# Coinductive Uniform Proofs: An Extended Abstract

Yue Li and Ekaterina Komendantskaya*

Heriot-Watt University, Edinburgh, Scotland, UK
{yl55,ek19}@hw.ac.uk

## 1 Introduction

Automating coinductive reasoning in Horn clause logic is a challenge. The operational semantics of Horn clause logic is given by the SLD-resolution [3]. Given a possibly non-terminating SLD derivation, the problem is to prove automatically that the derivation does not terminate. Consider the following example of a Horn clause that defines an infinite stream of zeros:

$$\forall x \quad zeros\ x \supset zeros\ (scons\ 0\ x)$$

Given a goal *zeros x*, the initial steps of an infinite SLD-derivation are: $zeros\ x \overset{x:=scons\ 0\ x'}{\rightarrow} zeros\ x' \rightarrow \cdots$

In order to coinductively prove that the above derivation does not terminate, we must find an invariant of this potentially infinite derivation. The state of the art is CoLP [5, 2], that unifies (without occurs check) the two subgoals $zeros\ (scons\ 0\ x')$ and $zeros\ x'$, computing the unifier $x' := (scons\ 0\ x')$. The term that solves this equation is a fixed-point of the function $(\lambda y\,.\,scons\ 0\ y)$, denoted as $(fix\ \lambda y\,.\,scons\ 0\ y)$. Then the coinductive invariant found by CoLP is $(zeros\ (fix\ \lambda y\,.\,scons\ 0\ y))$. It is well known that this method can only find invariants in *regular* derivations. When subgoals do not unify, the method fails to produce any proof.

To circumvent this limitation, Fu et al [1] proposed an additional heuristic. For example, given the Horn clause

$$\forall x \quad p\ (s\ x) \supset p\ x$$

it gives rise to an infinite SLD-derivation: $p\ a \rightarrow p\ (s\ a) \rightarrow \cdots$
Fu et al's algorithm can suggest the coinductive invariant $\forall x\ (p\ x)$, which is then proven in a specially suggested calculus. The limitation of this approach is that it works for Horn clauses and derivations that admit resolution steps by term-matching only, i.e. steps not involving substitutions to the initial goal or any of the subgoals. In particular, this method does not work for clauses defining streams, like in our first example.

Of course Horn clause logic is Turing complete, and an automated discovery of coinductive invariants is an undecidable problem in theory and a hard task in practice. However, we claim that a more principled, proof theoretic [4], approach to the problem could help to advance the current state of the art. We propose a meta-theory for categorizing non-terminating SLD-derivations in terms of the abstract language in which their suitable coinductive invariants are formulated. This new meta-theory can be regarded as a unifying proof-theoretic formalization of the two existing but distinct coinductive invariant discovery algorithms. Moreover, it is a pre-requisite for formulating more general algorithms.

The next example, defining an infinite irregular stream $[0, (s\ 0), (s\ (s\ 0)), \ldots]$, cannot be handled by either of the existing approaches.

$$\forall xy \quad from\ (s\ x)\ y \supset from\ x\ (scons\ x\ y)$$

The infinite derivation for it is given by $from\ 0\ y \overset{y:=scons\ 0\ y'}{\rightarrow} from\ (s\ 0)\ y' \rightarrow \cdots$

*The work is supported by EPSRC grant EP/N014758/1.

All three given examples can be given a uniform and proof-theoretic rendering using the proposed meta-theory called *coinductive uniform proof*. It is an extension of the uniform proof theory by Miller et al [4] with two more elements: *fixed-point terms* built using the *fix* primitive, as in $(\mathit{fix}\ \lambda y\,.\,\mathit{scons}\ 0\ y)$, to help denote infinite objects; and a *coinductive inference rule*. The abstract languages by Miller et al are: first-order Horn clause (*fohc*), higher-order Horn clause (*hohc*), first-order hereditary Harrop formula (*fohh*) and higher-order hereditary Harrop formula (*hohh*). We obtain their coinductive extensions and call the respective languages *co-fohc*, *co-hohc*, *co-fohh* and *co-hohh*. Note that the sense of "higher-order" of our languages does *not* concern quantification over predicates, but only concerns quantification over functions, and such quantification only occur in fixed-point terms.

Our proposed theory is sound w.r.t the greatest fixed-point (*gfp*) models [3], which are the greatest sets of all ground atomic formulae that satisfy the given Horn clauses. For example, the *gfp* model of the Horn clause that defines *zeros* contains a single atom $(\mathit{zeros}\ (\mathit{fix}\ \lambda y\,.\,\mathit{scons}\ 0\ y))$, the *gfp* model of the Horn clause that defines $p$ contains all finite atoms of the form $(p\ (s^n\ a))$ for $n \geq 0$, where $s^n$ denotes composition of $s$ with itself for $n$ times, i.e. $\{p\ a,\ p(s\ a),\ p\ (s(s\ a)),\ldots\}$, together with the infinite atom $(p\ (\mathit{fix}\ \lambda y\,.\,s\ y))$.

## 2 Meta-theory: Coinductive Uniform Proofs

Our term system extends simply typed lambda terms (typically $M,N$) by allowing constructs of the form $\mathit{fix}\ \lambda x\,.\,M$, which shall satisfy standard guarding conditions to denote infinite objects. We use $=_{\mathit{fix}\beta}$ for equivalence of two infinite objects (formal details omitted). For example,

$$\mathit{fix}\ \lambda y\,.\,s\ y =_{\mathit{fix}\beta} s(\mathit{fix}\ \lambda y\,.\,s\ y) =_{\mathit{fix}\beta} s(s(\mathit{fix}\ \lambda y\,.\,s\ y)) =_{\mathit{fix}\beta} \cdots$$

The infinite stream $[0,\ (s\ 0),\ s(s\ 0),\ s(s(s\ 0)),\ldots]$ is defined by the higher-order term

$$((\mathit{fix}\ \lambda f n\ .\ \mathit{scons}\ \ n\ \ (f\ \ (s\ n)))\ \ 0)$$

for which we write $\mathit{fr\_str}\ 0$ as a short hand, and which satisfies the following relations

$$\mathit{fr\_str}\ 0 =_{\mathit{fix}\beta} \mathit{scons}\ 0\ \ (\mathit{fr\_str}\,(s\ 0)) =_{\mathit{fix}\beta} \ \ \mathit{scons}\ 0\ \ \big(\mathit{scons}\,(s\ 0)\ \ \big(\mathit{fr\_str}\,\big(s^2\ 0\big)\big)\big) =_{\mathit{fix}\beta} \cdots$$

The rest of syntax specifications follow the uniform proof theory. We use simple types involving the formula type $o$, and terms are built using constants from a signature $\Sigma$ and variables from the countably infinite set *Var*. An atomic formula $B : o$ has the form $(h\ N_1\ \ldots\ N_n)$ where $h$ is either a constant different from $\wedge, \vee, \forall_\tau, \exists_\tau$ and $\supset$, or a variable; $B$ is *rigid* (respectively, *flexible*) if $h$ is a constant (respectively, variable). A term is *closed* if it does not have free variables. We use $\equiv$ for syntactical identity modulo $\alpha-$equivalence, $=_\beta$ for $\beta-$equivalence. We define $\mathscr{U}_1^\Sigma$ as the set of all terms over $\Sigma$ that do *not* contain $\forall_\tau$ and $\supset$, and $\mathscr{U}_2^\Sigma$ as the set of all terms over $\Sigma$ that do *not* contain $\supset$. Table 1 defines, for each of the four languages, the set $D$ of *program clauses* and the set $G$ of *goals*. Given a signature $\Sigma$, a *program* $P$ is a finite set of *closed* $D$-formulae over $\Sigma$.

We have two kinds of sequents. One kind of sequents are in the form $\Sigma; P \longrightarrow G$, encoding the proposition that the closed goal formula $G$ is provable in intuitionistic logic from the program $P$ on $\Sigma$. We use Miller et al's uniform proof rules (with slight extension to support the $=_{\mathit{fix}\beta}$ relation, see Figure 1) to prove sequents of this kind. We are interested in proving the other kind of sequents, which are in the form $\Sigma; P \looparrowright G$, encoding that the closed goal formula $G$ is *coinductively* provable from the program $P$ on $\Sigma$.

Proving sequents on $\looparrowright$ is closely related to proving sequents on $\longrightarrow$, and for this point we give both formal and informal explanations. Informally, consider the scenario where we begin with proving

$\Sigma; P \rightarrowtail G$, which amounts to prove $\Sigma; P, G \longrightarrow G$ next, but the way we can apply inference rules to prove $\Sigma; P, G \longrightarrow G$ is more *restricted*, compared to a related but different scenario in which we begin with proving $\Sigma; P, G \longrightarrow G$. The motivation for such restriction is to ensure consistency, i.e. to avoid erroneously making arbitrary formulae coinductively provable. Formally, we use the CO-FIX rule (Figure 2) for sequents on $\rightarrowtail$, and we introduce the notation $\langle\rangle$ in the CO-FIX rule, so that a formula marked with $\langle\rangle$ is *guarded*[1] and a sequent with guarded formulae shall be reduced using rules in Figure 3, which encodes the restriction we mentioned in the earlier informal account.

A (coinductive uniform) *proof* is a finite tree such that the root is labeled with $\Sigma; P \rightarrowtail M$, and leaves are labeled with *initial sequents* which are sequents that can occur as a lower sequent in the rules INITIAL or INITIAL$\langle\rangle$. A proof is constructed in *co-fohc* if all formulae in the proof satisfy the language syntax of *co-fohc*. Proofs constructed in *co-fohh*, *co-hohc*, or *co-hohh* are defined similarly.

| | Program Clauses | Goals |
|---|---|---|
| *co-fohc* | $D ::= A^1 \mid G \supset D \mid D \wedge D \mid \forall Var\ D$ | $G ::= \top \mid A^1 \mid G \wedge G \mid G \vee G \mid \exists Var\ G$ |
| *co-hohc* | $D ::= A_r \mid G \supset D \mid D \wedge D \mid \forall Var\ D$ | $G ::= \top \mid A \mid G \wedge G \mid G \vee G \mid \exists Var\ G$ |
| *co-fohh* | $D ::= A^1 \mid G \supset D \mid D \wedge D \mid \forall Var\ D$ | $G ::= \top \mid A^1 \mid G \wedge G \mid G \vee G \mid \exists Var\ G \mid D \supset G \mid \forall Var\ G$ |
| *co-hohh* | $D ::= A_r \mid G \supset D \mid D \wedge D \mid \forall Var\ D$ | $G ::= \top \mid A \mid G \wedge G \mid G \vee G \mid \exists Var\ G \mid D \supset G \mid \forall Var\ G$ |

Table 1: **D- and G-formulae**. $A$ and $A_r$ denote atoms and rigid atoms, respectively. $A^1$ denote first-order atoms. In the setting of *co-hohc*, $A$ and $A_r$ are from $\mathscr{U}_1^\Sigma$; in the setting of *co-hohh*, $A$ and $A_r$ are from $\mathscr{U}_2^\Sigma$.

$$\frac{\Sigma; P, D \longrightarrow G}{\Sigma; P \longrightarrow D \supset G}\ \supset R \qquad \frac{c : \tau, \Sigma; P \longrightarrow G[x := c]}{\Sigma; P \longrightarrow \forall_\tau x\ G}\ \forall R \qquad \frac{\Sigma; P \longrightarrow G[x := N]}{\Sigma; P \longrightarrow \exists_\tau x\ G}\ \exists R$$

$$\frac{\Sigma; P \longrightarrow G_1}{\Sigma; P \longrightarrow G_1 \vee G_2}\ \vee R \quad \frac{\Sigma; P \longrightarrow G_2}{\Sigma; P \longrightarrow G_1 \vee G_2}\ \vee R \qquad \frac{\Sigma; P \longrightarrow G_1 \quad \Sigma; P \longrightarrow G_2}{\Sigma; P \longrightarrow G_1 \wedge G_2}\ \wedge R$$

$$\frac{\Sigma; P \xrightarrow{D} A \quad \Sigma; P \longrightarrow G}{\Sigma; P \xrightarrow{G \supset D} A}\ \supset L \qquad \frac{\Sigma; P \xrightarrow{D_1} A}{\Sigma; P \xrightarrow{D_1 \wedge D_2} A}\ \wedge L \quad \frac{\Sigma; P \xrightarrow{D_2} A}{\Sigma; P \xrightarrow{D_1 \wedge D_2} A}\ \wedge L \qquad \frac{\Sigma; P \xrightarrow{D[x:=N]} A}{\Sigma; P \xrightarrow{\forall_\tau x\ D} A}\ \forall L$$

$$\frac{\Sigma; P \xrightarrow{D} A}{\Sigma; P \longrightarrow A}\ \text{DECIDE} \qquad \frac{}{\Sigma; P \xrightarrow{A'} A}\ \text{INITIAL} \qquad \frac{}{\Sigma; P \longrightarrow \top}\ \top R$$

Figure 1: **Uniform proof rules**. *Rule restrictions:* in $\exists R$ and $\forall L$, $N : \tau$ is a closed term on $\Sigma$. Moreover, if used in *co-fohc* or *co-fohh*, then $N$ is first order; if used in *co-hohc*, then $N \in \mathscr{U}_1^\Sigma$; if used in *co-hohh*, then $N \in \mathscr{U}_2^\Sigma$. In $\forall R$, $c : \tau \notin \Sigma$ ($c$ is also known as an *eigenvariable*). In DECIDE, $D \in P$. In the rule INITIAL, $A =_{fix\beta} A'$.

$$\frac{\Sigma; P, \langle M \rangle \longrightarrow \langle M \rangle}{\Sigma; P \rightarrowtail M}\ \text{CO-FIX} \qquad \begin{array}{llll} \textit{co-fohc} & M := A^1 \mid M \wedge M & \textit{co-fohh} & M := A^1 \mid M \wedge M \mid M \supset M \mid \forall Var\ M \\ \textit{co-hohc} & M := A_r \mid M \wedge M & \textit{co-hohh} & M := A_r \mid M \wedge M \mid M \supset M \mid \forall Var\ M \end{array}$$

Figure 2: **The coinductive fixed-point rule and syntax for core formulae.** *Note:* In the upper sequent of CO-FIX rule, the left occurrence of $M$ is called a *coinductive hypothesis*, and the right occurrence of $M$ is called a *coinductive goal*. The formula $M$ occurs on *both* sides of the upper sequent in the CO-FIX rule, therefore $M$ must satisfy the syntax of both program clauses and goals. Formulae with such syntactic character as $M$ are called *core formulae* [4].

[1] There are two distinct notions of *guard* in coinductive uniform proof: one is for the syntax of fixed-point terms, to ensure that they model infinite objects; the other is for formulae in certain sequents, to ensure consistency.

$$\frac{\Sigma; P, \langle M_1\rangle \longrightarrow \langle M_2\rangle}{\Sigma; P \longrightarrow \langle M_1 \supset M_2\rangle}\ \supset R\langle\rangle \qquad \frac{c:\tau, \Sigma; P \longrightarrow \langle M\,[x := c]\rangle}{\Sigma; P \longrightarrow \langle \forall_\tau x\, M\rangle}\ \forall R\langle\rangle \qquad \frac{\Sigma; P \longrightarrow \langle M_1\rangle \quad \Sigma; P \longrightarrow \langle M_2\rangle}{\Sigma; P \longrightarrow \langle M_1 \wedge M_2\rangle}\ \wedge R\langle\rangle$$

$$\frac{\Sigma; P^* \xrightarrow{D} A \quad \Sigma; P^* \longrightarrow G}{\Sigma; P \xrightarrow{G \supset D} \langle A\rangle}\ \supset L\langle\rangle \qquad \frac{\Sigma; P \xrightarrow{D_1} \langle A\rangle}{\Sigma; P \xrightarrow{D_1 \wedge D_2} \langle A\rangle}\ \wedge L\langle\rangle \qquad \frac{\Sigma; P \xrightarrow{D_2} \langle A\rangle}{\Sigma; P \xrightarrow{D_1 \wedge D_2} \langle A\rangle}\ \wedge L\langle\rangle \qquad \frac{\Sigma; P \xrightarrow{D[x:=N]} \langle A\rangle}{\Sigma; P \xrightarrow{\forall x\, D} \langle A\rangle}\ \forall L\langle\rangle$$

$$\frac{\Sigma; P \xrightarrow{D^*} \langle A\rangle}{\Sigma; P \longrightarrow \langle A\rangle}\ \textsc{decide}\langle\rangle \qquad \frac{}{\Sigma; P \xrightarrow{A'} \langle A\rangle}\ \textsc{initial}\langle\rangle$$

Figure 3: **Rules for guarded coinductive goals.** *Rule restrictions*: In DECIDE$\langle\rangle$, $D^*$ must be a formula without $\langle\rangle$ mark. In $\supset L\langle\rangle$, $P^*$ results from erasing all $\langle\rangle$ marks in $P$. The restrictions for INITIAL$\langle\rangle$, $\forall L\langle\rangle$ and $\forall R\langle\rangle$ are the same as for INITIAL, $\forall L$ and $\forall R$ respectively. *Note:* Formulae added to the left-hand side by CO-FIX and $\supset R\langle\rangle$ are guarded, so that they are not selected by the DECIDE$\langle\rangle$ rule for back-chaining with guarded atomic goals. The $\supset L\langle\rangle$ rule frees all formulae from being guarded for each upper sequent, then rules in Figure 1 become applicable in further sequent reductions.

# 3 Discussion

Using coinductive uniform proofs, we can categorize infinite SLD-derivations, and we can uniformly and proof-theoretically formalize the coinductive reasoning performed by the two algorithms mentioned earlier. For instance, in the CoLP example, we need *co-fohc* to express and prove the coinductive invariant $(zeros\ (fix\ \lambda y\,.\,scons\ 0\ y))$, and the root sequent for the uniform proof is $\Sigma_1; P_1 \looparrowright (zeros\ (fix\ \lambda y\,.\,scons\ 0\ y))$; in the example for Fu et al's algorithm, we need *co-fohh* to express and prove the coinductive invariant $\forall x\ (p\ x)$, and the root sequent is $\Sigma_2; P_2 \looparrowright \forall x\ (p\ x)$; in the third example, we need *co-hohh* to express and prove the coinductive invariant $\forall x\ from\ x\ (fr\_str\ x)$, with the root sequent $\Sigma_3; P_3 \looparrowright \forall x\ from\ x\ (fr\_str\ x)$. The full sequent proof for the last example is given in Appendix A.

We omit technical details of the proof of soundness of coinductive uniform proofs w.r.t the *gfp* models. Intuitively, the proof proceeds by defining a scheme by which we can reconstruct a corresponding non-terminating derivation, and then showing that the proofs are sound w.r.t the *gfp* models. However, in contrast with CoLP, the reconstruction is generally more complicated and involves *(i)* a construction of a function that generates countably many different substitution instances for the derivation scheme, and *(ii)* showing that these instances can be composed in a certain way in order to restore the full infinite derivation. The proof is constructive, and in addition uses a coinductive proof principle when showing correspondence of the derivation schemes to the *gfp* model construction.

The fact that the CO-FIX rule can only be applied once and as the first step in a proof, is a simplification that helps to highlight the basic coinductive argument performed by the coinductive uniform proofs. The absence of nested coinduction in the meta-theory can be mitigated by allowing using the already proven coinductive invariants as lemmas to prove further coinductive conclusions.

# References


[1] P. Fu, E. Komendantskaya, T. Schrijvers, and A. Pond. Proof relevant corecursive resolution. In *FLOPS'16*, pages 126–143. Springer, 2016.

[2] E. Komendantskaya and Y. Li. Productive corecursion in logic programming. *J. TPLP (ICLP'17 post-proc.)*, 17(5-6):906–923, 2017.

[3] J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 2nd edition, 1987.

[4] D. Miller and G. Nadathur. *Programming with Higher-order logic*. Cambridge University Press, 2012.

[5] L. Simon, A. Mallya, A. Bansal, and G. Gupta. Coinductive logic programming. In *ICLP*, pages 330–345, 2006.

# A Detailed proof for the example

We give the *co-hohh* proof[2] for the sequent $\Sigma_3; P_3 \looparrowright \forall x(\mathit{from}\ x\ (\mathit{fr\_str}\ x))$. Note that $fr\_str$ is defined in Section 2, $Z$ is an arbitrary eigenvariable, $CH$ abbreviates the coinductive hypothesis $\forall x(\mathit{from}\ x\ (\mathit{fr\_str}\ x))$, and the step marked by ✓ indicates involvement of the relation

$$\mathit{from}\ Z\ (\mathit{scons}\ Z\ (fr\_str\ (s\ Z))) =_{fix\beta} \mathit{from}\ Z\ (\mathit{fr\_str}\ Z)$$

The two $\forall L\langle\rangle$ steps involve the substitutions $x := Z, y := (fr\_str\ (s\ Z))$. The $\forall L$ step involves the substitution $x := s\ Z$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Z,\Sigma;P,CH \xrightarrow{\mathit{from}\ Z\ (\mathit{scons}\ Z\ (fr\_str\ (s\ Z)))} \mathit{from}\ Z\ (\mathit{fr\_str}\ Z)}\ \text{INITIAL}\checkmark \quad \dfrac{\dfrac{\dfrac{}{Z,\Sigma;P,CH \xrightarrow{\mathit{from}\ (s\ Z)\ (\mathit{fr\_str}\ (s\ Z))} \mathit{from}\ (s\ Z)\ (\mathit{fr\_str}\ (s\ Z))}\ \text{INITIAL}}{Z,\Sigma;P,CH \xrightarrow{CH} \mathit{from}\ (s\ Z)\ (\mathit{fr\_str}\ (s\ Z))}\ \forall L}{Z,\Sigma;P,CH \longrightarrow \mathit{from}\ (s\ Z)\ (\mathit{fr\_str}\ (s\ Z))}\ \text{DECIDE}}{Z,\Sigma;P,\langle CH\rangle \xrightarrow{\mathit{from}\ (s\ Z)(\mathit{fr\_str}\ (s\ Z))\ \supset \mathit{from}\ Z\ (\mathit{scons}\ Z\ (\mathit{fr\_str}\ (s\ Z)))} \langle \mathit{from}\ Z\ (\mathit{fr\_str}\ Z)\rangle}\ \supset L\langle\rangle}{Z,\Sigma;P,\langle CH\rangle \xrightarrow{\forall xy\ \mathit{from}\ (s\ x)\ y \supset \mathit{from}\ x\ (\mathit{scons}\ x\ y)} \langle \mathit{from}\ Z\ (\mathit{fr\_str}\ Z)\rangle}\ \forall L\langle\rangle\ (2\ \text{times})$$

$$\dfrac{\dfrac{Z,\Sigma;P,\langle CH\rangle \longrightarrow \langle \mathit{from}\ Z\ (\mathit{fr\_str}\ Z)\rangle}{\Sigma;P,\langle CH\rangle \longrightarrow \langle \forall x(\mathit{from}\ x\ (\mathit{fr\_str}\ x))\rangle}\ \forall R\langle\rangle}{\Sigma;P \looparrowright \forall x(\mathit{from}\ x\ (\mathit{fr\_str}\ x))}\ \text{DECIDE}\langle\rangle\ \text{CO-FIX}$$

[2] We omit the subscript 3 for $\Sigma, P$ in the proof.