \tau$ and $M \equiv N$, then $\Gamma \vdash N : \tau$; and $M \equiv N$ iff there is a term P, s.t. M and N reduce in finitely many steps to P.

The *order* of a type $\tau \in \mathbb{T}$ is given as usual by $\mathrm{ord}(\iota) = 0$ and $\mathrm{ord}(\sigma \to \tau) = \max\{\mathrm{ord}(\sigma) + 1, \mathrm{ord}(\tau)\}$. If $\mathrm{ord}(\tau) \leq 1$, then the arity of τ is given by $\mathrm{ar}(\iota) = 0$ and $\mathrm{ar}(\iota \to \tau) = \mathrm{ar}(\tau) + 1$. A signature Σ is called *first-order*, if for all $f : \tau \in \Sigma$ we have $\mathrm{ord}(\tau) \leq 1$. We let the arity of f then be $\mathrm{ar}(\tau)$ and denote it by $\mathrm{ar}(f)$.

Definition 5. The set of *guarded base terms* over a first-order signature Σ is given by the following type-driven rules.

$$\frac{x : \tau \in \Gamma \qquad \mathrm{ord}(\tau) \leq 1}{\Gamma \vdash_g x : \tau} \quad \frac{f : \tau \in \Sigma}{\Gamma \vdash_g f : \tau} \quad \frac{\Gamma \vdash_g M : \sigma \to \tau \qquad \Gamma \vdash_g N : \sigma}{\Gamma \vdash_g M\, N : \tau}$$

$$\frac{f \in \Sigma \qquad \mathrm{ord}\, \tau \leq 1 \qquad \Gamma, x : \tau, y_1 : \iota, \ldots, y_{\mathrm{ar}(\tau)} : \iota \vdash_g M_i : \iota \qquad 1 \leq i \leq \mathrm{ar}(f)}{\Gamma \vdash_g \mathrm{fix}\, x.\, \lambda \overrightarrow{y}.\, f\, \overrightarrow{M} : \tau}$$

General *guarded terms* are terms M, such that all fix-subterms are guarded base terms, which means that they are generated by the following grammar.

$$G ::= M \text{ (with } \vdash_g M : \tau \text{ for some type } \tau) \mid c \in \Sigma \mid x \in \mathrm{Var} \mid G\, G \mid \lambda x.\, G$$

Finally, M is a *first-order* term over Σ with $\Gamma \vdash M : \tau$ if $\mathrm{ord}(\tau) \leq 1$ and the types of all variables occurring in Γ are of order 0. We denote the set of guarded first-order terms M with $\Gamma \vdash M : \iota$ by $\Lambda_{\Sigma}^{G,1}(\Gamma)$ and the set of guarded terms in Γ by $\Lambda_{\Sigma}^{G}(\Gamma)$. If Γ is empty, we just write $\Lambda_{\Sigma}^{G,1}$ and Λ_{Σ}^{G}, respectively.

Note that an important aspect of guarded terms is that no free variables occur under a fix-operator. *Guarded base terms* should be seen as a specific form of fixed point terms that will allow us to unfold them into potentially infinite trees later, while *guarded terms* close guarded base terms under the constructions of the simply typed λ-calculus.

Example 6. Let us provide a few examples that illustrate (first-order) guarded terms. We use the first-order signature $\Sigma = \{\mathrm{scons} \colon \iota \to \iota \to \iota, s \colon \iota \to \iota, 0 : \iota\}$.

1. Let $s_{\mathrm{fr}} = \mathrm{fix}\, f.\, \lambda x.\, \mathrm{scons}\ x\ (f\ (s\ x))$ be the function that computes the streams of numerals starting at the given argument. It is easy to show that $\vdash_g s_{\mathrm{fr}} : \iota \to \iota$ and so $s_{\mathrm{fr}}\ 0 \in \Lambda_{\Sigma}^{G,1}$.
2. For the same signature Σ we also have $x : \iota \vdash_g x : \iota$. Thus $x \in \Lambda_{\Sigma}^{G,1}(x : \iota)$ and $s\ x \in \Lambda_{\Sigma}^{G,1}(x : \iota)$.
3. We have $x : \iota \to \iota \vdash_g x\, 0 : \iota$, but $(x\ 0) \notin \Lambda_{\Sigma}^{G,1}(x : \iota \to \iota)$.

The purpose of guarded terms is that these are productive, that is, we can reduce them to a term that either has a function symbol at the root or is just a variable. In other words, guarded terms have head normal forms: We say that a term M is in *head normal form*, if $M = f\ \overrightarrow{N}$ for some $f \in \Sigma$ or if $M = x$ for some variable x. The following lemma is a technical result that is needed to show in Lem. 8 that all guarded terms have a head normal form.

Lemma 7. *Let M and N be guarded base terms with $\Gamma, x : \sigma \vdash_g M : \tau$ and $\Gamma \vdash_g N : \sigma$. Then $M[N/x]$ is a guarded base term with $\Gamma \vdash_g M[N/x] : \tau$.*

Lemma 8. *If M is a first-order guarded term with $M \in \Lambda^{G,1}_{\Sigma}(\Gamma)$, then M reduces to a unique head normal form. This means that either (i) there is either a unique $f \in \Sigma$ and terms $N_1, \ldots, N_{\mathrm{ar}(f)}$ with $\Gamma \vdash_g N_k : \iota$ and $M \longrightarrow\!\!\!\!\rightarrow f\,\overrightarrow{N}$, and moreover if $M \longrightarrow\!\!\!\!\rightarrow f\,\overrightarrow{L}$, then $\overrightarrow{N} \equiv \overrightarrow{L}$; or (ii) $M \longrightarrow\!\!\!\!\rightarrow x$ for some $x : \iota \in \Gamma$.*

We end this section by introducing the notion of an atom and refinements thereof. This will enable us to define the different logics and thereby to analyse the strength of coinduction hypotheses, which we promised in the introduction.

Definition 9. A formula φ of the shape $\top$ or $p\ M_1 \cdots\ M_n$ is an *atom* and a

- *first-order atom*, if p and all the terms M_i are first-order;
- *guarded atom*, if all term M_i are guarded; and
- *simple atom*, if all terms M_i are non-recursive, that is, are in Λ^{-}_{Σ}.

First-order, guarded and simple atoms are denoted by At_1, At^g and At^s_{ω}. We denote conjunctions of these predicates by $\mathrm{At}^g_1 = \mathrm{At}_1 \cap \mathrm{At}^g$ and $\mathrm{At}^s_1 = \mathrm{At}_1 \cap \mathrm{At}^s_{\omega}$.

Note that the restriction for At^g only applies to fixed point terms. Hence, any formula that contains terms without fix is already in At^g and $\mathrm{At}^g \cap \mathrm{At}^s_{\omega} = \mathrm{At}^s_{\omega}$. Since these notions are rather subtle, we give a few examples

Example 10. We list three examples of first-order variants atoms.

1. For $x : \iota$ we have **stream** $x \in \mathrm{At}_1$, but there are also "garbage" formulae like "**stream** $(\mathrm{fix}\ x.\, x)$" in At_1. Examples of atoms that are not first-order are $p\ M$, where $p : (\iota \to \iota) \to o$ or $x : \iota \to \iota \vdash M : \tau$.
2. Our running example "**from** $0\ (s_{\mathrm{fr}}\ 0)$" is a first-order guarded atom in At^g_1.
3. The formulae in At^s_1 may not contain recursion and higher-order features. However, the atoms of Horn clauses in a logic program fit in here.

3 Coinductive Uniform Proofs

This section introduces the eight logics of the coinductive uniform proof framework announced and motivated in the introduction. The major difference of uniform proofs with, say, a sequent calculus is the "uniformity" property, which means that the choice of the application of each proof rule is deterministic and all proofs are in normal form (cut free). This models the operational semantics of resolution, in which the proof search is always goal directed. Hence, the main challenge, that we set out to solve in this section, is to extend the uniform proof framework with coinduction, while preserving this valuable operational property.

We begin by introducing the different goal formulae and definite clauses that determine the logics that were presented in the cube for coinductive uniform proofs in the introduction. These clauses and formulae correspond directly to those of the original work on uniform proofs [47] with the only difference being

	Definite Clauses	Goals
co-fohc	D_1	$G ::= \mathrm{At}_1^s \mid G \wedge G \mid G \vee G \mid \exists x : \tau.\, G$
co-hohc	D_ω	$G ::= \mathrm{At}_\omega^s \mid G \wedge G \mid G \vee G \mid \exists x : \tau.\, G$
co-fohh	D_1	$G ::= \mathrm{At}_1^s \mid G \wedge G \mid G \vee G \mid \exists x : \tau.\, G \mid D \to G \mid \forall x : \tau.\, G$
co-hohh	D_ω	$G ::= \mathrm{At}_\omega^s \mid G \wedge G \mid G \vee G \mid \exists x : \tau.\, G \mid D \to G \mid \forall x : \tau.\, G$

Table 2. D- and G-formulae for coinductive uniform proofs.

that we need to distinguish atoms with and without fixed point terms. The general idea is that goal formulae (G-formulae) occur on the right of a sequent, thus are the *goal* to be proved. Definite clauses (D-formulae), on the other hand, are selected from the context as assumptions. This will become clear once we introduce the proof system for coinductive uniform proofs.

Definition 11. Let D_i be generated by the following grammar with $i \in \{1, \omega\}$.

$$D_i ::= \mathrm{At}_i^s \mid G \to D \mid D \wedge D \mid \forall x : \tau.\, D$$

The sets of definite clauses (D-formulae) and goals (G-formulae) of the four logics *co-fohc*, *co-fohh*, *co-hohc*, *co-hohh* are the well-formed formulae of the corresponding shapes defined in Tab. 2. For the variations *co-fohh*$_{\text{fix}}$ etc. of these logics that allow fixed point terms, we replace upper index "b" with "g" everywhere in Tab. 2. A D-formula of the shape $\forall \vec{x}.\, A_1 \wedge \cdots \wedge A_n \to A_0$ is called *H-formula* or *Horn clause* if $A_k \in \mathrm{At}_1^s$, and *H^g-formula* if $A_k \in \mathrm{At}_1^g$. Finally, a *logic program* (or *program*) P is a set of H-formulae.

Note that any set of D-formulae in *fohc* can be transformed into an intuitionistically equivalent set of H-formulae, as was shown in [47].

We are now ready to introduce the rules. The first block of them, given in Fig. 4 simply restates the usual uniform proof rules of Miller and Nadathur [47]. Of special notice is the rule DECIDE that mimics the operational behaviour of resolution in logic programming, choosing to resolve the goal against a clause D contained in the given program. The second block, given in Fig. 5, introduces the coinduction rule (CO-FIX). This rule mimics the typical guard condition that one finds in coinductive programs and proofs [27,35,5,8,17]. One can read the rule as follows: to prove a formula φ coinductively, we assume φ as a coinduction hypothesis and derive φ in a guarded way. This guardedness condition is formalised by applying the guarding modality $\langle \_ \rangle$ on the formula being proven by coinduction and the proof rules that allow us to distribute the guard over certain logical connectives, see Fig. 5. The guarding modality may be discharged only if the guarded goal was resolved against a clause in the initial program or any hypothesis, except for the coinduction hypotheses. This is reflected in the rule DECIDE$\langle\rangle$, where we may only pick a clause from P, and is in contrast to the rule DECIDE, in which we can pick *any* hypothesis. The proof may only terminate with the INITIAL step if the goal is no longer guarded.

Note that the CO-FIX rule introduces a goal as a new hypothesis. Hence, we have to require that this goal is also a definite clause. Since coinduction hypotheses play such an important role, they deserve a separate definition.

$$\frac{\Sigma; P; \Delta \overset{D}{\Rightarrow} A \quad D \in P \cup \Delta}{\Sigma; P; \Delta \Longrightarrow A}\ \textsc{decide} \qquad \frac{A \equiv A'}{\Sigma; P; \Delta \overset{A'}{\Rightarrow} A}\ \textsc{initial} \qquad \frac{}{\Sigma; P; \Delta \Longrightarrow \top}\ \top R$$

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

$$\frac{\Sigma; P; \Delta \overset{D_x}{\Longrightarrow} A \qquad x \in \{1, 2\}}{\Sigma; P; \Delta \xrightarrow{D_1 \wedge D_2} A}\ \wedge L \qquad \frac{\Sigma; P; \Delta \Longrightarrow G_1 \qquad \Sigma; P; \Delta \Longrightarrow G_2}{\Sigma; P; \Delta \Longrightarrow G_1 \wedge G_2}\ \wedge R$$

$$\frac{\Sigma; P; \Delta \xrightarrow{D[N/x]} A \qquad \varnothing \vdash_g N : \tau}{\Sigma; P; \Delta \xrightarrow{\forall x.\, D} A}\ \forall L \qquad \frac{c : \tau, \Sigma; P; \Delta \Longrightarrow G\,[c/x] \qquad c : \tau \notin \Sigma}{\Sigma; P; \Delta \Longrightarrow \forall x : \tau.\, G}\ \forall R$$

$$\frac{\Sigma; P; \Delta \Longrightarrow G\,[N/x] \qquad \varnothing \vdash_g N : \tau}{\Sigma; P; \Delta \Longrightarrow \exists x : \tau.\, G}\ \exists R \qquad \frac{\Sigma; P; \Delta \Longrightarrow G_x \qquad x \in \{1, 2\}}{\Sigma; P; \Delta \Longrightarrow G_1 \vee G_2}\ \vee R$$

Fig. 4. Uniform Proof Rules.

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

$$\frac{\Sigma; P; \Delta \overset{D}{\Rightarrow} A \qquad D \in P}{\Sigma; P; \Delta \Longrightarrow \langle A \rangle}\ \textsc{decide}\langle\rangle \qquad \frac{c : \tau, \Sigma; P; \Delta \Longrightarrow \langle \varphi\,[c/x] \rangle \qquad c : \tau \notin \Sigma}{\Sigma; P; \Delta \Longrightarrow \langle \forall x : \tau.\, \varphi \rangle}\ \forall R\langle\rangle$$

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

Fig. 5. Coinductive Uniform Proof Rules

Definition 12. Given a language L from Tab. 2, a formula φ is a *coinduction goal* of L if φ simultaneously is a D- and a G-formula of L.

Note that the coinduction goals of *co-fohc* and *co-fohh* can be transformed into equivalent H- or H^g-formulae, since any coinduction goal is a D-formula.

Let us now formally introduce the coinductive uniform proof system.

Definition 13. Let P and Δ be finite sets of, respectively, definite clauses and coinduction goals, over the signature Σ, and suppose that G is a goal and φ is a coinduction goal. A *sequent* is either a *uniform provability sequent* of the form $\Sigma; P; \Delta \Longrightarrow G$ or $\Sigma; P; \Delta \overset{D}{\Rightarrow} A$ as defined in Fig. 4, or it is a *coinductive uniform provability sequent* of the form $\Sigma; P \looparrowright \varphi$ as defined in Fig. 5. Let L be a language from Tab. 2. We say that φ is *coinductively provable* in L, if P is a set of D-formulae in L, φ is a coinduction goal in L and $\Sigma; P \looparrowright \varphi$ holds.

The logics we have introduced impose different syntactic restrictions on D- and G-formulae, and will therefore admit coinductive goals of different strength. This ability to explicitly use stronger coinductive hypotheses within a goal-directed search was missing in CoLP, for example. And it allows us to account for different coinductive properties of Horn clauses as described in the introduction. We finish this section by illustrating this strengthening.

The first example is one for the logic *co-fohc*, in which we illustrate the framework on the problem of type class resolution.

Example 14. Let us restate the Haskell type class inference problem discussed in the introduction in terms of Horn clauses:

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

To prove $\mathbf{eq}\ (\mathrm{odd}\ \mathrm{i})$ for this set of Horn clauses, it is sufficient to use this formula directly as coinduction hypothesis, as shown in Fig. 6. Note that this formula is indeed a coinduction goal of *co-fohc*, hence we find ourselves in the simplest scenario of coinductive proof search. In Tab. 1, γ_1 is a representative for this kind of coinductive proofs with simplest atomic goals.

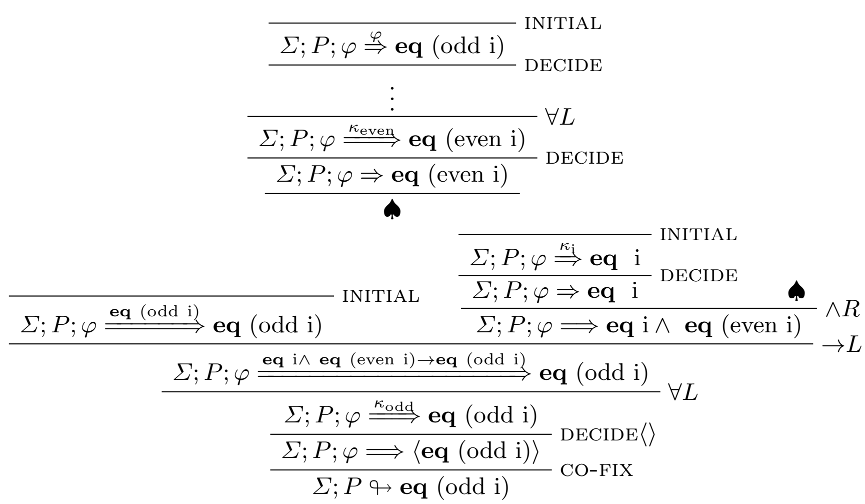


Fig. 6. The *co-fohc* proof for Horn clauses arising from Haskell Type class examples. φ abbreviates the coinduction hypothesis $\mathbf{eq}\ (\mathrm{odd}\ \mathrm{i})$. Note its use in the branch ♠.

It was pointed out in [32] that Haskell's type class inference can also give rise to irregular corecursion. Such cases may require the more general coinduction hypothesis (e.g. universal and/or implicative) of *co-fohh* or *co-hohh*. The below set of Horn clauses is a simplified representation of a problem given in [32]:

$$\begin{aligned}\kappa_{\mathrm{i}} &: \mathbf{eq}\ \mathrm{i}\\ \kappa_{s} &: \forall x.\, (\mathbf{eq}\ x) \wedge \mathbf{eq}\ (s\ (g\ x)) \to \mathbf{eq}\ (s\ x)\\ \kappa_{g} &: \forall x.\, \mathbf{eq}\ x \qquad\qquad\qquad\quad\ \to \mathbf{eq}\ (g\ x)\end{aligned}$$

Trying to prove $\mathbf{eq}\ (s\ \mathrm{i})$ by using $\mathbf{eq}\ (s\ \mathrm{i})$ directly as a coinduction hypothesis is deemed to fail, as the coinductive proof search is irregular and this coinduction hypothesis would not be applicable in any guarded context. But it is possible to prove **eq** $(s\ \mathrm{i})$ as a corollary of another theorem: $\forall x.\,(\mathbf{eq}\ x) \rightarrow \mathbf{eq}\ (s\ x)$. Using this formula as coinduction hypothesis leads to a successful proof, which we omit here. From this more general goal, we can derive the original goal by instantiating the quantifier with i and eliminating the implication with κ_{i}. This second derivation is sound with respect to the models, as we show in Thm. 34.

We encounter γ_2 from Tab. 1 in a similar situation: To prove $p\,a$, we first have to prove $\forall x.\,p\ x$ in *co-fohh*, and then obtain $p\ a$ as a corollary by appealing to Thm. 34. The next example shows that we can cover all cases in Tab. 1 by providing a proof in *co-hohh*$_{\mathrm{fix}}$ that involves irregular recursive terms.

Example 15. Recall the clause $\forall x\ y.\,\mathbf{from}\ (s\ x)\ y \rightarrow \mathbf{from}\ x\ (\mathrm{scons}\ x\ y)$ that we named $\kappa_{\mathbf{from0}}$ in the introduction. Proving $\exists y.\,\mathbf{from}\ 0\ y$ is again not possible directly. Instead, we can use the term $s_{\mathrm{fr}} = \mathrm{fix}\,f.\,\lambda x.\,\mathrm{scons}\ x\ (f\ (s\ x))$ from Ex. 6 and prove $\forall x.\,\mathbf{from}\ x\ (s_{\mathrm{fr}}\ x)$ coinductively, as shown in Fig. 10. This formula gives a coinduction hypothesis of sufficient generality. Note that the correct coinduction hypothesis now requires the fixed point definition of an infinite stream of successive numbers and universal quantification in the goal. Hence the need for the richer language of *co-hohh*$_{\mathrm{fix}}$. From this more general goal we can derive our initial goal $\exists\, y.\mathbf{from}\ 0\ y$ by instantiating y with $s_{\mathrm{fr}}\ 0$.

$$\spadesuit := \dfrac{\dfrac{\dfrac{}{c,\Sigma;P;\varphi \xRightarrow{\mathbf{from}\ (s\ c)\ (s_{\mathrm{fr}}\ (s\ c))} \mathbf{from}\ (s\ c)\ (s_{\mathrm{fr}}\ (s\ c))}\ \textsc{initial}}{c,\Sigma;P;\varphi \xRightarrow{\varphi} \mathbf{from}\ (s\ c)\ (s_{\mathrm{fr}}\ (s\ c))}\ \forall L}{c,\Sigma;P;\varphi \Longrightarrow \mathbf{from}\ (s\ c)\ (s_{\mathrm{fr}}\ (s\ c))}\ \textsc{decide}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{c,\Sigma;P;\varphi \xRightarrow{\mathbf{from}\ c\ (\mathrm{scons}\ c\ (s_{\mathrm{fr}}\ (s\ c)))} \mathbf{from}\ c\ (s_{\mathrm{fr}}\ c)}\ \textsc{initial} \qquad \spadesuit}{c,\Sigma;P;\varphi \xRightarrow{\mathbf{from}\ (s\ c)\ (s_{\mathrm{fr}}\ (s\ c)) \rightarrow \mathbf{from}\ c\ (\mathrm{scons}\ c\ (s_{\mathrm{fr}}\ (s\ c)))} \mathbf{from}\ c\ (s_{\mathrm{fr}}\ c)}\ {\rightarrow}L}{c,\Sigma;P;\varphi \xRightarrow{\kappa_{\mathbf{from0}}} \mathbf{from}\ c\ (s_{\mathrm{fr}}\ c)}\ \forall L\,(2\ \text{times})}{c,\Sigma;P;\varphi \Longrightarrow \langle \mathbf{from}\ c\ (s_{\mathrm{fr}}\ c)\rangle}\ \textsc{decide}\langle\rangle}{\Sigma;P;\varphi \Longrightarrow \langle \forall x.\,\mathbf{from}\ x\ (s_{\mathrm{fr}}\ x)\rangle}\ \forall R\langle\rangle}{\Sigma;P \looparrowright \forall x.\,\mathbf{from}\ x\ (s_{\mathrm{fr}}\ x)}\ \textsc{co-fix}$$

Fig. 7. The *co-hohh*$_{\mathrm{fix}}$ proof for $\varphi = \forall x.\,\mathbf{from}\ x\ (s_{\mathrm{fr}}\ x)$. Note that the last step of the leftmost branch involves $\mathbf{from}\ c\ (\mathrm{scons}\ c\ (s_{\mathrm{fr}}\ (s\ c))) \equiv \mathbf{from}\ c\ (s_{\mathrm{fr}}\ c)$.

There are examples of coinductive proofs that require a fixed point definition of an infinite stream, but do not require the syntax of higher-order terms or hereditary Harrop formulae. Such proofs can be performed in the *co-fohc*$_{\mathrm{fix}}$ logic. A good example is a proof that the stream of zeros satisfies the Horn clause theory defining the predicate **stream** in the introduction. The goal $(\mathbf{stream}\ s_0)$, with $s_0 = \mathrm{fix}\,x.\,\mathrm{scons}\ 0\ x$ can be proven directly by coinduction. Thus, CoLP's [36] loop detection with circular unification corresponds to coinductive proofs in the logic *co-fohc*$_{\text{fix}}$. Similarly, one can type self-application with the infinite type $a = \text{fix}\, t.\, t \to b$ for some given type b. The proof for **typed** $[x : a]$ (app x x) b is then in *co-fohc*$_{\text{fix}}$. Clause γ_3 is also in this group.

4 Coinductive Uniform Proofs and Intuitionistic Logic

In the last section, we introduced the framework of coinductive uniform proofs, which gives an operational account to proofs for coinductively interpreted logic programs. Having this framework at hand, we need to position it in the existing ecosystem of logical systems. The goal of this section is to prove that coinductive uniform proofs are in fact constructive. We show this by first introducing an extension of intuitionistic first-order logic that allows us to deal with recursive proofs for coinductive predicates. Afterwards, we show that coinductive uniform proofs are sound relative to this logic by means of a proof tree translation. The model-theoretic soundness proofs for both logics will be provided in Section 5.

We begin by introducing an extension of intuitionistic first-order logic with the so-called *later modality*, written $\blacktriangleright$. This modality is the essential ingredient that allows us to equip proofs with a controlled form of recursion. The later modality stems originally from provability logic, which characterises transitive, well-founded Kripke frames [63], and thus allows one to carry out induction without an explicit induction scheme [15]. Later, the later modality was picked up by the type-theoretic community to control recursion in coinductive programming [51,8,9,19,49], mostly with the intent to replace syntactic guardedness checks for coinductive definitions by type-based checks of well-definedness.

Formally, the logic $\mathbf{iFOL}_\blacktriangleright$ is given by the following definition.

Definition 16. The formulae of $\mathbf{iFOL}_\blacktriangleright$ are given by the rules in Def. 3 and the following one.

$$\frac{\Gamma \Vdash \varphi}{\Gamma \Vdash \blacktriangleright \varphi}$$

Conversion extends to these formulae in the obvious way. Let φ be a formula and Δ a sequence of formulae in $\mathbf{iFOL}_\blacktriangleright$. We say φ is *provable in context* Γ *under the assumptions* Δ in $\mathbf{iFOL}_\blacktriangleright$, if $\Gamma \mid \Delta \vdash \varphi$ holds. The *provability relation* $\vdash$ is thereby given inductively by the rules in the Figures 8 and 9.

The rules in Fig. 8 are the usual rules for intuitionistic first-order logic and should come at no surprise. More interesting are the rules in Fig. 9, where the rule **(Löb)** introduces recursion into the proof system. Furthermore, the rule **(Mon)** allows us to to distribute the later modality over implication, and consequently over conjunction and universal quantification. This is essential in the soundness result in Thm. 18 below. Finally, the rule **(Next)** gives us the possibility to proceed without any recursion, if necessary.

Note that so far it is not possible to use the assumption $\blacktriangleright \varphi$ introduced in the **(Löb)**-rule. The idea is that the formulae of a logic program provide us the obligations that we have to prove, possibly by recursion, in order to prove a coinductive predicate. This is cast in the following definition.

$$\frac{\Gamma \Vdash \Delta \qquad \varphi \in \Delta}{\Gamma \mid \Delta \vdash \varphi}\ \textbf{(Proj)} \qquad \frac{\Gamma \mid \Delta \vdash \varphi' \qquad \varphi \equiv \varphi'}{\Gamma \mid \Delta \vdash \varphi}\ \textbf{(Conv)} \qquad \frac{\Gamma \Vdash \Delta}{\Gamma \mid \Delta \vdash \top}\ (\top\textbf{-I})$$

$$\frac{\Gamma \mid \Delta \vdash \varphi \qquad \Gamma \mid \Delta \vdash \psi}{\Gamma \mid \Delta \vdash \varphi \wedge \psi}\ (\wedge\textbf{-I}) \qquad \frac{\Gamma \mid \Delta \vdash \varphi_1 \wedge \varphi_2 \qquad i \in \{1,2\}}{\Gamma \mid \Delta \vdash \varphi_i}\ (\wedge_i\textbf{-E})$$

$$\frac{\Gamma \mid \Delta \vdash \varphi_i \qquad \Gamma \Vdash \varphi_j \qquad j \neq i}{\Gamma \mid \Delta \vdash \varphi_1 \vee \varphi_2}\ (\vee_i\textbf{-I}) \qquad \frac{\Gamma \mid \Delta, \varphi_1 \vdash \psi \qquad \Gamma \mid \Delta, \varphi_2 \vdash \psi}{\Gamma \mid \Delta, \varphi_1 \vee \varphi_2 \vdash \psi}\ (\vee\textbf{-E})$$

$$\frac{\Gamma \mid \Delta, \varphi \vdash \psi}{\Gamma \mid \Delta \vdash \varphi \to \psi}\ (\to\textbf{-I}) \qquad \frac{\Gamma \mid \Delta \vdash \varphi \to \psi \qquad \Gamma \mid \Delta \vdash \varphi}{\Gamma \mid \Delta \vdash \psi}\ (\to\textbf{-E})$$

$$\frac{\Gamma, x : \tau \mid \Delta \vdash \varphi \qquad x \notin \Gamma}{\Gamma \mid \Delta \vdash \forall x : \tau.\, \varphi}\ (\forall\textbf{-I}) \qquad \frac{\Gamma \mid \Delta \vdash \forall x : \tau.\, \varphi \qquad M : \tau \in \Lambda^{G}_{\Sigma}(\Gamma)}{\Gamma \mid \Delta \vdash \varphi\,[M/x]}\ (\forall\textbf{-E})$$

$$\frac{M : \tau \in \Lambda^{G}_{\Sigma}(\Gamma) \quad \Gamma \mid \Delta \vdash \varphi\,[M/x]}{\Gamma \mid \Delta \vdash \exists x : \tau.\, \varphi}\ (\exists\textbf{-I}) \qquad \frac{\Gamma \Vdash \psi \quad \Gamma, x : \tau \mid \Delta, \varphi \vdash \psi \quad x \notin \Gamma}{\Gamma \mid \Delta, \exists x : \tau.\, \varphi \vdash \psi}\ (\exists\textbf{-E})$$

Fig. 8. Intuitionistic Rules for Standard Connectives

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

Fig. 9. Rules for the Later Modality

Definition 17. Given an H-formula φ of the shape $\forall \vec{x}.\,(A_1 \wedge \cdots \wedge A_n) \to \psi$, we define its *guarding* $\overline{\varphi}$ to be $\forall \vec{x}.\,(\blacktriangleright A_1 \wedge \cdots \wedge \blacktriangleright A_n) \to \psi$. For a logic program P, we define its guarding $\overline{P}$ by guarding each formula in P.

This translation of a logic program into formulae that admit recursion corresponds to the unfolding of a coinductive predicate, cf. [14]. We show now how to transform the tree of a coinductive uniform proof into a proof tree in $\mathbf{iFOL}_{\blacktriangleright}$, such that the recursion and guarding mechanisms in both logics match up.

Theorem 18. *If P is a logic program over a first-order signature Σ and the sequent $\Sigma; P \looparrowright \varphi$ is provable in co-hohh$_{\text{fix}}$, then $\overline{P} \vdash \varphi$ is provable in* $\mathbf{iFOL}_{\blacktriangleright}$.

To prove this theorem, one uses that each coinductive uniform proof tree starts with an initial tree that has an application of the CO-FIX-rule at the root and that eliminates the guard by using the rules in Fig. 5. At the leaves of this tree, one finds proof trees that proceed only by means of the rules in Fig. 4. The initial tree is then translated into a proof tree in $\mathbf{iFOL}_{\blacktriangleright}$ that starts with an application of the **(Löb)**-rule, which corresponds to the CO-FIX-rule, and that simultaneously transforms the coinduction hypothesis and applies introduction rules for conjunctions etc. This ensures that we can match the coinduction hypothesis with the guarded formulae of the program P.

The results of this section give an answer to the common discussion of the significance of imposing guarding modalities on the left (in $\mathbf{iFOL}_{\blacktriangleright}$ style) or the right (in CUP style) of the sequents in coinductive proof rules. CUP uses modality on the right to preserve proof uniformity (and thus normal form property), whereas $\mathbf{iFOL}_{\blacktriangleright}$ gives an account for any proof in sequent calculus. However, one style of coinductive proofs is convertible into another.

5 Herbrand Models and Soundness

In Sec. 4 we showed that coinductive uniform proofs are sound relative to the intuitionistic logic $\mathbf{iFOL}_{\blacktriangleright}$. This gives us a handle on the constructive nature of coinductive uniform proofs. Since $\mathbf{iFOL}_{\blacktriangleright}$ is a non-standard logic, we still need to provide semantics for that logic. We do this by interpreting in Sec. 5.4 the formulae of $\mathbf{iFOL}_{\blacktriangleright}$ over the well-known (complete) Herbrand models and prove the soundness of the accompanying proof system with respect to these models. Although we obtain soundness of coinductive uniform proofs over Herbrand models from this, this proof is indirect and does not give a lot of information about the models captured by the different calculi *co-fohc* etc. For this reason, we will give in Sec. 5.3 a direct soundness proof for coinductive uniform proofs. We also obtain coinduction invariants from this proof for each of the calculi, which allows us to describe their proof strength.

5.1 Coinductive Herbrand Models and Semantics of Terms

Before we come to the soundness proofs, we introduce in this section (complete) Herbrand models by using the terminology of final coalgebras. We then utilise this description to give operational and denotational semantics to guarded terms. These semantics show that guarded terms allow the description and computation of potentially infinite trees.

This approach has been proven very successful both in logic and programming [1,66,67]. We will only require very little category theoretical vocabulary and assume that the reader is familiar with the category **Set** of sets and functions, and functors, see for example [12,23,44]. The terminology of algebras and coalgebras [4,42,57,58] is given by the following definition.

Definition 19. A *coalgebra* for a functor $F\colon \mathbf{Set} \to \mathbf{Set}$ is a map $c\colon X \to FX$. Given coalgebras $d : Y \to FY$ and $c\colon X \to FX$, we say that a map $h : A \to B$ is a *homomorphism* if $Fh \circ d = c \circ h$. We call a coalgebra $c\colon X \to FX$ *final*, if for every coalgebra d there is a unique homomorphism h from d to c. We will refer to h as the *coinductive extension* of d.

The idea of (complete) Herbrand models is that a set of Horn clauses determines for each predicate symbol a set of potentially infinite terms. Such terms are nothing but (potentially infinite) trees, whose nodes are labelled by function symbols and whose branching is given by the arity of these function symbols. To be able to deal with open terms, we will also allow such trees to have leaves labelled by variables. Formally, the trees are given as final coalgebra for a functor determined by the signature.

Definition 20. Let Σ be first-order signature. The *extension* of a first-order signature Σ is a (polynomial) functor [33] $[\![\Sigma]\!] : \mathbf{Set} \to \mathbf{Set}$ given by

$$[\![\Sigma]\!](X) = \coprod_{f \in \Sigma} X^{\mathrm{ar}(f)},$$

where $\mathrm{ar}\colon \Sigma \to \mathbb{N}$ is defined in Sec. 2 and X^n is the n-fold product of X. We define for a set V a functor $[\![\Sigma]\!]+V \colon \mathbf{Set} \to \mathbf{Set}$ by $([\![\Sigma]\!]+V)(X) = [\![\Sigma]\!](X)+V$, where $+$ is the coproduct (disjoint union) in $\mathbf{Set}$.

For the following definition, note that we can view Π is a signature as well and thus obtain its extension $[\![\Pi]\!]$.

Definition 21. Let Σ be a first-order signature. The *coterms* over Σ are the final coalgebra $\mathrm{root}\colon \Sigma^\infty \to [\![\Sigma]\!](\Sigma^\infty(V)) + V$. For brevity, we denote $\Sigma^\infty(\emptyset)$ by Σ^∞, and call it the *(complete) Herbrand universe* and its elements *ground* coterms. Finally, we let the *(complete) Herbrand base* $\mathcal{B}^\infty$ be the set $[\![\Pi]\!](\Sigma^\infty)$.

The construction $\Sigma^\infty(V)$ gives rise to a functor $\Sigma^\infty \colon \mathbf{Set} \to \mathbf{Set}$, called the *free completely iterative monad* [5]. If there is no ambiguity, we will drop the injections κ_i when describing elements of $\Sigma^\infty(V)$. Note that $\Sigma^\infty(V)$ is final with property that for every $s \in \Sigma^\infty(V)$ either there are $f \in \Sigma$ and $\overrightarrow{t} \in (\Sigma^\infty(V))^{\mathrm{ar}(f)}$ with $\mathrm{root}(s) = f(\overrightarrow{t})$, or there is $x \in V$ with $\mathrm{root}(s) = x$. Finality allows us to specify unique maps into $\Sigma^\infty(V)$ by giving a coalgebra $X \to [\![\Sigma]\!](X) + V$. In particular, one can define for each $\sigma\colon V \to \Sigma^\infty$ the substitution $t[\sigma]$ of variables in the coterm t by σ as the coinductive extension of the following coalgebra.

$$\Sigma^\infty(V) \xrightarrow{\mathrm{root}} [\![\Sigma]\!](\Sigma^\infty(V)) + V \xrightarrow{[\mathrm{id}, \mathrm{root} \circ \sigma]} [\![\Sigma]\!](\Sigma^\infty(V))$$

Now that we have set up the basic terminology of coalgebras, we can give semantics to guarded terms from Def. 5. The idea is that guarded terms guarantee that we can always compute with them so far that we find a function symbol in head position, see Lem. 8. This function symbol determines then the label and branching of a node in the tree generated by a guarded term. If the computation reaches a constant or a variable, then we stop creating the tree at the present branch. This idea is captured by the following lemma.

Lemma 22. *There is a map* $[\![-]\!]_1 \colon \Lambda^{G,1}_\Sigma(\Gamma) \to \Sigma^\infty(\Gamma)$ *that is unique with*

1. *if* $M \equiv N$*, then* $[\![M]\!]_1 = [\![N]\!]_1$*, and*
2. $\mathrm{root}([\![M]\!]_1) = f\big(\overrightarrow{[\![N]\!]_1}\big)$ *for all* M *with* $M \longrightarrow\!\!\!\!\!\rightarrow f\,\overrightarrow{N}$*, and* $\mathrm{root}([\![M]\!]_1) = x$ *for all* M *with* $M \longrightarrow\!\!\!\!\!\rightarrow x$*,*

Proof sketch. By Lem. 8, we can define a coalgebra on the quotient of guarded terms by convertibility $c\colon \Lambda^{G,1}_\Sigma(\Gamma)/_\equiv \to [\![\Sigma]\!]\Big(\Lambda^{G,1}_\Sigma(\Gamma)/_\equiv\Big) + \Gamma$ with $c[M] = f[\overrightarrow{N}]$ if $M \longrightarrow\!\!\!\!\!\rightarrow f\,\overrightarrow{N}$ and $c[M] = x$ if $M \longrightarrow\!\!\!\!\!\rightarrow x$. This yields a map $h\colon \Lambda^{G,1}_\Sigma(\Gamma)/_\equiv \to \Sigma^\infty(\Gamma)$ and we can define $[\![-]\!]_1 = h \circ [-]$. The rest follows from uniqueness of h.

5.2 Interpretation of Basic Intuitionistic First-Order Formulae

In this section, we give an interpretation of the formulae in Def. 3, in which we restrict ourselves to guarded terms. This interpretation will be relative to models in the complete Herbrand universe. Since we later extend these models to Kripke models to be able to handle the later modality, we formulate these models already now in the language of fibrations [16,41].

Definition 23. Let $p\colon \mathbf{E} \to \mathbf{B}$ be a functor. Given an object $I \in \mathbf{B}$, the *fibre* $\mathbf{E}_I$ above I is the category of objects $A \in \mathbf{E}$ with $p(A) = I$ and morphisms $f\colon A \to B$ with $p(f) = \mathrm{id}_I$. The functor p is a *(split) fibration* if for every morphism $u\colon I \to J$ in $\mathbf{B}$ there is functor $u^*\colon \mathbf{E}_J \to \mathbf{E}_I$, such that $\mathrm{id}_I^* = \mathrm{Id}_{\mathbf{E}_I}$ and $v \circ u^* = u^* \circ v^*$. We call u^* the *reindexing along* u.

To give an interpretation of formulae, consider the following category **Pred**.

$$\mathbf{Pred} = \begin{cases} \text{objects:} & (X, P) \text{ with } X \in \mathbf{Set} \text{ and } P \subseteq X \\ \text{morphisms:} & f\colon (X,P) \to (Y,Q) \text{ is map } f\colon X \to Y \text{ with } f(P) \subseteq Q \end{cases}$$

The functor $\mathbb{P}\colon \mathbf{Pred} \to \mathbf{Set}$ with $\mathbb{P}(X, P) = X$ and $\mathbb{P}(f) = f$ is a split fibration, see [41], where the reindexing functor for $f\colon X \to Y$ is given by taking preimages: $f^*(Q) = f^{-1}(Q)$. Note that each fibre $\mathbf{Pred}_X$ is isomorphic to the complete lattice of predicates over X ordered by set inclusion. Thus, we refer to this fibration as the *predicate fibration*.

Let us now expose the logical structure of the predicate fibration. This will allow us to conveniently interpret first-order formulae over this fibration, but it comes at the cost of having to introduce a good amount of category theoretical language. However, doing so will pay off in Sec. 5.4, where we will construct another fibration out of the predicate fibration. We can then use category theoretical results to show that this new fibration admits the same logical structure and allows the interpretation of the later modality.

The first notion we need is that of fibred products, coproducts and exponents, which will allow us to interpret conjunction, disjunction and implication.

Definition 24. A fibration $p\colon \mathbf{E} \to \mathbf{B}$ has *fibred finite products* $(\mathbf{1}, \times)$, if each fibre $\mathbf{E}_I$ has finite products $(\mathbf{1}_I, \times_I)$ and these are preserved by reindexing: for all $f\colon I \to J$, we have $f^*(\mathbf{1}_J) = \mathbf{1}_I$ and $f^*(A \times_J B) = f^*(A) \times_I f^*(B)$. Fibred finite coproducts and exponents are defined analogously.

The fibration $\mathbb{P}$ is a so-called first-order fibration, which allows us to interpret first-order logic, see [41, Def. 4.2.1].

Definition 25. A fibration $p\colon \mathbf{E} \to \mathbf{B}$ is a *first-order fibration* if[4]

– $\mathbf{B}$ has finite products and the fibres of p are preorders;

[4] Technically, the quantifiers should also fulfil the Beck-Chevalley and Frobenius conditions, and the fibration should admit equality. Since these are fulfilled in all our models and we do not need equality, we will not discuss them here.

- p has fibred finite products $(\top, \wedge)$ and coproducts $(\bot, \vee)$ that distribute;
- p has fibred exponents $\to$; and
- p has existential and universal quantifiers $\exists_{I,J} \dashv \pi^*_{I,J} \dashv \forall_{I,J}$ for all projections $\pi_{I,J} \colon I \times J \to I$.

A *first-order λ-fibration* is a first-order fibration with Cartesian closed base $\mathbf{B}$.

The fibration $\mathbb{P} \colon \mathbf{Pred} \to \mathbf{Set}$ is a first-order λ-fibration, as all its fibres are posets and $\mathbf{Set}$ is Cartesian closed; $\mathbb{P}$ has fibred finite products $(\top, \cap)$, given by $\top_X = X$ and intersection; fibred distributive coproducts $(\emptyset, \cup)$; fibred exponents $\Rightarrow$, given by $(P \Rightarrow Q) = \{\vec{t} \mid \text{if } \vec{t} \in P, \text{then } \vec{t} \in Q\}$; and universal and existential quantifiers given for $P \in \mathbf{Pred}_{X \times Y}$ by

$$\forall_{X,Y} P = \{x \in X \mid \forall y \in Y.\, (x,y) \in P\} \quad \exists_{X,Y} P = \{x \in X \mid \exists y \in Y.\, (x,y) \in P\}.$$

The purpose of first-order fibrations is to capture the structure of first-order logic, while the λ-part takes care of higher-order features of the term language. In the following, we interpret types, contexts, guarded terms and formulae over guarded terms in the fibration $\mathbb{P} \colon \mathbf{Pred} \to \mathbf{Set}$. To this end, we define for types τ and context Γ sets $[\![\tau]\!]$ and $[\![\Gamma]\!]$, for guarded terms M with $\Gamma \vdash M : \tau$ a map $[\![M]\!] \colon [\![\Gamma]\!] \to [\![\tau]\!]$ in $\mathbf{Set}$ and for a formula $\Gamma \Vdash \varphi$ a predicate $[\![\varphi]\!] \in \mathbf{Pred}_{[\![\Gamma]\!]}$.

The semantics of types and contexts are given inductively, where the base type ι is interpreted as the set of coterms:

$$\begin{aligned} [\![\iota]\!] &= \Sigma^\infty & [\![\emptyset]\!] &= \mathbf{1} \\ [\![\tau \to \sigma]\!] &= [\![\sigma]\!]^{[\![\tau]\!]} & [\![\Gamma, x : \tau]\!] &= [\![\Gamma]\!] \times [\![\tau]\!] \end{aligned}$$

We note that a coterm $t \in \Sigma^\infty(V)$ can be seen as a map $(\Sigma^\infty)^V \to \Sigma^\infty$ by applying a substitution in $(\Sigma^\infty)^V$ to t: $\sigma \mapsto t[\sigma]$. In particular, the semantics of a guarded first-order term $M \in \Lambda^{G,1}_\Sigma(\Gamma)$ is equivalently a map $[\![M]\!]_1 \colon [\![\Gamma]\!] \to \Sigma^\infty$. We can now extend this map inductively to $[\![M]\!] \colon [\![\Gamma]\!] \to [\![\tau]\!]$ for all guarded terms $M \in \Lambda^G_\Sigma(\Gamma)$ with $\Gamma \vdash M : \tau$ by

$$\begin{aligned} [\![M]\!](\gamma)(\vec{t}) &= [\![M\ \vec{x}]\!]_1([\vec{x} \mapsto \vec{t}]) \qquad \vdash_g M : \sigma \text{ with } \mathrm{ar}(\sigma) = |\vec{t}| = |\vec{x}| \\ [\![c]\!](\gamma)(\vec{t}) &= c\ \vec{t} \\ [\![x]\!](\gamma) &= \gamma(x) \\ [\![M\ N]\!](\gamma) &= [\![M]\!](\gamma)\big([\![N]\!](\gamma)\big) \\ [\![\lambda x.\, M]\!](\gamma)(t) &= [\![M]\!](\gamma[x \mapsto t]) \end{aligned}$$

Lemma 26. *The mapping $[\![-]\!]$ is a well-defined function from guarded terms to functions, such that $\Gamma \vdash M : \tau$ implies $[\![M]\!] \colon [\![\Gamma]\!] \to [\![\tau]\!]$.*

Since $\mathbb{P} \colon \mathbf{Pred} \to \mathbf{Set}$ is a first-order fibration, we can interpret inductively all logical connectives of the formulae from Def. 3 in this fibration. The only case that is missing is the base case of predicate symbols. Their interpretation will be given over a Herbrand model that is constructed as the largest fixed point of an operator over all predicate interpretations in the Herbrand base. Both the operator and the fixed point are the subjects of the following definition.

Definition 27. We let the set of *interpretations* $\mathcal{I}$ be the powerset $\mathcal{P}(\mathcal{B}^\infty)$ of the complete Herbrand base. For $I \in \mathcal{I}$ and $p \in \Pi$, we denote by $I(p)$ the interpretation of p in I (the fibre of I above p)

$$I(p) = \left\{ \vec{t} \in (\Sigma^\infty)^{\mathrm{ar}(p)} \;\middle|\; p(\vec{t}) \in I \right\}.$$

Given a set P of Herbrand clauses, we define a monotone function $\Phi \colon \mathcal{I} \to \mathcal{I}$ by

$$\Phi(I) = \{\psi[\sigma] \mid (\forall \vec{x}.\, \textstyle\bigwedge_{k=1}^{n} \varphi_k \to \psi) \in P, \sigma \colon |\vec{x}| \to \Sigma^\infty, \forall k.\, \varphi_k[\sigma] \in I\},$$

where $\varphi_k[\sigma]$ is the extension of substitution from coterms to the Herbrand base by using functoriality of $[\![\Pi]\!]$. The *(complete) Herbrand model* $\mathcal{M}_P$ of P is the largest fixed point of Φ, which exists because $\mathcal{I}$ is a complete lattice.

Given a formula φ with $\Gamma \Vdash \varphi$ that contains only guarded terms, we define the semantics of φ in **Pred** from an interpretation $I \in \mathcal{I}$ inductively as follows.

$$\begin{aligned}
[\![\Gamma \Vdash p\, \vec{M}]\!]_I &= \left(\overrightarrow{[\![M]\!]}\right)^* (I(p)) & \\
[\![\Gamma \Vdash \top]\!]_I &= \top_{[\![\Gamma]\!]} & \\
[\![\Gamma \Vdash \varphi \,\Box\, \psi]\!]_I &= [\![\Gamma \Vdash \varphi]\!]_I \,\Box\, [\![\Gamma \Vdash \psi]\!]_I & \Box \in \{\wedge, \vee, \to\} \\
[\![\Gamma \Vdash Qx : \tau.\, \varphi]\!]_I &= Q_{[\![\Gamma]\!],[\![\tau]\!]}\, [\![\Gamma, x : \tau \Vdash \varphi]\!]_I & Q \in \{\forall, \exists\}
\end{aligned}$$

Lemma 28. *The mapping $[\![-]\!]$ is a well-defined function from formulae to predicates, such that $\Gamma \Vdash \varphi$ implies $[\![\varphi]\!] \subseteq [\![\Gamma]\!]$ or, equivalently, $[\![\varphi]\!] \in \mathbf{Pred}_{[\![\Gamma]\!]}$.*

This concludes the semantics of types, terms and formulae. We now turn to show that coinductive uniform proofs are sound for this interpretation.

5.3 Soundness of Coinductive Uniform Proofs for Herbrand Models

In this section, we give a direct proof of soundness for the coinductive uniform proof system from Sec. 3. Later, we will obtain another soundness result by combining the proof translation from Thm. 18 with the soundness of $\mathbf{iFOL}_{\blacktriangleright}$ (Thm. 39 and 42). The purpose of giving a direct soundness proof for uniform proofs is that it allows the extraction of a coinduction invariant, see Lem. 32.

The main idea is as follows. Given a formula φ and a uniform proof π for $\Sigma; P \looparrowright \varphi$, we construct an interpretation $I \in \mathcal{I}$ that validates φ, i.e. $[\![\varphi]\!]_I = \top$, and that is contained in the complete Herbrand model $\mathcal{M}_P$. Combining these two facts, we obtain that $[\![\varphi]\!]_{\mathcal{M}_P} = \top$, and thus the soundness of uniform proofs.

To show that the constructed interpretation I is contained in $\mathcal{M}_P$, we use the usual coinduction proof principle, as it is given in the following definition.

Definition 29. An *invariant for* $K \in \mathcal{I}$ is a set $I \in \mathcal{I}$, such that $K \subseteq I$ and I is a Φ-invariant, that is, $I \subseteq \Phi(I)$. If K has an invariant, then $K \subseteq \mathcal{M}_P$ because $\mathcal{M}_P$ is the largest fixed point of Φ.

Thus, our goal is now to construct an interpretation together with an invariant. This invariant will essentially collect and iterate all the substitution that appear in a proof. For this we need the ability to compose substitutions of coterms, which we derive from the monad [5] $(\Sigma^{\infty}, \eta, \mu)$ with $\mu\colon \Sigma^{\infty}\Sigma^{\infty} \Rightarrow \Sigma^{\infty}$.

Definition 30. A *(Kleisli-)substitution* θ from V to W, written $\theta\colon V \twoheadrightarrow W$, is map $V \to \Sigma^{\infty}(W)$. Composition of $\theta\colon V \twoheadrightarrow W$ and $\delta\colon U \twoheadrightarrow V$ is given by

$$\theta \circledcirc \delta = U \xrightarrow{\delta} \Sigma^{\infty}(V) \xrightarrow{\Sigma^{\infty}(\theta)} \Sigma^{\infty}(\Sigma^{\infty}(W)) \xrightarrow{\mu_W} \Sigma^{\infty}(W).$$

The notions in the following definition will allow us to easily organise and iterate the substitutions that occur in a uniform proof.

Definition 31. Let S be a set with $S = \{1, \ldots, n\}$ for some $n \in \mathbb{N}$. We call the set S^* of lists over S the set of *substitution identifiers*. Suppose that we have substitutions $\theta_0\colon V \twoheadrightarrow \emptyset$ and $\theta_k\colon V \twoheadrightarrow V$ for each $k \in S$. Then we can define a map $\Theta\colon S^* \to (\Sigma^{\infty})^V$, which turns each substitution identifier into a substitution, by iteration from the right:

$$\Theta(\varepsilon) = \theta_0 \quad \text{and} \quad \Theta(w : k) = \Theta(w) \circledcirc \theta_k$$

After introducing all these notations, we can finally give the outline of the soundness proof for uniform proofs relative to the complete Herbrand model. Given an H-formula $\forall \vec{x}.\, \varphi$, we note that a uniform proof π for $\Sigma; P \looparrowright \forall \vec{x}.\, \varphi$ has to start with

$$\cfrac{\cfrac{\vec{c} : \iota, \Sigma; P; \Delta \Longrightarrow \langle \varphi[\vec{c}/\vec{x}] \rangle \qquad \vec{c} : \iota \notin \Sigma}{\Sigma; P; \forall \vec{x}.\, \varphi \Longrightarrow \langle \forall \vec{x}.\, \varphi \rangle}\ \forall R\langle\rangle}{\Sigma; P \looparrowright \forall \vec{x}.\, \varphi}\ \text{CO-FIX}$$

where the eigenvariables in $\vec{c}$ are all distinct. We let Σ^c be the signature $\vec{c} : \iota, \Sigma$ and C be the set of variables in $\vec{c}$. Furthermore, suppose the following is a valid uniform proof tree in π.

$$\cfrac{\cfrac{\Sigma^c; P; \Delta \xRightarrow{\varphi[\vec{N}/\vec{x}]} A}{\Sigma^c; P; \Delta \xRightarrow{\forall \vec{x}.\, \varphi \in \Delta} A}\ \forall L}{\Sigma^c; P; \Delta \Longrightarrow A}\ \text{DECIDE}$$

This proof tree gives rise to a substitution $\delta\colon C \twoheadrightarrow C$ by $\delta(c) = [\![N_c]\!]$, which we call an *agent* of π. We let $D \subseteq \mathrm{At}_1^g$ be the set of atoms that are proven in π:

$$D = \{A \mid \Sigma^c; P; \Delta \Longrightarrow \langle A \rangle \text{ or } \Sigma^c; P; \Delta \Longrightarrow A \text{ appears in } \pi\}$$

From the agents and atoms in π we extract an invariant for the goal formula.

Lemma 32. *Suppose that φ is an H^g-formula of the form $\forall \vec{x}.\, A_1 \wedge \cdots \wedge A_n \to A_0$ and that there is a proof π for $\Sigma; P \looparrowright \varphi$. Let D be the proven atoms in π and $\theta_0, \ldots, \theta_s$ be the agents of π. Define $A^c_k = A_i[\vec{c}/\vec{x}]$ and suppose further that I_1 is an invariant for $\{A^c_k[\Theta(\varepsilon)] \mid 1 \leq k \leq n\}$. If we put*

$$I_2 = \bigcup_{w \in S^*} D[\Theta(w)]$$

then $I_1 \cup I_2$ is an invariant for $A^c_0[\Theta(\varepsilon)]$.

Once we have Lem. 32 the following soundness theorem is easily proven.

Theorem 33. *If φ is an H^g-formula and $\Sigma; P \looparrowright \varphi$, then $[\![\varphi]\!]_{\mathcal{M}_P} = \top$.*

Finally, we show that extending logic programs with coinductively proven lemmas is sound. This follows easily by coinduction.

Theorem 34. *Let φ be an H^g-formula of the shape $\forall \vec{x}.\, \psi_1 \to \psi_2$ and $\overrightarrow{M} \in \left(\Lambda^{G,1}_{\Sigma}\right)^{|\vec{x}|}$. If $\Sigma; P \looparrowright \varphi$ and $\left[\!\!\left[\psi_1\left[\overrightarrow{M}/\vec{x}\right]\right]\!\!\right]_{\mathcal{M}_P} = \top$, then $\mathcal{M}_{P \cup \left\{\psi_2\left[\overrightarrow{M}/\vec{x}\right]\right\}} = \mathcal{M}_P$. Hence, $P \cup \left\{\psi_2\left[\overrightarrow{M}/\vec{x}\right]\right\}$ is a conservative extension of P with respect to the complete Herbrand model.*

As a consequence of this theorem, we can prove a formula φ by a coinductive uniform proof for $\Sigma; P \looparrowright \varphi$ and then prove other formulae ψ by assuming ground atoms A that follow from φ. In other words, proofs for $\Sigma; P, A \looparrowright \psi$ are sound with respect to $\mathcal{M}_P$.

5.4 Soundness of iFOL$_\blacktriangleright$ over Herbrand Models

In this section, we demonstrate how the logic **iFOL**$_\blacktriangleright$ can be interpreted over Herbrand models. Recall that we obtained a fixed point model from the monotone map Φ on interpretations. In what follows, it is crucial that we construct the greatest fixed point of Φ by iteration, c.f. [6,28,68]: Let **Ord** be the class of all ordinals equipped with their (well-founded) order. We denote by $\mathbf{Ord}^{\mathrm{op}}$ the class of ordinals with their reversed order and define a monotone function $\overleftarrow{\Phi} : \mathbf{Ord}^{\mathrm{op}} \to \mathcal{I}$, where we write the argument ordinal in the subscript, by

$$\overleftarrow{\Phi}_\alpha = \bigcap\nolimits_{\beta < \alpha} \Phi\big(\overleftarrow{\Phi}_\beta\big).$$

Note that this definition is well-defined because $<$ is well-founded and because of monotonicity of Φ, see [14]. Since $\mathcal{I}$ is a complete lattice, there is an ordinal α such that $\overleftarrow{\Phi}_\alpha = \Phi\big(\overleftarrow{\Phi}_\alpha\big)$, at which point $\overleftarrow{\Phi}_\alpha$ is the largest fixed point $\mathcal{M}_P$ of Φ. In what follows, we will utilise this construction to give semantics to **iFOL**$_\blacktriangleright$.

The fibration $\mathbb{P} : \mathbf{Pred} \to \mathbf{Set}$ gives rise to another fibration as follows. We let $\overline{\mathbf{Pred}}$ be the category of functors (monotone maps) with fixed predicate domain:

$$\overline{\mathbf{Pred}} = \begin{cases} \text{objects:} & u : \mathbf{Ord}^{\mathrm{op}} \to \mathbf{Pred}, \text{such that } \mathbb{P} \circ u \text{ is constant} \\ \text{morphisms:} & u \to v \text{ are natural transformations } f : u \to v, \\ & \text{such that } \mathbb{P}f : \mathbb{P} \circ u \Rightarrow \mathbb{P} \circ v \text{ is the identity} \end{cases}$$

The fibration $\overline{\mathbb{P}}\colon \overline{\mathbf{Pred}} \to \mathbf{Set}$ is defined by evaluation at any ordinal (here 0), i.e. by $\overline{\mathbb{P}}(u) = \mathbb{P}(u(0))$ and $\overline{\mathbb{P}}(f) = (\mathbb{P}f)_0$, and reindexing along $f\colon m \to n$ by applying the reindexing of $\mathbb{P}$ point-wise, i.e. by $f^{\#}(u)_\alpha = f^*(u_\alpha)$.

Note that there is a (full) embedding $K\colon \mathbf{Pred} \to \overline{\mathbf{Pred}}$ that is given by $K(X, P) = (X, \overline{P})$ with $\overline{P}_\alpha = P$. One can show [14] that $\overline{\mathbb{P}}$ is again a first-order fibration and that it models the later modality, as in the following theorem.

Theorem 35. *The fibration $\overline{\mathbb{P}}$ is a first-order fibration. If necessary, we denote the first-order connectives by $\dot{\top}$, $\dot{\wedge}$ etc. to distinguish them from those in* $\mathbf{Pred}$*. Otherwise, we drop the dots. Finite (co)products and quantifiers are given point-wise, while for $X \in \mathbf{Set}$ and $u, v \in \overline{\mathbf{Pred}}_X$ exponents are given by*

$$(v \mathrel{\dot{\Rightarrow}} u)_\alpha = \bigcap\nolimits_{\beta \leq \alpha} (v_\beta \Rightarrow u_\beta).$$

There is a fibred functor $\blacktriangleright\colon \overline{\mathbf{Pred}} \to \overline{\mathbf{Pred}}$ with $\overline{\pi} \circ \blacktriangleright = \overline{\pi}$ given on objects by

$$(\blacktriangleright u)_\alpha = \bigcap\nolimits_{\beta < \alpha} u_\beta$$

and a natural transformation $\mathrm{next}\colon \mathrm{Id} \Rightarrow \blacktriangleright$ *from the identity functor to $\blacktriangleright$. The functor $\blacktriangleright$ preserves reindexing, products, exponents and universal quantification: $\blacktriangleright(f^{\#}u) = f^{\#}(\blacktriangleright u)$, $\blacktriangleright(u \wedge v) = \blacktriangleright u \wedge \blacktriangleright v$, $\blacktriangleright(u^v) \to (\blacktriangleright u)^{\blacktriangleright v}$, $\blacktriangleright(\overline{\forall}_n u) = \overline{\forall}_n(\blacktriangleright u)$. Finally, for all $X \in \mathbf{Set}$ and $u \in \overline{\mathbf{Pred}}_X$, there is* $\text{löb}\colon (\blacktriangleright u \mathrel{\dot{\Rightarrow}} u) \to u$ *in $\overline{\mathbf{Pred}}_X$.*

Using the above theorem, we can extend the interpretation of formulae to $\mathbf{iFOL}_\blacktriangleright$ as follows. Let $u\colon \mathbf{Ord}^{\mathrm{op}} \to \mathcal{I}$ be a descending sequence of interpretations. As before, we define the restriction of u to a predicate symbol $p \in \Pi$ by $u(p)_\alpha = u_\alpha(p) = \{\vec{t} \mid p(\vec{t}) \in u_\alpha\}$. The semantics of formulae in $\mathbf{iFOL}_\blacktriangleright$ as objects in $\overline{\mathbf{Pred}}$ is given by the following iterative definition.

$$\begin{aligned}
[\![\Gamma \Vdash p\,\vec{M}]\!]_u &= \left(\overrightarrow{[\![M]\!]}\right)^{\#}(u(p)) \\
[\![\Gamma \Vdash \top]\!]_I &= \dot{\top}_{[\![\Gamma]\!]} \\
[\![\Gamma \Vdash \varphi \mathbin{\Box} \psi]\!]_u &= [\![\Gamma \Vdash \varphi]\!]_u \mathbin{\Box} [\![\Gamma \Vdash \psi]\!]_u && \Box \in \{\wedge, \vee, \to\} \\
[\![\Gamma \Vdash Qx : \tau.\, \varphi]\!]_u &= Q_{[\![\Gamma]\!], [\![\tau]\!]}\, [\![\Gamma, x : \tau \Vdash \varphi]\!]_u && Q \in \{\forall, \exists\} \\
[\![\Gamma \Vdash \blacktriangleright \varphi]\!]_u &= \blacktriangleright [\![\Gamma \Vdash \varphi]\!]_u
\end{aligned}$$

The following lemma is the analogue of Lem. 28 for the interpretation of formulae without the later modality.

Lemma 36. *The mapping $[\![-]\!]$ is a well-defined map from formulae in* $\mathbf{iFOL}_\blacktriangleright$ *to sequences of predicates, such that $\Gamma \Vdash \varphi$ implies $[\![\varphi]\!] \in \overline{\mathbf{Pred}}_{[\![\Gamma]\!]}$.*

Lemma 37. *All rules of* $\mathbf{iFOL}_\blacktriangleright$ *are sound with respect to the interpretation $[\![-]\!]_u$ of formulae in $\overline{\mathbf{Pred}}$, that is, if $\Gamma \mid \Delta \vdash \varphi$, then $\left(\bigwedge_{\psi \in \Delta} [\![\psi]\!]_u \mathrel{\dot{\Rightarrow}} [\![\varphi]\!]_u\right) = \dot{\top}$. In particular, $\Gamma \vdash \varphi$ implies $[\![\varphi]\!]_u = \dot{\top}$.*

The following lemma shows that the guarding of a set of formulae is valid in the chain model that they generate.

Lemma 38. *If φ is an H-formula in P, then* $[\![\overline{\varphi}]\!]_{\overleftarrow{\Phi}} = \dot{\top}$.

Combining this with soundness from Lem. 37, we obtain that provability in $\mathbf{iFOL}_{\blacktriangleright}$ relative to a logic program P is sound for the model of P.

Theorem 39. *For all logic programs P, if $\Gamma \mid \overline{P} \vdash \varphi$ then* $[\![\varphi]\!]_{\overleftarrow{\Phi}} = \dot{\top}$.

The final result of this section is to show that the descending chain model, which we used to interpret formulae of $\mathbf{iFOL}_{\blacktriangleright}$, is sound and complete for the fixed point model, which we used to interpret the formulae of coinductive uniform proofs. This will be proved in Thm. 42 below. The easiest way to prove this result is by establishing a functor $\overline{\mathbf{Pred}} \to \mathbf{Pred}$ that maps the chain $\overleftarrow{\Phi}$ to the model $\mathcal{M}_P$, and that preserves and reflects truth of first-order formulae (Prop. 41). We will phrase the preservation of truth of first-order formulae by a functor by appealing to the following notion of fibrations maps, cf. [41, Def.4.3.1].

Definition 40. Let $p \colon \mathbf{E} \to \mathbf{B}$ and $q \colon \mathbf{D} \to \mathbf{A}$ be fibrations. A *fibration map* $p \to q$ is a pair $(F \colon \mathbf{E} \to \mathbf{D}, G \colon \mathbf{B} \to \mathbf{A})$ of functors, s.t. $q \circ F = G \circ p$ and F preserves Cartesian morphisms: if $f \colon X \to Y$ in $\mathbf{E}$ is Cartesian over $p(f)$, then $F(f)$ is Cartesian over $G(p(f))$. (F, G) is a map of *first-order (λ-)fibrations*, if p and q are first-order (λ-)fibrations, and F and G preserve this structure.

Let us now construct a first-order λ-fibration map $\overline{\mathbf{Pred}} \to \mathbf{Pred}$. We note that since every fibre of the predicate fibration is a complete lattice, for every chain $u \in \overline{\mathbf{Pred}}_X$ there exists an ordinal α at which u stabilises. This means that there is a limit $\lim u$ of u in $\mathbf{Pred}_X$, which is the largest subset of X, such that $\forall \alpha.\ \lim u \subseteq u_\alpha$. This allows us to define a map $L \colon \overline{\mathbf{Pred}} \to \mathbf{Pred}$ by

$$
\begin{aligned}
L(X, u) &= (X, \lim u) \\
L(f \colon (X, u) \to (Y, v)) &= f.
\end{aligned}
$$

In the following proposition, we show that L gives us the ability to express first-order properties of limits equivalently through their approximating chains. This, in turn, provides soundness and completeness for the interpretation of the logic $\mathbf{iFOL}_{\blacktriangleright}$ over descending chains with respect to the largest Herbrand model.

Proposition 41. $L \colon \overline{\mathbf{Pred}} \to \mathbf{Pred}$*, as defined above, is a map of first-order fibrations. Furthermore, L is right-adjoint to the embedding $K \colon \mathbf{Pred} \to \overline{\mathbf{Pred}}$. Finally, for each $p \in \Pi$ and $u \in \overline{\mathbf{Pred}}_{\mathcal{B}^\infty}$, we have $L(u(p)) = L(u)(p)$.*

We get from Prop. 41 soundness and completeness of $\overleftarrow{\Phi}$ for Herbrand models.

Theorem 42. *If φ is $\blacktriangleright$-free (Def. 3) then $[\![\varphi]\!]_{\overleftarrow{\Phi}} = \dot{\top}$ if and only if $[\![\varphi]\!]_{\mathcal{M}_P} = \top$.*

Proof sketch. First, one shows for all $\blacktriangleright$-free formulae φ that $L([\![\varphi]\!]_{\overleftarrow{\Phi}}) = [\![\varphi]\!]_{\mathcal{M}_P}$ by induction on φ and using Prop. 41. Using this identity and $K \dashv L$, the result is then obtained by the following adjoint correspondence.

$$
\begin{array}{rl}
\dot{\top} = K(\top) \longrightarrow [\![\varphi]\!]_{\overleftarrow{\Phi}} & \text{in } \overline{\mathbf{Pred}} \\
\hline\hline
\top \longrightarrow L\big([\![\varphi]\!]_{\overleftarrow{\Phi}}\big) = [\![\varphi]\!]_{\mathcal{M}_P} & \text{in } \mathbf{Pred}
\end{array}
$$

□

6 Conclusion, Related Work and the Future

In this paper, we provided a comprehensive theory of resolution in coinductive Horn-clause theories and coinductive logic programs. This theory comprises of a uniform proof system that features a form of guarded recursion and that provides operational semantics for proofs of coinductive predicates. Further, we showed how to translate proofs in this system into proofs for an extension of intuitionistic FOL with guarded recursion, and we provided sound semantics for both proof systems in terms of coinductive Herbrand models. The Herbrand models and semantics were thereby presented in a modern style that utilises coalgebras and fibrations to provide a conceptual view on the semantics.

Related Work

It may be surprising that automated *proof search for coinductive predicates* in first-order logic does not have a coherent and comprehensive theory, even after three decades [53,3], despite all the attention that it received as programming [2,26,37,39] and proof [29,30,34,35,40,52,57,58,59,60] method. The work that comes close to algorithmic proof search is the system CIRC [56]. However, CIRC is limited to equality proofs, and cannot handle general coinductive predicates and corecursive programming. Inductive and coinductive data types are also being added to SMT solvers [22,55]. However, both CIRC and SMT solving are inherently based on classical logic and are therefore not suited to situations where proof objects are relevant, like programming, type class inference or (dependent) type theory. Moreover, the proposed solutions, just like those in [36,61] can only deal with regular data, while our approach also works for irregular data, as we have seen in the **from**-example.

This paper subsumes Haskell type class inference [45,32] and exposes that the inference presented in those papers corresponds to coinductive proofs in *co-fohc* and *co-hohh*. Given that the proof systems proposed in this paper are constructive and that uniform proofs provide proofs (type inhabitants) in normal form, we could give a propositions-as-types interpretation to all eight coinductive uniform proof systems. This was done for *co-fohc* and *co-hohh* in [32], but we leave the remaining cube for future work.

Future Work

There are several directions that we wish to pursue in the future. First, we know that CUP is incomplete for the presented models, as it is intuitionistic and it lacks an admissible cut rule. The first can be solved by moving to Kripke/Beth-models, while the second is more delicate. To obtain an admissible cut rule one has to be able to prove several propositions simultaneously by coinduction. We leave this as an investigation for future work. Furthermore, we aim to extend our ideas to other situations like higher-order Horn clauses [38,25] and interactive proof assistants [7,10,27,21], typed logic programming, and logic programming that mix inductive and coinductive predicates.

References

1. Abbott, M., Altenkirch, T., Ghani, N.: Containers: Constructing strictly positive types. Theoretical Computer Science **342**(1), 3–27 (Sep 2005). 10.1016/j.tcs.2005.06.002
2. Abel, A., Pientka, B., Thibodeau, D., Setzer, A.: Copatterns: programming infinite structures by observations. In: POPL'13. pp. 27–38 (2013). 10.1145/2429069.2429075
3. Aczel, P.: Non-well-founded sets. Center for the Study of Lnaguage and Information, Stanford University (1988)
4. Aczel, P.: Algebras and Coalgebras. In: Algebraic and Coalgebraic Methods in the Mathematics of Program Construction. Springer (2002)
5. Aczel, P., Adámek, J., Milius, S., Velebil, J.: Infinite trees and completely iterative theories: A coalgebraic view. TCS **300**(1–3), 1–45 (May 2003). 10.1016/S0304-3975(02)00728-4
6. Adámek, J.: On final coalgebras of continuous functors. Theor. Comput. Sci. **294**(1/2), 3–29 (2003). 10.1016/S0304-3975(01)00240-7
7. team Agda, P.L.: Agda Documentation. Tech. rep., Chalmers and Gothenburg University (2015), http://wiki.portal.chalmers.se/agda/, version 2.4.2.5
8. Appel, A.W., Melliès, P.A., Richards, C.D., Vouillon, J.: A very modal model of a modern, major, general type system. In: POPL. pp. 109–122. ACM (2007). 10.1145/1190216.1190235
9. Atkey, R., McBride, C.: Productive coprogramming with guarded recursion. In: ICFP. pp. 197–208. ACM (2013). 10.1145/2500365.2500597
10. Baelde, D., et al.: Abella: A system for reasoning about relational specifications. J. Formalized Reasoning **7**(2), 1–89 (2014). 10.6092/issn.1972-5787/4650
11. Barendregt, H., Dekkers, W., Statman, R.: Lambda Calculus with Types. Cambridge University Press, Cambridge ; New York (Jul 2013)
12. Barr, M., Wells, C.: Category Theory for Computing Science. Prentice Hall international series in computer science, Prentice Hall, 2nd edn. (1995), http://www.tac.mta.ca/tac/reprints/articles/22/tr22abs.html
13. Basold, H.: Mixed Inductive-Coinductive Reasoning: Types, Programs and Logic. Ph.D. thesis, Radboud University Nijmegen (2018), http://hdl.handle.net/2066/190323
14. Basold, H.: Breaking the Loop: Recursive Proofs for Coinductive Predicates in Fibrations. ArXiv e-prints (Feb 2018), https://arxiv.org/abs/1802.07143
15. Beklemishev, L.D.: Parameter Free Induction and Provably Total Computable Functions. Theor. Comput. Sci. **224**(1-2), 13–33 (1999). 10.1016/S0304-3975(98)00305-3
16. Bénabou, J.: Fibered categories and the foundations of naive category theory. Journal of Symbolic Logic **50**(1), 10–37 (1985). 10.2307/2273784
17. Birkedal, L., Møgelberg, R.E.: Intensional Type Theory with Guarded Recursive Types qua Fixed Points on Universes. In: LICS. pp. 213–222. IEEE Computer Society (2013). 10.1109/LICS.2013.27
18. Birkedal, L., Møgelberg, R.E., Schwinghammer, J., Støvring, K.: First Steps in Synthetic Guarded Domain Theory: Step-Indexing in the Topos of Trees. In: Proceedings LICS 2011. pp. 55–64. IEEE Computer Society (2011). 10.1109/LICS.2011.16
19. Bizjak, A., Grathwohl, H.B., Clouston, R., Møgelberg, R.E., Birkedal, L.: Guarded Dependent Type Theory with Coinductive Types. In: FoSSaCS. Lecture Notes in Computer Science, vol. 9634, pp. 20–35. Springer (2016), https://arxiv.org/abs/1601.01586

20. Bjørner, N., et al.: Horn clause solvers for program verification. In: Fields of Logic and Computation II. LNCS, vol. 9300, pp. 24–51 (2015)
21. Blanchette, J., et al.: Foundational nonuniform (co)datatypes for higher-order logic. In: LICS'17. pp. 1–12. IEEE Computer Society (2017)
22. Blanchette, J.C., Peltier, N., Robillard, S.: Superposition with datatypes and codatatypes. In: IJCAR 2018 (2018)
23. Borceux, F.: Handbook of Categorical Algebra: Volume 1, Basic Category Theory. Cambridge University Press (Apr 2008)
24. Bottu, G., et al.: Quantified class constraints. In: Haskell Symp. pp. 148–161. ACM (2017)
25. Burn, T.C., Ong, C.L., Ramsay, S.J.: Higher-order constrained horn clauses for verification. PACMPL **2**(POPL), 11:1–11:28 (2018). `10.1145/3158099`, `http://doi.acm.org/10.1145/3158099`
26. Capretta, V.: General Recursion via Coinductive Types. Logical Methods in Computer Science **1**(2) (Jul 2005). `10.2168/LMCS-1(2:1)2005`
27. Coquand, T.: Infinite objects in type theory. In: TYPES'93. vol. 806, pp. 62–78 (1994)
28. Cousot, P., Cousot, R.: Constructive versions of Tarski's fixed point theorems. Pacific J. Math. **82**(1), 43–57 (1979), `http://projecteuclid.org/euclid.pjm/1102785059`
29. Dax, C., Hofmann, M., Lange, M.: A Proof System for the Linear Time μ-Calculus. In: Arun-Kumar, S., Garg, N. (eds.) Proceedings of FSTTCS 2006. LNCS, vol. 4337, pp. 273–284. Springer (2006). `10.1007/11944836_26`
30. Endrullis, J., Hansen, H.H., Hendriks, D., Polonsky, A., Silva, A.: A coinductive framework for infinitary rewriting and equational reasoning. In: RTA'15. pp. 143–159 (2015)
31. Farka, F., Komendantskaya, E., Hammond, K.: Coinductive soundness of corecursive type class resolution. In: Post-proc. of LOPSTR'16. LNCS, vol. 10184, pp. 311–327. Springer (2017)
32. Fu, P., Komendantskaya, E., Schrijvers, T., Pond, A.: Proof relevant corecursive resolution. In: FLOPS'16. pp. 126–143. Springer (2016)
33. Gambino, N., Kock, J.: Polynomial functors and polynomial monads. Math. Proc. Cambridge Phil. Soc. **154**(01), 153–192 (Jan 2013). `10.1017/S0305004112000394`
34. Giesl, J., Aschermann, C., Brockschmidt, M., Emmes, F., Frohn, F., Fuhs, C., Hensel, J., Otto, C., Plücker, M., Schneider-Kamp, P., Ströder, T., Swiderski, S., Thiemann, R.: Analyzing program termination and complexity automatically with aprove. J. Autom. Reasoning **58**(1), 3–31 (2017)
35. Giménez, E.: Structural recursive definitions in type theory. In: ICALP'98. pp. 397–408 (1998)
36. Gupta, G., Bansal, A., Min, R., Simon, L., Mallya, A.: Coinductive logic programming and its applications. In: Dahl, V., Niemelä, I. (eds.) Logic Programming. pp. 27–44. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
37. Hagino, T.: A typed lambda calculus with categorical type constructors. In: Category Theory in Computer Science. pp. 140–157. Lecture Notes in Computer Science, Springer (1987). `10.1007/3-540-18508-9_24`
38. Hashimoto, K., Unno, H.: Refinement type inference via horn constraint optimization. In: SAS. pp. 199–216 (2015)
39. Howard, B.T.: Inductive, Coinductive, and Pointed Types. In: Harper, R., Wexelblat, R.L. (eds.) Proceedings of ICFP '96. pp. 102–109. ACM (1996). `10.1145/232627.232640`

40. Hur, C.K., Neis, G., Dreyer, D., Vafeiadis, V.: The Power of Parameterization in Coinductive Proof. In: Proceedings of POPL 2013. pp. 193–206. POPL '13, ACM, New York, NY, USA (2013). 10.1145/2429069.2429093
41. Jacobs, B.: Categorical Logic and Type Theory. No. 141 in Studies in Logic and the Foundations of Mathematics, North Holland, Amsterdam (1999)
42. Jacobs, B.: Introduction to Coalgebra: Towards Mathematics of States and Observation. No. 59 in Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (2016). 10.1017/CBO9781316823187, http://www.cs.ru.nl/B.Jacobs/CLG/JacobsCoalgebraIntro.pdf
43. Kowalski, R.: Predicate logic as a programming language. Information Processing **74**, 569–574 (1974)
44. Lambek, J., Scott, P.J.: Introduction to Higher-Order Categorical Logic. Cambridge University Press (Mar 1988)
45. Lämmel, R., Jones, S.P.: Scrap your boilerplate with class: Extensible generic functions. In: ICFP'05. pp. 204–215 (2005)
46. Lloyd, J.W.: Foundations of Logic Programming, 2nd Edition. Springer (1987)
47. Miller, D., Nadathur, G.: Programming with Higher-order logic. Cambridge University Press (2012)
48. Milner, R.: A theory of type polymorphism in programming. J. Comput. Syst. Sci. **17**(3), 348–375 (1978)
49. Møgelberg, R.E.: A type theory for productive coprogramming via guarded recursion. In: CSL-LICS. pp. 71:1–71:10. ACM (2014). 10.1145/2603088.2603132
50. Nadathur, G., Mitchell, D.J.: Teyjus - A compiler and abstract machine based implementation of lambda-prolog. In: CADE'16. LNCS, vol. 1632, pp. 287–291. Springer (1999)
51. Nakano, H.: A Modality for Recursion. In: LICS. pp. 255–266. IEEE Computer Society (2000). 10.1109/LICS.2000.855774
52. Niwinski, D., Walukiewicz, I.: Games for the μ-Calculus. TCS **163**(1&2), 99–116 (1996). 10.1016/0304-3975(95)00136-0
53. Park, D.M.R.: Concurrency and Automata on Infinite Sequences. In: Deussen, P. (ed.) Proceedings of TCS'81. LNCS, vol. 104, pp. 167–183. Springer (1981). 10.1007/BFb0017309
54. Plotkin, G.D.: LCF considered as a programming language. Theoretical Computer Science **5**(3), 223–255 (Dec 1977). 10.1016/0304-3975(77)90044-5
55. Reynolds, A., Kuncak, V.: Induction for SMT solvers. In: VMCAI'15. LNCS, vol. 8931, pp. 80–98 (2015)
56. Roşu, G., Lucanu, D.: Circular Coinduction: A Proof Theoretical Foundation. In: CALCO. LNCS, vol. 5728, pp. 127–144. Springer (2009). 10.1007/978-3-642-03741-2_10
57. Rutten, J.: Universal Coalgebra: A Theory of Systems. TCS **249**(1), 3–80 (2000). 10.1016/S0304-3975(00)00056-6
58. Sangiorgi, D.: Introduction to Bisimulation and Coinduction. Cambridge University Press, New York, NY, USA (2011)
59. Santocanale, L.: A Calculus of Circular Proofs and Its Categorical Semantics. In: FoSSaCS. pp. 357–371 (2002). 10.1007/3-540-45931-6_25
60. Santocanale, L.: μ-Bicomplete Categories and Parity Games. RAIRO - ITA **36**(2), 195–227 (2002), https://doi.org/10.1051/ita:2002010
61. Simon, L., Bansal, A., Mallya, A., Gupta, G.: Co-logic programming: Extending logic programming with coinduction. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) Automata, Languages and Programming. pp. 472–483. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)

62. Smoryński, C.: Self-Reference and Modal Logic. Universitext, Springer-Verlag (1985)
63. Solovay, R.M.: Provability interpretations of modal logic. Israel Journal of Mathematics **25**(3), 287–304 (1976). 10.1007/BF02757006
64. Sulzmann, M., Stuckey, P.J.: Hm(x) type inference is clp(x) solving. J. Funct. Program. **18**(2), 251–283 (2008)
65. Terese: Term Rewriting Systems. Cambridge University Press (2003)
66. Turner, D.A.: Elementary Strong Functional Programming. In: Hartel, P.H., Plasmeijer, M.J. (eds.) Proceedings of FPLE'95. LNCS, vol. 1022, pp. 1–13. Springer (1995). 10.1007/3-540-60675-0_35
67. van den Berg, B., de Marchi, F.: Non-well-founded trees in categories. Annals of Pure and Applied Logic **146**(1), 40–59 (Apr 2007). 10.1016/j.apal.2006.12.001
68. Worrell, J.: On the final sequence of a finitary set functor. Theor. Comput. Sci. **338**(1-3), 184–199 (2005). 10.1016/j.tcs.2004.12.009

A Proofs for Section 2 (Preliminaries: Terms and Formulae)

Lemma 7. *Let M and N be guarded base terms with $\Gamma, x : \sigma \vdash_g M : \tau$ and $\Gamma \vdash_g N : \sigma$. Then $M[N/x]$ is a guarded base term with $\Gamma \vdash_g M[N/x] : \tau$.*

Proof. Let M and N be as above. We proceed by induction on the derivation that M is guarded to show that $M[N/x]$ is guarded as well.

- Suppose $M = y$ for some variable. If $y = x$, then $M[N/x] = N$ and $\tau = \sigma$. Thus, we have $\Gamma \vdash_g M[N/x] : \tau$.
- The case for signature symbols is immediate, as for $f \in \Sigma$ we have $f[N/x] = f$.
- Suppose $\Gamma, x : \sigma \vdash_g M\, P : \tau$. By the IH, we have $\Gamma \vdash_g M[N/x] : \gamma \to \tau$ and $\Gamma \vdash_g P[N/x] : \gamma$. Thus, we obain $\Gamma \vdash_g (M\, P)[N/x] : \tau$.
- Finally, assume that $\Gamma, x : \sigma \vdash_g \mathrm{fix}\, z. \lambda \overrightarrow{y}. f\, \overrightarrow{M} : \tau$. Then by IH, we have

$$\Gamma, z : \tau, y_1 : \iota, \ldots, y_{\mathrm{ar}(\tau)} : \iota \vdash_g M_i[N/x] : \iota$$

  and so $\Gamma \vdash_g \left(\mathrm{fix}\, z. \lambda \overrightarrow{y}. f\, \overrightarrow{M}\right)[N/x] : \tau$.

Lemma 8. *If M is a first-order guarded term with $M \in \Lambda_{\Sigma}^{G,1}(\Gamma)$, then M reduces to a unique head normal form. This means that either (i) there is either a unique $f \in \Sigma$ and terms $N_1, \ldots, N_{\mathrm{ar}(f)}$ with $\Gamma \vdash_g N_k : \iota$ and $M \longrightarrow\!\!\!\!\to f\, \overrightarrow{N}$, and moreover if $M \longrightarrow\!\!\!\!\to f\, \overrightarrow{L}$, then $\overrightarrow{N} \equiv \overrightarrow{L}$; or (ii) $M \longrightarrow\!\!\!\!\to x$ for some $x : \iota \in \Gamma$.*

Proof. The term M with $\Gamma \vdash_g M : \iota$ can have either of the following three shapes:

1. x, where $x : \iota \in \Gamma$
2. $f\, \overrightarrow{N}$ with $\Gamma \vdash_g N_k : \iota$, or
3. $(\mathrm{fix}\, x. \lambda \overrightarrow{y}. f\, \overrightarrow{M})\, \overrightarrow{N}$ with $\Gamma, x : \tau, y_1 : \iota, \ldots, y_{\mathrm{ar}(\tau)} : \iota \vdash_g M_k : \iota$ for $k = 1, \ldots, \mathrm{ar}(f)$ and $\Gamma \vdash_g N_i : \iota$ for $i = 1, \ldots, \mathrm{ar}(\tau)$,

because variables can only occur in argument position due to the order restriction of the types in Γ. In the first two cases we are done immediately. For the third case, we let $P = \mathrm{fix}\, x. \lambda \overrightarrow{y}. f\, \overrightarrow{M}$ and then find that

$$P\, \overrightarrow{N} \longrightarrow\!\!\!\!\to f\, \left(\overrightarrow{M}\left[P/x, \overrightarrow{N}/\overrightarrow{y}\right]\right).$$

Lemma 7 gives us now that each $M_i\left[P/x, \overrightarrow{N}/\overrightarrow{y}\right]$ is guarded. Finally, if $M \longrightarrow\!\!\!\!\to f\overrightarrow{L}$, then $\overrightarrow{N} \equiv \overrightarrow{L}$ by confluence of the reduction relation.

In Lem. 7 we have shown that guarded base terms are stable under substitution, that is, substituting a guarded base term into another results into a guarded base term. The following lemma shows that the same is true for guarded terms. This result is necessary to define substitution for formulae over guarded terms, see Def. 9.

Lemma 43. *Let $M \in \Lambda^{G}_{\Sigma}(\Gamma, x)$ and $N \in \Lambda^{G}_{\Sigma}(\Gamma)$ be guarded terms with $\Gamma, x : \tau \vdash M : \sigma$ and $\Gamma \vdash N : \tau$. Then $M\,[N/x] \in \Lambda^{G}_{\Sigma}(\Gamma)$ and $\Gamma \vdash M\,[N/x] : \sigma$.*

Proof. By an easy induction on M.

B Proofs for Section 4 (Coinductive Uniform Proofs and Intuitionistic Logic)

Theorem 18. *If P is a logic program over a first-order signature Σ and the sequent $\Sigma; P \looparrowright \varphi$ is provable in co-hohh$_{\text{fix}}$, then $\overline{P} \vdash \varphi$ is provable in* **iFOL$_\blacktriangleright$**.

The following admissible rules are easily derivable in the logic **iFOL$_\blacktriangleright$** and are essential in showing soundness of *co-hohh* with respect to **iFOL$_\blacktriangleright$**.

Lemma 44.

$$\frac{\Gamma \Vdash \varphi \qquad \varphi \equiv \psi}{\Gamma \Vdash \psi} \qquad \frac{\Gamma \mid \Delta \vdash \varphi}{\Gamma \Vdash \varphi} \qquad \frac{\Gamma \mid \Delta \vdash \varphi \qquad x : \tau \notin \Gamma}{\Gamma, x : \tau \mid \Delta \vdash \varphi}\ \textbf{(Weak)}$$

$$\frac{\Gamma \mid \Delta \vdash \blacktriangleright(\varphi \wedge \psi)}{\Gamma \mid \Delta \vdash \blacktriangleright\varphi \wedge \blacktriangleright\psi} \qquad \frac{\Gamma \mid \Delta \vdash \blacktriangleright\varphi \wedge \blacktriangleright\psi}{\Gamma \mid \Delta \vdash \blacktriangleright(\varphi \wedge \psi)}$$

$$\frac{\Gamma \mid \Delta \vdash \blacktriangleright(\forall x : \tau.\, \varphi)}{\Gamma \mid \Delta \vdash \forall x : \tau.\, \blacktriangleright\varphi} \qquad \frac{\Gamma \mid \Delta \vdash \forall x : \tau.\, \blacktriangleright\varphi}{\Gamma \mid \Delta \vdash \blacktriangleright(\forall x : \tau.\, \varphi)}$$

Proof Proof Sketch for Theorem 18. First we note that the coinduction goal in *co-hohh*$_{\text{fix}}$ is given by the following grammar.

$$\mathrm{CG} ::= \mathrm{At}^{s}_{\omega} \mid \mathrm{CG} \to \mathrm{CG} \mid \mathrm{CG} \wedge \mathrm{CG} \mid \forall x : \tau.\, \mathrm{CG}$$

Thus, a coinduction goal is the restriction of FOL to implication, conjunction and universal quantification. Note that such a coinduction goal is intuitionistically equivalent to a conjunction of Horn-clauses.

Assume that we are given a uniform proof tree T. We translate this tree into a proof tree T' in **iFOL$_\blacktriangleright$**. The proof proceeds in the following steps.

1. The first step of a proof tree T starting in $\Sigma; P \looparrowright \varphi$ must be an application of the CO-FIX rule to a proof tree T_1 ending in $\Sigma; P; \varphi \Longrightarrow \langle\varphi\rangle$. This step can be directly translated into an application of the Löb rule. Hence, if T_1' is the translation of T_1 with conclusion $\overline{P}, \blacktriangleright\varphi \vdash \varphi$, then T' is given by applying **(Löb)** to T_1', thereby obtaining a proof tree ending in the desired sequent $\overline{P} \vdash \varphi$.
2. The next step must then be either $\forall R\langle\rangle$, $\wedge R\langle\rangle$, $\to R\langle\rangle$ or DECIDE$\langle\rangle$. To prove this by induction on the proof tree, we need to define coinduction goal contexts. These are contexts $\varphi[-]$ with a hole $[-]$, such that plugging an atom from At^{s}_{ω} into the hole yields a coinduction goal. More generally, we will

need contexts with multiple holes $[-]_i$ that are indexed from 0 to n for some $n \in \mathbb{N}$. Formally, such contexts are given by the following grammar.

$$H ::= [-]_i \mid [-]_i \to H \mid \forall x : \tau.\, H$$
$$C ::= H \mid C \wedge C$$

Let C be a context, we write $C[\overrightarrow{\varphi}]$ for the formula that arises by replacing the holes $[-]_i$ by φ_i. Note that this may result in binding of free variables in φ_i and ψ.

We prove by induction on proof trees that for any context Γ, any set of formulae P, any context C and any proof for $\Sigma, \Gamma; P; C[\varphi] \Longrightarrow \langle\varphi\rangle$ that there is a proof for $\Gamma \mid \overline{P}, C[\blacktriangleright \varphi] \vdash \varphi$. The translation for this step follows then by taking Γ to be empty and C to be $[-]$.

In the $\forall R\langle\rangle$ case, we have that a proof tree ending in $\Sigma, \Gamma; P; C[\forall x.\, \varphi] \Longrightarrow \langle\forall x.\, \varphi\rangle$ has $\Sigma, \Gamma, x; P; C[\forall x.\, \varphi] \Longrightarrow \langle\varphi\rangle$ as its only premise. By putting $C' = C[\forall x.\, [-]]$, this premise can be written as $\Sigma, \Gamma, x; P; C'[\varphi] \Longrightarrow \langle\varphi\rangle$ from which we obtain by the induction hypothesis a proof tree for $\Gamma, x \mid \overline{P}, C'[\blacktriangleright \varphi] \vdash \varphi$. Using the derived rule **($\blacktriangleright$-Pres-$\forall_l$)** from Lem. 44 and by the rule **($\forall$-I)**, we thus obtain a proof tree for the sequent $\Gamma \mid \overline{P}, C[\blacktriangleright(\forall x.\, \varphi)] \vdash \forall x.\, \varphi$.

For the cases $\wedge R\langle\rangle$ and $\to R\langle\rangle$, one proceeds in similarly as for $\forall R\langle\rangle$ by appealing to the fact that $\blacktriangleright$ preserves conjunction and implication, respectively. In the case of $\to R\langle\rangle$ it is important to note that if $\varphi \to \psi$ is a Horn-clause, then $\overline{\varphi} = \varphi$ and so $\overline{P}, \varphi = \overline{P, \varphi}$. Hence, if a hypothesis is added by $\to R\langle\rangle$ to P, then all the formulae in P are still guarded.

Finally, the DECIDE$\langle\rangle$ rule is dealt with in the next step.

3. For an application of either of the decide rules, there are generally two cases to consider: either the clause D is selected from P by DECIDE$\langle\rangle$ or DECIDE, or DECIDE selects $C[\overrightarrow{A}]$. In both cases, we proceed by induction to analyse of the proof tree for $\Sigma, \Gamma; P; C[\overrightarrow{A}] \overset{D}{\Rightarrow} B$.

   Define $H \coloneqq C[\overrightarrow{A}]$. We then obtain the following cases from the fact that D and H are Horn-clauses with the later modality in specific places.

   (a) $D \in P$ is selected. Then the resulting proof tree in **iFOL$_\blacktriangleright$** will have at the root $\Gamma \mid \overline{P}, H \vdash B$ and as leaves sequents of the form $\Gamma \mid \overline{P}, H \vdash \blacktriangleright C$ for some atoms C.

   (b) H is selected. Then the resulting proof tree in **iFOL$_\blacktriangleright$** will have at the root $\Gamma \mid \overline{P}, H \vdash \blacktriangleright A_k$ for some k, and as leaves sequents of the form $\Gamma \mid \overline{P}, H \vdash \blacktriangleright A_i$ for some i.

   Our goal is now to combine such proof trees. The only mismatch might occurs whenever we have a proof tree that has $\Gamma \mid \overline{P}, H \vdash B$ as root (first case) that has to be attached to a leave of another proof tree (from either case), which will be of the form $\Gamma \mid \overline{P}, H \vdash \blacktriangleright C$ for some atom C. Since this match arises from a uniform proof, we have that $C = B$. Hence, we can combine these two trees by appealing to the **(Next)** rule:

$$\dfrac{\dfrac{\vdots}{\Gamma \mid \overline{P}, H \vdash B}}{\Gamma \mid \overline{P}, H \vdash \blacktriangleright B}\ \textbf{(Next)}$$
$$\vdots$$

In all the other cases, the trees can be combined directly.

Example 45. Let P denote the program consisting of clause (1).

$$\forall x. \forall t. \textbf{from}\ (s\ x)\ t \to \textbf{from}\ x\ (\text{scons}\ x\ t) \tag{1}$$

The term-form ground instance of clause (1) due to substitution $[c/x\,,\,(fr\_str\ (s\ c))/t]$ on $\{c : \iota\} \cup \Sigma$ (where $c : \iota \notin \Sigma$) , is as clause (2).

$$\textbf{from}\ (s\ c)\ (fr\_str\ (s\ c)) \to \textbf{from}\ c\ (\text{scons}\ c\ (fr\_str\ (s\ c))) \tag{2}$$

The CUP Proof The *co-hohh*proof for $\Sigma; P \looparrowright \forall x. \textbf{from}\ x\ (fr\_str\ x)$ is given in Fig. 10.

Translating to Löb Proof The guarding of clause (1) is given as clauses (3).

$$\forall x. \forall t.\ \blacktriangleright(\textbf{from}\ (s\ x)\ t) \to \textbf{from}\ x\ (\text{scons}\ x\ t) \tag{3}$$

Note that to save space, when we build a proof in $\textbf{iFOL}_\blacktriangleright$ using **(∀-I)**, **(∀-E)** or **(Conv)**, etc., we may omit printing the condition branch, which is $x : \tau \notin \Gamma$, $\Gamma \vdash M : \tau$ or $\psi \equiv \psi'$ respectively, if and only if we know that the condition holds.

Now let $\overline{P}$ denote the singleton set of clause (3). In Fig. 11 we display the $\textbf{iFOL}_\blacktriangleright$ proofs for $\overline{P} \vdash \forall x. \textbf{from}\ x\ (fr\_str\ x)$ that arises from the CUP proof.

C Proofs for Section 5 (Herbrand Models and Soundness)

Lemma 22. *There is a map* $[\![-]\!]_1 \colon \Lambda_\Sigma^{G,1}(\Gamma) \to \Sigma^\infty(\Gamma)$ *that is unique with*

1. *if* $M \equiv N$*, then* $[\![M]\!]_1 = [\![N]\!]_1$*, and*
2. $\mathrm{root}([\![M]\!]_1) = f\big(\overrightarrow{[\![N]\!]_1}\big)$ *for all* M *with* $M \longrightarrow\!\!\!\!\!\rightarrow f\ \overrightarrow{N}$*, and* $\mathrm{root}([\![M]\!]_1) = x$ *for all* M *with* $M \longrightarrow\!\!\!\!\!\rightarrow x$*,*

Proof. We define a coalgebra $c \colon \Lambda_\Sigma^{G,1}(\Gamma)/\!\equiv\ \to [\![\Sigma]\!]\Big(\Lambda_\Sigma^{G,1}(\Gamma)/\!\equiv\Big) + \Gamma$ on the quotient of guarded terms by convertibility as follows.

$$c[M] = \begin{cases} f[\overrightarrow{N}], & \text{if } M \longrightarrow\!\!\!\!\!\rightarrow f\ \overrightarrow{N} \\ x, & \text{if } M \longrightarrow\!\!\!\!\!\rightarrow x \end{cases}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{c, \Sigma; P; M \xRightarrow{\textbf{from}\ c\ (\text{scons}\ c\ (fr\_str\ (s\ c)))} \textbf{from}\ c\ (fr\_str\ c)}\ \textsc{initial}\checkmark \qquad \spadesuit}{c, \Sigma; P; M \overset{(2)}{\Longrightarrow} \textbf{from}\ c\ (fr\_str\ c)}\ \to L}{c, \Sigma; P; M \overset{(1)}{\Longrightarrow} \textbf{from}\ c\ (fr\_str\ c)}\ \forall L\ (2\ \text{times})}{c, \Sigma; P; M \Longrightarrow \langle \textbf{from}\ c\ (fr\_str\ c) \rangle}\ \textsc{decide}\langle\rangle}{\Sigma; P; M \Longrightarrow \langle \forall x.\, \textbf{from}\ x\ (fr\_str\ x) \rangle}\ \forall R\langle\rangle}{\Sigma; P \looparrowright \forall x.\, \textbf{from}\ x\ (fr\_str\ x)}\ \textsc{co-fix}$$

. . . . ♠ . . . .

$$\dfrac{\dfrac{\dfrac{}{c, \Sigma; P; M \xRightarrow{\textbf{from}\ (s\ c)\ (fr\_str\ (s\ c))} \textbf{from}\ (s\ c)\ (fr\_str\ (s\ c))}\ \textsc{initial}}{c, \Sigma; P; M \overset{M}{\Longrightarrow} \textbf{from}\ (s\ c)\ (fr\_str\ (s\ c))}\ \forall L}{c, \Sigma; P; M \Longrightarrow \textbf{from}\ (s\ c)\ (fr\_str\ (s\ c))}\ \textsc{decide}$$

Fig. 10. The *co-hohh*proof for Example 45. Note that *fr_str* is defined in Example, c is an arbitrary eigenvariable, M abbreviates the coinductive hypothesis $\forall x.\, \textbf{from}\ x\ (fr\_str\ x)$, and the step marked by ✓ indicates involvement of the relation $\textbf{from}\ c\ (\text{scons}\ c\ (fr\_str\ (s\ c))) \equiv \textbf{from}\ c\ (fr\_str\ c)$. The last $\forall L$ step involves the substitution $[(s\ c)/x]$.

This is a well-defined map by Lem. 8. By finality of $\Sigma^\infty(\Gamma)$, we obtain a unique homomorphism $h \colon \Lambda^{G,1}_{\Sigma}(\Gamma)/_{\equiv} \to \Sigma^\infty(\Gamma)$. This allows us to define $[\![-]\!]_1 = h \circ [-]$, which gives us immediately for $M \equiv N$ that $[\![M]\!]_1 = h[M] = h[N] = [\![N]\!]_1$. Moreover, we have

$$\begin{aligned} \text{root}([\![M]\!]_1) &= \text{root}(h[M]) \\ &= ([\![\Sigma]\!](h) + \text{id})(c[M]) \\ &= \begin{cases} [\![\Sigma]\!](h)(f\ [\overrightarrow{N}]) = f\ \overrightarrow{h[N]} = f\ \overrightarrow{[\![N]\!]_1}, & \text{if } M \longrightarrow\!\!\!\!\!\rightarrow f\ \overrightarrow{N} \\ \text{id}(x) = x, & \text{if } M \longrightarrow\!\!\!\!\!\rightarrow x \end{cases} \end{aligned}$$

Finally, assume that we are given a map $k \colon \Lambda^{G,1}_{\Sigma}(\Gamma) \to \Sigma^\infty(\Gamma)$ with the above two properties. The first allows us to lift k to a map $k' \colon \Lambda^{G,1}_{\Sigma}(\Gamma)/_{\equiv} \to \Sigma^\infty(\Gamma)$ with $k' \circ [-] = k$. Due to the second property we know that k' is then a coalgebra homomorphism and by finality $k' = h$. Hence, we obtain from $[\![-]\!]_1 = h \circ [-] = k' \circ [-] = k$ that $[\![-]\!]_1$ is unique.

Let us illustrate the semantics of guarded terms on our running example.

Example 46. Recall s_{fr} from Ex. 6 and note that $s_{\text{fr}}\ 0 \longrightarrow\!\!\!\!\!\rightarrow \text{scons}\ 0\ (s_{\text{fr}}\ (s\ 0))$. Hence, we have $\text{root}([\![s_{\text{fr}}\ 0]\!]_1) = \text{scons}\ [\![0]\!]_1\ [\![s_{\text{fr}}\ (s\ 0)]\!]_1$. If we continue unfolding $[\![s_{\text{fr}}\ 0]\!]_1$, then we obtain the infinite tree $\text{scons}\ 0 \longrightarrow \text{scons}\ (s\ 0) \longrightarrow \text{scons}\ (s\ (s\ 0)) \longrightarrow \cdots$.

Remark 47. It should be noted that we give in the following an interpretation over concrete fibrations with their base over **Set**. However, the interpretations

$$\frac{\dfrac{\overline{x \mid \Delta \vdash \forall x. \forall t.\ \blacktriangleright \mathbf{from}\ (s\ x)\ t \to \mathbf{from}\ x\ (\text{scons}\ x\ t)}\ \textbf{(Proj)}}{x \mid \Delta \vdash \blacktriangleright \mathbf{from}\ (s\ x)\ (\mathit{fr\_str}\ (s\ x)) \to \mathbf{from}\ x\ (\text{scons}\ c\ (\mathit{fr\_str}\ (s\ x)))}\ (\forall\textbf{-E}),\ (\forall\textbf{-E}) \qquad \spadesuit}{x \mid \Delta \vdash \mathbf{from}\ x\ (\text{scons}\ x\ (\mathit{fr\_str}\ (s\ x)))}\ (\to\textbf{-E})$$

$$\frac{x \mid \Delta \vdash \mathbf{from}\ x\ (\text{scons}\ x\ (\mathit{fr\_str}\ (s\ x)))}{x \mid \Delta \vdash \mathbf{from}\ x\ (\mathit{fr\_str}\ x)}\ \textbf{(Conv)}$$

$$\frac{x \mid \Delta \vdash \mathbf{from}\ x\ (\mathit{fr\_str}\ x)}{\overline{P}, \forall x.\ \blacktriangleright \mathbf{from}\ x\ (\mathit{fr\_str}\ x) \vdash \forall x. \mathbf{from}\ x\ (\mathit{fr\_str}\ x)}\ (\forall\textbf{-I})$$

$$\frac{\overline{P}, \forall x.\ \blacktriangleright \mathbf{from}\ x\ (\mathit{fr\_str}\ x) \vdash \forall x. \mathbf{from}\ x\ (\mathit{fr\_str}\ x)}{\overline{P}, \blacktriangleright(\forall x. \mathbf{from}\ x\ (\mathit{fr\_str}\ x)) \vdash \forall x. \mathbf{from}\ x\ (\mathit{fr\_str}\ x)}\ \textbf{(}\blacktriangleright\textbf{-Pres-}\forall_l\textbf{)}$$

$$\frac{\overline{P}, \blacktriangleright(\forall x. \mathbf{from}\ x\ (\mathit{fr\_str}\ x)) \vdash \forall x. \mathbf{from}\ x\ (\mathit{fr\_str}\ x)}{\overline{P} \vdash \forall x. \mathbf{from}\ x\ (\mathit{fr\_str}\ x)}\ \textbf{(Löb)}$$

. . . . $\spadesuit$ . . . .

$$\frac{\dfrac{}{x \mid \Delta \vdash \forall x.\ \blacktriangleright(\mathbf{from}\ x\ (\mathit{fr\_str}\ x))}\ \textbf{(Proj)}}{x \mid \Delta \vdash \blacktriangleright \mathbf{from}\ (s\ x)\ (\mathit{fr\_str}\ (s\ x))}\ (\forall\textbf{-E})$$

Fig. 11. $\mathbf{iFOL}_{\blacktriangleright}$ proof for Example 45. Δ abbreviates $\overline{P}, \forall x.\ \blacktriangleright(\mathbf{from}\ x\ (\mathit{fr\_str}\ x))$.

could also be given over general first-order λ-fibrations $p \colon \mathbf{E} \to \mathbf{B}$. The main issues is to get an interpretation of guarded terms over a final coalgebra for $[\![\Sigma]\!]$ in an general category $\mathbf{B}$. Currently, this interpretation crucially requires the category of sets as base category, see Lem. 22.

Lemma 26. *The mapping $[\![-]\!]$ is a well-defined function from guarded terms to functions, such that $\Gamma \vdash M : \tau$ implies $[\![M]\!] \colon [\![\Gamma]\!] \to [\![\tau]\!]$.*

Proof. Immediate by induction on M.

Lemma 28. *The mapping $[\![-]\!]$ is a well-defined function from formulae to predicates, such that $\Gamma \Vdash \varphi$ implies $[\![\varphi]\!] \subseteq [\![\Gamma]\!]$ or, equivalently, $[\![\varphi]\!] \in \mathbf{Pred}_{[\![\Gamma]\!]}$.*

Proof. Immediate by induction on φ.

Let us demonstrate the interpretation of formulae on an example.

Example 48. Recall the formula $\forall x\, y. \mathbf{from}\ (s\ x)\ y \to \mathbf{from}\ x\ (\mathit{scons}\ x\ y)$, which we introduced as clause $\kappa_{\mathbf{from0}}$. We spell out the interpretation of this formula. Note that $\text{root}([\![s\,x]\!]) = s\,[\![x]\!] = s\,x$. Abusing notation, we write $s\,u$ for $[\![s\,x]\!]\,[u/x]$, and analogously for the terms y, x and $\mathit{scons}\,x\,y$. We then have

$$\begin{aligned} [\![\mathbf{from}\ (s\,x)\ y]\!]_I &= ([\![s\,x]\!], [\![y]\!])^*(I(\mathbf{from})) \\ &= \{(u,v) \in (\Sigma^\infty)^2 \mid (s\,u, v) \in I(\mathbf{from})\} \end{aligned}$$

Using similar calculations for the other terms in the clause $\kappa_{\mathbf{from0}}$, we obtain

$$\begin{aligned} [\![\kappa_{\mathbf{from0}}]\!]_I &= [\![\forall x\, y. \mathbf{from}\ (s\,x)\ y \to \mathbf{from}\ x\ (\mathit{scons}\,x\,y)]\!]_I \\ &= \forall_1 \forall_2 \left([\![\mathbf{from}\ (s\,x)\ y]\!]_I \Rightarrow [\![\mathbf{from}\ x\ (\mathit{scons}\,x\,y)]\!]_I\right) \\ &= \{* \mid \forall u, v.\ \text{if}\ (s\,u, v) \in I(\mathbf{from}),\ \text{then}\ (u, \mathit{scons}\,u\,v) \in I(\mathbf{from})\} \end{aligned}$$

As expected, we thus have $[\![\kappa_{\mathbf{from0}}]\!]_I = \{*\}$ if I validates $\kappa_{\mathbf{from0}}$.

Theorem 34. *Let φ be an H^g-formula of the shape $\forall \vec{x}.\, \psi_1 \to \psi_2$ and $\vec{M} \in \left(\Lambda_{\Sigma}^{G,1}\right)^{|\vec{x}|}$. If $\Sigma; P \looparrowright \varphi$ and $\left[\!\left[\psi_1\left[\vec{M}/\vec{x}\right]\right]\!\right]_{\mathcal{M}_P} = \top$, then $\mathcal{M}_{P \cup \left\{\psi_2\left[\vec{M}/\vec{x}\right]\right\}} = \mathcal{M}_P$. Hence, $P \cup \left\{\psi_2\left[\vec{M}/\vec{x}\right]\right\}$ is a conservative extension of P with respect to the complete Herbrand model.*

Proof. Let $\mathcal{M} = \mathcal{M}_P$ and $\mathcal{M}' = \mathcal{M}_{P \cup \left\{\psi_2\left[\vec{M}/\vec{x}\right]\right\}}$. One first shows that $\Phi_{P \cup Q} = \Phi_P \sqcup \Phi_Q$ for any set Q of ground atoms. The direction $\mathcal{M} \subseteq \mathcal{M}'$ follows easily from this by coinduction. For the other direction, one uses soundness and coinduction again.

Lemma 36. *The mapping $[\![-]\!]$ is a well-defined map from formulae in* **iFOL**$_\blacktriangleright$ *to sequences of predicates, such that $\Gamma \Vdash \varphi$ implies $[\![\varphi]\!] \in \overline{\mathbf{Pred}}_{[\![\Gamma]\!]}$.*

Proof. Immediate by induction on φ.

Lemma 37. *All rules of* **iFOL**$_\blacktriangleright$ *are sound with respect to the interpretation $[\![-]\!]_u$ of formulae in $\overline{\mathbf{Pred}}$, that is, if $\Gamma \mid \Delta \vdash \varphi$, then $\left(\bigwedge_{\psi \in \Delta} [\![\psi]\!]_u \mathrel{\dot\Rightarrow} [\![\varphi]\!]_u\right) = \dot\top$. In particular, $\Gamma \vdash \varphi$ implies $[\![\varphi]\!]_u = \dot\top$.*

Proof. The soundness for the rules of first-order logic in Fig. 8 is standard for the given interpretation over a first-order fibration as in Thm. 35, see [41, Sec. 4.3]. Soundness of the rules for the rules of the later modality in Fig. 9 follows from the existence of the morphisms next and löb, and functoriality of $\blacktriangleright$ that were proved in Thm. 35, cf. [13, Sec. 5.2] and [14].

Lemma 38. *If φ is an H-formula in P, then $[\![\overline{\varphi}]\!]_{\overleftarrow{\Phi}} = \dot\top$.*

Proof. Let φ be a Horn clause in P of shape $\forall \vec{x} : \vec{\tau}.\ \bigwedge_{i=1,\dots,n} p_i\ \vec{M}_i \to q\ \vec{N}$ and let Γ be the context $\vec{x} : \vec{\tau}$. Our goal is to show that $[\![\overline{\varphi}]\!]_{\overleftarrow{\Phi}} = \dot\top$. First, we have by definition of the semantics for all $\alpha \in \mathbf{Ord}$ that

$$
\begin{aligned}
&[\![\textstyle\bigwedge_{i=1,\dots,n} \blacktriangleright(p_i\ \vec{M}_i) \to q\ \vec{N}]\!]_\alpha \\
&= \textstyle\bigcap_{\beta<\alpha} [\![\bigwedge_{i=1,\dots,n} \blacktriangleright(p_i\ \vec{M}_i)]\!]_\beta \Rightarrow [\![q\ \vec{N}]\!]_\beta \\
&= \textstyle\bigcap_{\beta<\alpha} \left(\bigcap_{i=1,\dots,n} [\![\blacktriangleright(p_i\ \vec{M}_i)]\!]_\beta\right) \Rightarrow [\![q\ \vec{N}]\!]_\beta \\
&= \textstyle\bigcap_{\beta<\alpha} \left(\bigcap_{i=1,\dots,n} \bigcap_{\gamma<\beta} [\![p_i\ \vec{M}_i]\!]_\gamma\right) \Rightarrow [\![q\ \vec{N}]\!]_\beta \\
&= \textstyle\bigcap_{\beta<\alpha} \left\{\sigma \in [\![\Gamma]\!] \;\middle|\; \left(\forall i.\, \forall \gamma < \beta.\, \vec{M}_i[\sigma] \in \overleftarrow{\Phi}_\gamma(p)\right)\right. \\
&\qquad\qquad\qquad \left. \Longrightarrow\ \vec{N}[\sigma] \in \overleftarrow{\Phi}_\beta(q)\right\} \\
&= \textstyle\bigcap_{\beta<\alpha} \left\{\sigma \in [\![\Gamma]\!] \;\middle|\; \left(\forall i.\, \forall \gamma < \beta.\, \vec{M}_i[\sigma] \in \overleftarrow{\Phi}_\gamma(p)\right)\right. \\
&\qquad\qquad\qquad \left. \Longrightarrow\ \left(\forall \gamma < \beta.\, \vec{N}[\sigma] \in \Phi\big(\overleftarrow{\Phi}\big)_\gamma(q)\right)\right\}
\end{aligned}
$$

We intent to show now that this set is equal to $\top_{[\![\Gamma]\!]}$. Let $\sigma \in [\![\Gamma]\!]$, such that $\forall i. \forall \gamma < \beta. \overrightarrow{M}_i[\sigma] \in \overleftarrow{\Phi}_\gamma(p)$. We have to show that $\forall \gamma < \beta. \overrightarrow{N}[\sigma] \in \Phi\big(\overleftarrow{\Phi}\big)_\gamma(q)$. To this end, suppose $\gamma < \beta$. Then $\forall i. \overrightarrow{M}_i[\sigma] \in \overleftarrow{\Phi}_\gamma(p)$ by assumption. By definition of Φ we obtain $\overrightarrow{N}[\sigma] \in \Phi\big(\overleftarrow{\Phi}\big)_\gamma(q)$ as required. Hence, $[\![\bigwedge_{i=1,\dots,n} \blacktriangleright(p_i\ \overrightarrow{M}_i) \to q\ \overrightarrow{N}]\!] = \dot{\top}_{[\![\Gamma]\!]}$. But then $[\![\overline{\varphi}]\!]_{\overleftarrow{\Phi}} = \forall_\Gamma \dot{\top}_{[\![\Gamma]\!]} = \dot{\top}$.

Theorem 39. *For all logic programs P, if $\Gamma \mid \overline{P} \vdash \varphi$ then $[\![\varphi]\!]_{\overleftarrow{\Phi}} = \dot{\top}$.*

Proof. Combine Lem. 37 and Lem. 38.

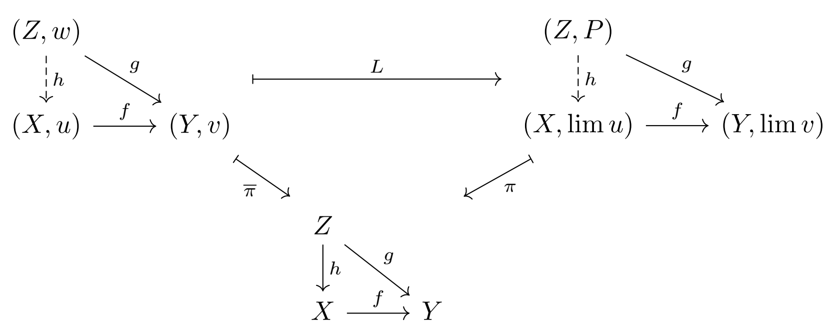
Proposition 41. *$L\colon \overline{\mathbf{Pred}} \to \mathbf{Pred}$, as defined above, is a map of first-order fibrations. Furthermore, L is right-adjoint to the embedding $K\colon \mathbf{Pred} \to \overline{\mathbf{Pred}}$. Finally, for each $p \in \Pi$ and $u \in \overline{\mathbf{Pred}}_{\mathcal{B}^\infty}$, we have $L(u(p)) = L(u)(p)$.*

Proof. First, we show that if $f\colon (X,u) \to (Y,v)$, then f is indeed a morphism $(X, \lim u) \to (Y, \lim v)$. This means that we have to show that $f(\lim u) \subseteq \lim v$. By the limit property, it suffices to show for all $\alpha \in \mathbf{Ord}$ that $f(\lim u) \subseteq v_\alpha$:

$$\begin{aligned} f(\lim u) &\subseteq f(u_\alpha) && \lim u \subseteq u_\alpha \text{ and image of } f \text{ monotone} \\ &\subseteq v_\alpha && f \text{ is morphism } (X,u) \to (Y,v) \end{aligned}$$

That L preserves identities and composition is evident, as is the preservation if indices: $\overline{\pi} = \pi \circ L$.

Next, we show that Cartesian morphism are preserved as well. Let $f\colon (X,u) \to (Y,v)$ be Cartesian in $\overline{\mathbf{Pred}}$, and suppose we are given g and h as in the lower triangle in the following diagram in $\mathbf{Set}$ and (Z,P) in $\mathbf{Pred}$.



We have to show that h is a morphism $(Z,P) \to (X, \lim u)$ in $\mathbf{Pred}$. To that end, we define a constant chain $w\colon \mathbf{Ord} \to \mathbf{Pred}_Z$ by $w_\alpha = P$. Note that $\lim w = P$, thus $L(Z,w) = (Z,P)$. Moreover, for all $\alpha \in \mathbf{Ord}$ we have that $g(w_\alpha) = g(Z) \subseteq \lim v$. Thus, $g(w_\alpha) \subseteq v_\alpha$ and g is a morphism in $\overline{\mathbf{Pred}}$. Since f is Cartesian, we obtain that h is a morphism $(Z,w) \to (X,u)$ in $\overline{\mathbf{Pred}}$, that is, for all α, $h(P) = h(w_\alpha) \subseteq u_\alpha$. This gives us in turn that $h(P) \subseteq \lim u$, which means that h is a morphism $(Z,P) \to (Y, \lim v)$ in $\mathbf{Pred}$.

Showing that L preserves the first-order structure is merely an exercise in patience. For all cases, but the disjunction case, one uses the abstract characterisation of limits of descending chains. Since the disjunction case is somewhat nasty, we demonstrate it here. That is, we want to prove that $\lim(u \mathbin{\dot\vee} v) = \lim u \vee \lim v$. We note now that, because u and v are descending, that there are ordinals α, β, γ such that $\lim(u \mathbin{\dot\vee} v) = (u \mathbin{\dot\vee} v)_\gamma$ and $\lim u \vee \lim v = u_\alpha \vee v_\beta$. Let now $\gamma' = \alpha \sqcup \beta \sqcup \gamma$ be the maximum of these ordinals. Then we have by the above assumptions that

$$\begin{aligned} \lim(u \mathbin{\dot\vee} v) &= (u \mathbin{\dot\vee} v)_\gamma \\ &= (u \mathbin{\dot\vee} v)_{\gamma'} && \text{Descending chains} \\ &= u_{\gamma'} \vee v_{\gamma'} && \text{Point-wise def. of } \dot\vee \\ &= u_\alpha \vee v_\beta && \text{Descending chains} \\ &= \lim u \vee \lim v. \end{aligned}$$

Thus, $L\big((X, u) \mathbin{\dot\vee} (X, v)\big) = L(X, u) \vee L(X, v)$ as desired.

Finally, to show that there is an adjunction $K \dashv L$, we have to show for all $(X, P) \in \mathbf{Pred}$ and $(Y, u) \in \overline{\mathbf{Pred}}$ that there is a natural isomorphism $\mathrm{Hom}_{\mathbf{Pred}}((X, P), (Y, \lim u)) \cong \mathrm{Hom}_{\overline{\mathbf{Pred}}}((X, \overline{P}), (Y, u))$. This boils down to showing that for any map $f\colon X \to Y$ we have $f(P) \subseteq \lim u \iff \forall \alpha.\, f(P) \subseteq u_\alpha$. In turn, this is immediately given by the limit property of $\lim u$.

Theorem 42. *If φ is $\blacktriangleright$-free (Def. 3) then $[\![\varphi]\!]_{\overleftarrow{\Phi}} = \dot\top$ if and only if $[\![\varphi]\!]_{\mathcal{M}_P} = \top$.*

Proof. First, we show for all $\blacktriangleright$-free formulae φ that

$$L([\![\varphi]\!]_{\overleftarrow{\Phi}}) = [\![\varphi]\!]_{\mathcal{M}_P} \tag{4}$$

by induction on φ and using Prop. 41 as follows. In the base case, we have that

$$\begin{aligned} L([\![p\,\overrightarrow{M}]\!]_{\overleftarrow{\Phi}}) &= L\Big([\![\overrightarrow{M}]\!]^{\#}\big(\overleftarrow{\Phi}(p)\big)\Big) \\ &= [\![\overrightarrow{M}]\!]^{*}\Big(L\big(\overleftarrow{\Phi}(p)\big)\Big) && L \text{ preserves reindexing} \\ &= [\![\overrightarrow{M}]\!]^{*}\Big(L\big(\overleftarrow{\Phi}\big)(p)\Big) && L \text{ preserves restrictions} \\ &= [\![\overrightarrow{M}]\!]^{*}(\mathcal{M}_P(p)) && \mathcal{M}_P \text{ is limit of } \overleftarrow{\Phi} \\ &= [\![p\,\overrightarrow{M}]\!]_{\mathcal{M}_P}. \end{aligned}$$

The other cases are given easily by using that L preserves the first-order structure (again Prop. 41). Thus, we obtain $L([\![\varphi]\!]_{\overleftarrow{\Phi}}) = [\![\varphi]\!]_{\mathcal{M}_P}$ for any formula φ.

To show that the semantics over $\mathbf{Pred}$ and $\overline{\mathbf{Pred}}$ coincide, that is, that we have the following one-to-one correspondence.

$$\frac{[\![\varphi]\!]_{\overleftarrow{\Phi}} = \dot\top}{[\![\varphi]\!]_{\mathcal{M}_P} = \top}$$

Since any predicate is included in the maximal predicate $\top$, it suffices to show that there is a adjoint correspondence as in

$$\frac{\dot{\top} \;\to\; [\![\varphi]\!]_{\overleftarrow{\Phi}}}{\top \;\to\; [\![\varphi]\!]_{\mathcal{M}_P}}$$

Note that $\dot{\top}$ is given by the embedding $K(\top)$. Using Prop. 41 and (4) we obtain the desired adjoint correspondence as follows.

$$\frac{\dot{\top} = K(\top) \longrightarrow [\![\varphi]\!]_{\overleftarrow{\Phi}} \quad \text{in } \overline{\mathbf{Pred}}}{\top \longrightarrow L\big([\![\varphi]\!]_{\overleftarrow{\Phi}}\big) = [\![\varphi]\!]_{\mathcal{M}_P} \quad \text{in } \mathbf{Pred}}$$

This concludes the proof of soundness and completeness of chains for formulae with respect to the Herbrand model.