Coinductive First-order Horn Clauses

Yue Li

PhD candidate supervised by Dr. Ekaterina Komendantskaya
School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh

April 2018

A bit history of Yue: from amazing China to bonnie Dundee

- Bachelor in Hunan University, China.
- Master in University of Dundee, United Kingdom.

Overview

1. Introduction: What is coinduction, in logic programming ?
2. Our Key Observation: Inadequacy of Horn clause logic to prove coinductive invariants of first-order Horn clause logic programs

The concept of "coinduction"

The concept of "coinduction"

- Coinduction refers to non-terminating computation.

The concept of "coinduction"

- Coinduction refers to non-terminating computation.
- It is common in logic programming and type inference (eg. Haskell type class resolution) due to recursive definitions.

An example of coinduction in logic programming

An example of coinduction in logic programming

Consider a program P_1:

Clause (1) $\forall X \ (r(X) \supset r(X))$ In Prolog, `r(X) :- r(X)`

We use $\supset$ for implication, read as "implies". `:-` is read as "be implied by".

An example of coinduction in logic programming

Consider a program P_1:

$$\text{Clause (1)} \qquad \forall X\ (r(X) \supset r(X)) \qquad \text{In Prolog, } \texttt{r(X) :- r(X)}$$

We use $\supset$ for implication, read as "implies". `:-` is read as "be implied by".
The goal `?-r(a)` gives rise to an infinite SLD-derivation

$$\texttt{r(a)} \overset{\text{apply clause (1)}}{\longrightarrow} \texttt{r(a)} \overset{\text{apply clause (1)}}{\longrightarrow} \texttt{r(a)} \overset{\text{apply clause (1)}}{\longrightarrow} \cdots$$

where we look for sufficient conditions for $r(a)$ to hold using clause (1).

Overview

1. Introduction: What is coinduction, in logic programming ?
2. Our Key Observation: Inadequacy of Horn clause logic to prove coinductive invariants of first-order Horn clause logic programs

A glimpse of research topics on coinductive logic programming

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,
- Find coinductive invariants for logic programs,

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,
- Find coinductive invariants for logic programs, so that we can have a finite *coinductive proof* instead of an equivalent, infinite SLD-derivation. (Details explained later)

Example of a “coinductive invariant”

Example of a “coinductive invariant”

For program P_1 : Clause (1) $\forall X \ (r(X) \supset r(X))$,
$r(a)$ is a coinductive invariant,

Example of a "coinductive invariant"

For program P_1 : Clause (1) $\forall X \ (r(X) \supset r(X))$,
$r(a)$ is a coinductive invariant, in the sense that if we assume $r(a)$ is true, we can then prove $r(a)$ using the extended program $P_1' = P_1 \cup \{r(a)\}$ given as follows.

Clause (1) $\forall X \ (r(X) \supset r(X))$
Clause (2) $r(a)$

Example of a "coinductive invariant"

For program P_1 : Clause (1) $\forall X \ (r(X) \supset r(X))$,
$r(a)$ is a coinductive invariant, in the sense that if we assume $r(a)$ is true, we can then prove $r(a)$ using the extended program $P_1' = P_1 \cup \{r(a)\}$ given as follows.

Clause (1) $\forall X \ (r(X) \supset r(X))$
Clause (2) $r(a)$

The finite coinductive proof is:

$$\texttt{r(a)} \xrightarrow{\text{apply clause (1)}} \texttt{r(a)} \xrightarrow{\text{apply clause (2)}} \text{Success}$$

Example of a "coinductive invariant"

For program P_1 : Clause (1) $\forall X \ (r(X) \supset r(X))$,
$r(a)$ is a coinductive invariant, in the sense that if we assume $r(a)$ is true, we can then prove $r(a)$ using the extended program $P_1' = P_1 \cup \{r(a)\}$ given as follows.

Clause (1) $\forall X \ (r(X) \supset r(X))$
Clause (2) $r(a)$

The finite coinductive proof is:

$$\texttt{r(a)} \xrightarrow{\text{apply clause (1)}} \texttt{r(a)} \xrightarrow{\text{apply clause (2)}} \text{Success}$$

The coinductive proof above, as well as the corresponding infinite derivation, are both sound, with respect to the greatest fixed point model of P_1.

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,
- Find coinductive invariants for logic programs, so that we can have a finite *coinductive proof* instead of an equivalent, infinite SLD-derivation.

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,
- Find coinductive invariants for logic programs, so that we can have a finite *coinductive proof* instead of an equivalent, infinite SLD-derivation.
- Deal with "unusual" coinductive invariants.

Example of an “unusual” coinductive invariant

Example of an "unusual" coinductive invariant

Consider program P_2:

Clause (3) $\forall X \ (p(s(X)) \supset p(X))$ In Prolog, `p(X) :- p(s(X))`

Example of an "unusual" coinductive invariant

Consider program P_2:

Clause (3) $\forall X \ (p(s(X)) \supset p(X))$ In Prolog, `p(X) :- p(s(X))`

The goal `?-p(a)` gives rise to an infinite SLD-derivation

$$p(a) \xrightarrow{\text{apply clause (3)}} p(s(a)) \xrightarrow{\text{apply clause (3)}} p(s(s(a))) \xrightarrow{\text{apply clause (3)}} \cdots$$

Example of “unusual” coinductive invariant (continued)

For P_2 : Clause (3) $\forall X \ (\, p\,(s\,(X)) \supset p(X)\,)$, the coinductive invariant is $\forall X \ p(X)$,

Example of "unusual" coinductive invariant (continued)

For P_2 : Clause (3) $\forall X \ (\, p\,(s\,(X)) \supset p(X)\,)$, the coinductive invariant is $\forall X \ p(X)$, in the sense that if we assume $\forall X \ p(X)$ is true, then we can prove $\forall X \ p(X)$ from the extended program $P_2' = P_2 \cup \{\forall X \ p(X)\}$:

Clause (3) $\forall X \ (\, p\,(s\,(X)) \supset p(X)\,)$
Clause (4) $\forall X \ p(X)$

Example of “unusual” coinductive invariant (continued)

For P_2 : Clause (3) $\forall X \ (\, p\,(s\,(X)) \supset p(X)\,)$, the coinductive invariant is $\forall X \ p(X)$, in the sense that if we assume $\forall X \ p(X)$ is true, then we can prove $\forall X \ p(X)$ from the extended program $P_2' = P_2 \cup \{\forall X \ p(X)\}$:

Clause (3) $\forall X \ (\, p\,(s\,(X)) \supset p(X)\,)$
Clause (4) $\forall X \ p(X)$

The finite coinductive proof is:

$$\forall X \ p(X) \overset{\text{eigenvariable } c}{\longrightarrow} p(c) \overset{\text{apply clause (3)}}{\longrightarrow} p(s(c)) \overset{\text{apply clause (4)}}{\longrightarrow} \text{Success}$$

which is sound wrt. the greatest fixed point model of P_2.

Example of “unusual” coinductive invariant (continued)

For P_2 : Clause (3) $\forall X \ (p(s(X)) \supset p(X))$, the coinductive invariant is $\forall X \ p(X)$, in the sense that if we assume $\forall X \ p(X)$ is true, then we can prove $\forall X \ p(X)$ from the extended program $P_2' = P_2 \cup \{\forall X \ p(X)\}$:

Clause (3) $\forall X \ (p(s(X)) \supset p(X))$
Clause (4) $\forall X \ p(X)$

The finite coinductive proof is:

$$\forall X \ p(X) \xrightarrow{\text{eigenvariable } c} p(c) \xrightarrow{\text{apply clause (3)}} p(s(c)) \xrightarrow{\text{apply clause (4)}} \text{Success}$$

which is sound wrt. the greatest fixed point model of P_2. We can then derive $p(a)$ as a corollary (recall that $p(a)$ has an infinite SLD-derivation wrt. P_2).

Example of "unusual" coinductive invariant (continued)

For P_2 : Clause (3) $\forall X \ (\, p\,(s\,(X)) \supset p(X)\,)$, the coinductive invariant is $\forall X \ p(X)$, in the sense that if we assume $\forall X \ p(X)$ is true, then we can prove $\forall X \ p(X)$ from the extended program $P_2' = P_2 \cup \{\forall X \ p(X)\}$:

Clause (3) $\forall X \ (\, p\,(s\,(X)) \supset p(X)\,)$
Clause (4) $\forall X \ p(X)$

The finite coinductive proof is:

$$\forall X \ p(X) \overset{\text{eigenvariable } c}{\longrightarrow} p(c) \overset{\text{apply clause (3)}}{\longrightarrow} p(s(c)) \overset{\text{apply clause (4)}}{\longrightarrow} \text{Success}$$

which is sound wrt. the greatest fixed point model of P_2. We can then derive $p(a)$ as a corollary (recall that $p(a)$ has an infinite SLD-derivation wrt. P_2).

The catch:

$\forall X \ p(X)$ is *not* a goal formula allowable by the syntax of Horn clause logic.

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,
- Find coinductive invariants for logic programs, so that we can have a finite *coinductive proof* instead of an equivalent, infinite SLD-derivation.
- Deal with "unusual" coinductive invariants

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,
- Find coinductive invariants for logic programs, so that we can have a finite *coinductive proof* instead of an equivalent, infinite SLD-derivation.
- Deal with "unusual" coinductive invariants which are not provable within Horn clause logic.

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,
- Find coinductive invariants for logic programs, so that we can have a finite *coinductive proof* instead of an equivalent, infinite SLD-derivation.
- Deal with "unusual" coinductive invariants which are not provable within Horn clause logic.
- Search for a logic (which must be richer than Horn clause logic) capable to prove both "usual" and "unusual" coinductive invariants involved in first-order Horn clause logic programming.

A glimpse of research topics on coinductive logic programming

We try to:

- Identify phenomena of coinduction,
- Find coinductive invariants for logic programs, so that we can have a finite *coinductive proof* instead of an equivalent, infinite SLD-derivation.
- Deal with "unusual" coinductive invariants which are not provable within Horn clause logic.
- Search for a logic (which must be richer than Horn clause logic) capable to prove both "usual" and "unusual" coinductive invariants involved in first-order Horn clause logic programming.

Thanks !