Proving Reachability Properties by Coinduction

Dorel Lucanu

(Joint work with Ștefan Ciobâcă)

Department of Computer Science
Alexandru Ioan Cuza University of *Iași*

SYNASC, September 21, 2018

---

1This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI - UEFISCDI, project number PN-III-P2-2.1-BG-2016-0394, within PNCDI III.
Motivation

Induction and Coinduction
- Motivation
- Theoretical Foundation
- (Co)Inductive Sets Defined by Ground Inference Systems

Reachability Predicates

Logical Constrained Rewriting Systems (LCTRSs)

A Coinductive Proof System for Reachability Formulas

Conclusion
Plan

1. Motivation

2. Induction and Coinduction
   - Motivation
   - Theoretical Foundation
   - (Co)Inductive Sets Defined by Ground Inference Systems

3. Reachability Predicates

4. Logical Constrained Rewriting Systems (LCTRSs)

5. A Coinductive Proof System for Reachability Formulas

6. Conclusion
A Specification of a Transition System

\[ \langle n \rangle \rightarrow \langle n, a \rangle \text{ if } \exists y \cdot y = a \wedge 2 \leq y \wedge y < n \]
\[ \langle n, a \rangle \rightarrow \langle \text{composite} \rangle \text{ if } \text{isEullerWitness}(a, n) \]
\[ \langle n, a \rangle \rightarrow \langle n \rangle \text{ if } \neg \text{isEullerWitness}(a, n) \]

where

\[ \text{isEullerWitness}(a, n) \equiv ((a \mid n) = 0 \lor (a^{\frac{n-1}{2}} \neq a \mid n \pmod{n})) \]

\( a \mid n \) is the Jacobi symbol

We want to show that

"if \( \langle n \rangle \) goes to \( \langle \text{composite} \rangle \) then
\( \exists n_1, n_2. \ n = n_1 \cdot n_2 \wedge n_1 > 1 \wedge n_2 > 1 \)."
The First Issue

The computations can be finite:

\[ \langle 4 \rangle \quad \langle 4, 2 \rangle \quad \langle \text{composite} \rangle \]

or infinite:

\[ \langle 5 \rangle \quad \langle 5, 2 \rangle \quad \langle 5, 3 \rangle \quad \langle 5, 4 \rangle \]
The First Issue

The computations can be finite:

or infinite:
The conditions

$$\exists y . y = a \land 2 \leq y \land y < n$$

$$((a \mid n) = 0 \lor (a^{\frac{n-1}{2}} \neq a \mid n \pmod{n}))$$

$$((a \mid n) = 0 \lor (a^{\frac{n-1}{2}} \neq a \mid n \pmod{n})) \rightarrow \exists n_1, n_2. n = n_1 \cdot n_2 \land n_1 > 1 \land n_2 > 1$$

are very complex first order formulas including functions and predicates whose definitions are orthogonal with that of the transition system.
Our Proposal

- we use coinduction in order to handle both finite and infinite computations
- we introduce LCTRSs (Logical Constrained Term Rewriting Systems) for specifying transition systems
- we propose an effective proof system that, given a LCTRS, proves valid reachability formulas, assuming an oracle (e.g., an SMT solver) that solves logical constraints
Our Proposal

- we use **coinduction** in order to handle both finite and infinite computations
- we introduce **LCTRSs (Logical Constrained Term Rewriting Systems)** for specifying transition systems
- we propose an effective **proof system** that, given a LCTRS, proves valid **reachability formulas**, assuming an oracle (e.g., an SMT solver) that solves logical constraints
Our Proposal

- we use coinduction in order to handle both finite and infinite computations
- we introduce LCTRSs (Logical Constrained Term Rewriting Systems) for specifying transition systems
- we propose an effective proof system that, given a LCTRS, proves valid reachability formulas, assuming an oracle (e.g., an SMT solver) that solves logical constraints
Plan

1. Motivation

2. Induction and Coinduction
   - Motivation
   - Theoretical Foundation
   - (Co)Inductive Sets Defined by Ground Inference Systems

3. Reachability Predicates

4. Logical Constrained Rewriting Systems (LCTRSs)

5. A Coinductive Proof System for Reachability Formulas

6. Conclusion
Outline

1 Motivation

2 Induction and Coinduction
   - Motivation
   - Theoretical Foundation
   - (Co)Inductive Sets Defined by Ground Inference Systems

3 Reachability Predicates

4 Logical Constrained Rewriting Systems (LCTRSs)

5 A Coinductive Proof System for Reachability Formulas

6 Conclusion
A Simpler Transition System Specification

\[
\langle a, b \rangle \rightarrow \langle a - b, b \rangle \text{ if } a \geq b
\]

\[
\langle a, b \rangle \rightarrow \langle a, b - a \rangle \text{ if } b \geq a
\]

\[
\langle a, b \rangle \rightarrow \langle a + b \rangle \text{ if } a = 0 \lor b = 0
\]
Two Computations

– finite computations

\[
\langle 6, 4 \rangle \xrightarrow{} \langle 2, 4 \rangle \xrightarrow{} \langle 2, 2 \rangle \xrightarrow{} \langle 2, 0 \rangle \xrightarrow{} \langle 2 \rangle
\]

– infinite computations

\[
\langle 6, -4 \rangle \xrightarrow{} \langle 10, -4 \rangle \xrightarrow{} \langle 14, -4 \rangle \xrightarrow{} \ldots
\]
A Possible Specification of Computations

- **using BNF notation:**
  \[ C ::= \langle a \rangle | \langle a, b \rangle \leadsto C \]

- **equivalently, using inference rules**
  \[
  \begin{array}{c}
  \hline
  \langle a \rangle & \langle a, b \rangle \leadsto C \\
  \hline
  \end{array}
  \]

**Question**

- Are these specifications appropriate?
- I.e., do they uniquely define the computations?

**Answer**

Yes, if

- we consider the **smallest fixed-point** satisfying the rules (finite computations, inductively defined), or
- we consider the **greatest fixed-point** satisfying the rules (finite computations + infinite computations, coinductively defined)
A Possible Specification of Computations

- using BNF notation:
  \[ C ::= \langle a \rangle \mid \langle a, b \rangle \rightarrow C \]

- equivalently, using inference rules
  \[
  \begin{align*}
  & C \\
  \hline
  & \langle a \rangle \\
  & \langle a, b \rangle \rightarrow C \\
  \end{align*}
  \]

Question

- Are these specifications appropriate?
- I.e., do they uniquely define the computations?

Answer

Yes, if

- we consider the smallest fixed-point satisfying the rules (finite computations, inductively defined), or
- we consider the greatest fixed-point satisfying the rules (finite computations + infinite computations, coinductively defined)
A Possible Specification of Computations

- using BNF notation:
  \[
  C ::= \langle a \rangle | \langle a, b \rangle \leadsto C
  \]
- equivalently, using inference rules
  \[
  \begin{align*}
  &\langle a \rangle C \\
  &\langle a, b \rangle \leadsto C
  \end{align*}
  \]

Question

- Are these specifications appropriate?
- I.e., do they uniquely define the computations?

Answer

Yes, if

- we consider the smallest fixed-point satisfying the rules (finite computations, inductively defined), or
- we consider the greatest fixed-point satisfying the rules (finite computations + infinite computations, coinductively defined)
A Possible Specification of Computations

- using BNF notation:
  \[ C ::= \langle a \rangle \mid \langle a, b \rangle \leadsto C \]
- equivalently, using inference rules
  \[
  \begin{align*}
  & \quad C \\
  \quad \langle a \rangle & \quad \langle a, b \rangle \leadsto C
  \end{align*}
  \]

Question

- Are these specifications appropriate?
- I.e., do they uniquely define the computations?

Answer

Yes, if

- we consider the smallest fixed-point satisfying the rules (finite computations, \textit{inductively defined}), or
- we consider the greatest fixed-point satisfying the rules (finite computations + infinite computations, \textit{coinductively defined})
Outline

1 Motivation

2 Induction and Coinduction
   • Motivation
   • Theoretical Foundation
     (Co)Inductive Sets Defined by Ground Inference Systems

3 Reachability Predicates

4 Logical Constrained Rewriting Systems (LCTRSs)

5 A Coinductive Proof System for Reachability Formulas

6 Conclusion
Complete Lattices

- **Partial ordered set (poset):** set $L$ together with a binary relation $\subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- **Complete lattice:** a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- **Notations:**
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- The most known (and used) example: $(\mathcal{P}(X), \subseteq)$, where $X$ is a set
Complete Lattices

- **partial ordered set (poset):** set $L$ together with a binary relation $\subseteq \subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- **complete lattice** = a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- **notations:**
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- the most known (and used) example: $(\mathcal{P}(X), \subseteq)$, where $X$ is a set
Complete Lattices

- **partial ordered set (poset)**: set $L$ together with a binary relation $\sqsubseteq \subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- **complete lattice** = a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- **notations:**
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- the most known (and used) example: $(\mathcal{P}(X), \subseteq)$, where $X$ is a set
**Complete Lattices**

- **partial ordered set (poset)**: set $L$ together with a binary relation $\subseteq \subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- **complete lattice** = a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- **notations**:
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- the most known (and used) example: $(P(X), \subseteq)$, where $X$ is a set
Complete Lattices

- **partial ordered set (poset):** set $L$ together with a binary relation $\subseteq \subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- **complete lattice** = a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- **notations:**
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- the most known (and used) example: $(\mathcal{P}(X), \subseteq)$, where $X$ is a set
Complete Lattices

- **partial ordered set (poset):** set $L$ together with a binary relation $\subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- **complete lattice** = a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- **notations:**
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- the most known (and used) example: $(\mathcal{P}(X), \subseteq)$, where $X$ is a set
Complete Lattices

- **partial ordered set (poset):** set $L$ together with a binary relation $\subseteq \subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- **complete lattice** = a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- **notations:**
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- the most known (and used) example: $(\mathcal{P}(X), \subseteq)$, where $X$ is a set
Complete Lattices

- partial ordered set (poset): set $L$ together with a binary relation $\subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- complete lattice = a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- notations:
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- the most known (and used) example: $(\mathcal{P}(X), \subseteq)$, where $X$ is a set
Complete Lattices

- **partial ordered set (poset):** set $L$ together with a binary relation $\subseteq L \times L$ that is
  - reflexive
  - transitive
  - antisymmetric

- **complete lattice** = a poset with all lubs (least upper bounds, joins), and hence also all glbs (greatest lower bounds, meets)

- **notations:**
  - lub of $x$ and $y$: $x \sqcup y$
  - lub of a set $A$: $\sqcup A$
  - glb of $x$ and $y$: $x \sqcap y$
  - glb of a set $A$: $\sqcap A$
  - $\bot = \sqcap L$
  - $\top = \sqcup L$

- the most known (and used) example: $(\mathcal{P}(X), \subseteq)$, where $X$ is a set
Examples of Lattices

$x \sqsubseteq y$ complete lattice

poset that is not a complete lattice
Fixed Points

Let \((L, \sqsubseteq)\) be a complete lattice and \(f : L \rightarrow L\).

- \(x \in L\) pre-fixed point of \(f\) if \(f(x) \sqsubseteq x\)
- \(x \in L\) post-fixed point of \(f\) if \(x \sqsubseteq f(x)\)
- \(x \in L\) fixed point of \(f\) if \(f(x) = x\)

Terminology:

- pre-fixed point, \(f\)-forward-closed, \(f\)-closed
- post-fixed point, \(f\)-backward-closed, \(f\)-stable, \(f\)-consistent
Knaster-Tarski Theorem 1/2

Let \((L, \sqsubseteq)\) be a complete lattice.
\(f : L \to L\) is monotone if \(f(x) \sqsubseteq f(y)\) whenever \(x \sqsubseteq y\).

**Theorem (Knaster-Tarski)**

Any \(f : L \to L\) monotone has
- a **least fixed point** \(\mu y. f(y)\) (on short \(\mu f\)), and
- a **greatest fixed point** \(\nu y. f(y)\) (on short \(\nu f\)).

Moreover,
- \(\mu f = \bigcap\{x \mid f(x) \sqsubseteq X\}\) and
- \(\nu f = \bigcup\{x \mid x \sqsubseteq f(x)\}\)

I.e.,
- \(\mu f\) is the meet of pre-fixed points
- \(\nu f\) is the join of post-fixed points
Knaster-Tarski Theorem 2/2

set of pre-fixed points

complete lattice of fixed points

set of post-fixed points

\[ f(x) \sqsubseteq x \]

\[ x = f(x) \]

\[ x \sqsubseteq f(x) \]

\[ \mu f \]

\[ \nu f \]
(Co)Induction Proof Principle

\( f : L \rightarrow L \) monotone

- **induction proof principle:**
  \[
  \frac{f(x) \sqsubseteq x}{\mu f \sqsubseteq x} \quad \mu\text{-rule}
  \]

- **coinduction proof principle:**
  \[
  \frac{x \sqsubseteq f(x)}{x \sqsubseteq \nu f} \quad \nu\text{-rule}
  \]
(Co)Continuous Functions and Kleene Theorem

\( f : L \to L \) is continuous if \( f(\bigcup_{n \geq 0} x_n) = \bigcup_{n \geq 0} f(x_n) \) for any increasing chain \( x_0 \sqsubseteq x_1 \sqsubseteq \cdots \).

\( f : L \to L \) is cocontinuous if \( f(\bigcap_{n \geq 0} x_n) = \bigcap_{n \geq 0} f(x_n) \) for any decreasing chain \( x_0 \sqsupseteq x_1 \sqsupseteq \cdots \).

**Theorem (Kleene)**

If \( f : L \to L \) is continuous then \( \mu f = \bigcup_{n \geq 0} f^n(\bot) \).

If \( f : L \to L \) is cocontinuous then \( \nu f = \bigcap_{n \geq 0} f^n(\top) \).

The theorem supplies a practical way to compute the least/greatest fixed points, or their approximations.
(Co)Continuous Functions and Kleene Theorem

$f : L \rightarrow L$ is continuous if $f(\bigcup_{n\geq 0} x_n) = \bigcup_{n\geq 0} f(x_n)$ for any increasing chain $x_0 \sqsubseteq x_1 \sqsubseteq \cdots$.

$f : L \rightarrow L$ is cocontinuous if $f(\bigcap_{n\geq 0} x_n) = \bigcap_{n\geq 0} f(x_n)$ for any decreasing chain $x_0 \sqsupseteq x_1 \sqsupseteq \cdots$.

**Theorem (Kleene)**

*If $f : L \rightarrow L$ is continuous then $\mu f = \bigcup_{n\geq 0} f^n(\bot)$.  
If $f : L \rightarrow L$ is cocontinuous then $\nu f = \bigcap_{n\geq 0} f^n(\top)$.  

The theorem supplies a practical way to compute the least/greatest fixed points, or their approximations.
Inductive and Coinductive Set Definitions

Context:
- $U$ an universe set
- $(L, ⊑) = (\mathcal{P}(U), ⊆)$
- $\bot = \emptyset$, $\top = U$
- $\bigcup X = \bigcup\{X \mid X \in X\}$,
- $\bigcap X = \bigcap\{X \mid X \in X\}$, where $X \subseteq \mathcal{P}(U)$

Definition

A set $X \subseteq U$ is inductively defined if there is $f$ monotone s.t. $X = \mu f$.
A set $X \subseteq U$ is coinductively defined if there is $f$ monotone s.t. $X = \nu f$. 
Inductive and Coinductive Set Definitions

Context:
- $U$ an universe set
- $(L, \sqsubseteq) = (\mathcal{P}(U), \subseteq)$
- $\bot = \emptyset$, $\top = U$
- $\bigcup \mathcal{X} = \bigcup \langle X \mid X \in \mathcal{X} \rangle$
- $\bigcap \mathcal{X} = \bigcap \langle X \mid X \in \mathcal{X} \rangle$, where $\mathcal{X} \subseteq \mathcal{P}(U)$

Definition

A set $X \subseteq U$ is inductively defined if there is $f$ monotone s.t. $X = \mu f$.

A set $X \subseteq U$ is coinductively defined if there is $f$ monotone s.t. $X = \nu f$. 
Inductive and Coinductive Set Definitions

Context:
- $U$ an universe set
- $(L, \sqsubseteq) = (\mathcal{P}(U), \subseteq)$
- $\bot = \emptyset$, $\top = U$
- $\bigsqcup X = \bigcup \langle X \mid X \in X \rangle$
- $\bigsqcap X = \bigcap \langle X \mid X \in X \rangle$, where $X \subseteq \mathcal{P}(U)$

Definition
A set $X \subseteq U$ is inductively defined if there is $f$ monotone s.t. $X = \mu f$.
A set $X \subseteq U$ is coinductively defined if there is $f$ monotone s.t. $X = \nu f$. 
Inductive and Coinductive Set Definitions

Context:
- $U$ an universe set
- $(L, \sqsubseteq) = (\mathcal{P}(U), \subseteq)$
- $\bot = \emptyset$, $\top = U$
- $\bigsqcup \mathcal{X} = \bigcup \langle X \mid X \in \mathcal{X} \rangle$
- $\bigsqcap \mathcal{X} = \bigcap \langle X \mid X \in \mathcal{X} \rangle$, where $\mathcal{X} \subseteq \mathcal{P}(U)$

Definition

A set $X \subseteq U$ is inductively defined if there is $f$ monotone s.t. $X = \mu f$.

A set $X \subseteq U$ is coinductively defined if there is $f$ monotone s.t. $X = \nu f$. 
Inductive and Coinductive Set Definitions

Context:
- \( U \) an universe set
- \( (L, \sqsubseteq) = (\mathcal{P}(U), \subseteq) \)
- \( \bot = \emptyset \), \( \top = U \)
- \( \bigcup \mathcal{X} = \bigcup\langle X \mid X \in \mathcal{X} \rangle \),
- \( \bigcap \mathcal{X} = \bigcap\langle X \mid X \in \mathcal{X} \rangle \), where \( \mathcal{X} \subseteq \mathcal{P}(U) \)

Definition

A set \( X \subseteq U \) is inductively defined if there is \( f \) monotone s.t. \( X = \mu f \).
A set \( X \subseteq U \) is coinductively defined if there is \( f \) monotone s.t. \( X = \nu f \).
Inductive and Coinductive Set Definitions

Context:
- $U$ an universe set
- $(L, \sqsubseteq) = (\mathcal{P}(U), \subseteq)$
- $\bot = \emptyset$, $\top = U$
- $\bigcup \mathcal{X} = \bigcup \langle X \mid X \in \mathcal{X} \rangle$,
- $\bigcap \mathcal{X} = \bigcap \langle X \mid X \in \mathcal{X} \rangle$, where $\mathcal{X} \subseteq \mathcal{P}(U)$

Definition
A set $X \subseteq U$ is **inductively defined** if there is $f$ monotone s.t. $X = \mu f$.
A set $X \subseteq U$ is **coinductively defined** if there is $f$ monotone s.t. $X = \nu f$. 
Outline

1 Motivation

2 Induction and Coinduction
   - Motivation
   - Theoretical Foundation
   - (Co)Inductive Sets Defined by Ground Inference Systems

3 Reachability Predicates

4 Logical Constrained Rewriting Systems (LCTRSs)

5 A Coinductive Proof System for Reachability Formulas

6 Conclusion
Definition

Let $U$ be a set. A ground (inference) rule on $U$ is a pair $(S, x)$, where $S \subseteq U$, $x \in U$.

S is called the premise of the rule and $x$ the conclusion of the rule.

If $S = \{x_1, x_2, \ldots\}$, then a rule $(S, x)$ is written as

$$
\frac{x_1, x_2, \ldots}{x}
$$

If $S = \emptyset$ then the rule is called axiom.
Functional of a Ground (Inference) System

Definition

A set $\mathcal{R}$ of ground rules yields a function $\hat{\mathcal{R}} : \mathcal{P}(U) \rightarrow \mathcal{P}(U)$ given by

$$\hat{\mathcal{R}}(X) = \{ x \mid (\exists S' \subseteq X)(S', x) \in \mathcal{R} \}.$$ 

Proposition

*If $\mathcal{R}$ is a set of ground rules, then $\hat{\mathcal{R}}$ is monotone.*

It follows that each set of ground rules $\mathcal{R}$ inductively defines a set $\mu \hat{\mathcal{R}}$ and coinductively defines a set $\nu \hat{\mathcal{R}}$. 
Functional of a Ground (Inference) System at Work

\[ U = (\mathbb{Z} \cup \{\langle\rangle, \langle\rangle', \langle\rangle'', \sim\})^\infty \]

\[
[A] \xrightarrow{\langle a \rangle} a \in \mathbb{Z} \\
[B] \xrightarrow{\tau} \langle a, b \rangle \sim \tau a, b \in \mathbb{Z}
\]

\[ R = \{[A], [B]\} \]

\[ X = \begin{cases} 
\langle 2, 5 \rangle \sim \langle 5, 3 \rangle, \\
\sim \langle 7, 2, 6 \rangle, \\
\langle 1 \rangle \sim \langle 1 \rangle \sim \ldots
\end{cases} \]

\[ \hat{R}(X) = \{\langle a \rangle | a \in \mathbb{Z}\} \cup \left\{ \langle a, b \rangle \sim \langle 2, 5 \rangle \sim \langle 5, 3 \rangle, \langle a, b \rangle \vdash \vdash \vdash \langle 7, 2, 6 \rangle, \langle a, b \rangle \sim \langle 1 \rangle \sim \langle 1 \rangle \sim \ldots | a, b \in \mathbb{Z} \right\} \]
Functional of a Ground (Inference) System at Work

\[ U = (\mathbb{Z} \cup \{\langle, \rangle, ",", \sim\})^{\infty} \]

\[ [A] \quad \frac{\langle a \rangle}{a \in \mathbb{Z}} \quad [B] \quad \frac{\tau}{\langle a, b \rangle \sim \tau} \quad a, b \in \mathbb{Z} \]

\[ \mathcal{R} = \{[A], [B]\} \]

\[ X = \left\{ \langle 2, 5 \rangle \sim \langle 5, 3 \rangle, \langle \circ \rangle 7, 2, 6 \langle, \langle 1 \rangle \sim \langle 1 \rangle \sim \ldots \right\} \]

\[ \hat{\mathcal{R}}(X) = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \left\{ \langle a, b \rangle \sim \langle 2, 5 \rangle \sim \langle 5, 3 \rangle, \langle a, b \rangle \sim \langle 1 \rangle \sim \langle 1 \rangle \sim \ldots \mid a, b \in \mathbb{Z} \right\} \]
Functional of a Ground (Inference) System at Work

\[ U = (\mathbb{Z} \cup \{\langle, \rangle, "", "", \sim\})^\infty \]

\[ [A] \xrightarrow{\langle a \rangle} a \in \mathbb{Z} \quad [B] \xrightarrow{\tau} \langle a, b \rangle \sim \tau a, b \in \mathbb{Z} \]

\[ R = \{[A], [B]\} \]

\[ X = \{\langle 2, 5 \rangle \sim \langle 5, 3 \rangle, \sim \langle 7, 2, 6 \rangle \langle, \langle 1 \rangle \sim \langle 1 \rangle \sim \ldots \} \]

\[ \hat{R}(X) = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \{\langle a, b \rangle \sim \langle 2, 5 \rangle \sim \langle 5, 3 \rangle, \langle a, b \rangle \sim \langle 7, 2, 6 \rangle \langle, \langle a, b \rangle \sim \langle 1 \rangle \sim \langle 1 \rangle \sim \ldots \mid a, b \in \mathbb{Z}\} \]
Construction of the Least Fixed Point

\[ U = (\mathbb{Z} \cup \{\langle, \rangle, "", "", \sim\})^\infty \]

\[
\begin{align*}
[A] & \quad \langle a \rangle \quad a \in \mathbb{Z} \\
[B] & \quad \tau \\ & \quad \langle a, b \rangle \sim \tau \quad a, b \in \mathbb{Z}
\end{align*}
\]

\[ \mathcal{R} = \{[A], [B]\} \]

Fortunately \( \hat{\mathcal{R}} \) is continuous

\[
\hat{\mathcal{R}}^0(\emptyset) = \emptyset
\]

\[
\hat{\mathcal{R}}^1(\emptyset) = \{\langle a \rangle | a \in \mathbb{Z}\}
\]

\[
\hat{\mathcal{R}}^2(\emptyset) = \{\langle a \rangle | a \in \mathbb{Z}\} \cup \{\langle b, c \rangle \sim \langle a \rangle | a, b, c \in \mathbb{Z}\} | a, b, c \in \mathbb{Z}\}
\]

\[
\hat{\mathcal{R}}^3(\emptyset) = \{\langle a \rangle | a \in \mathbb{Z}\} \cup \{\langle b, c \rangle \sim \langle a \rangle | a, b, c \in \mathbb{Z}\} \cup \{\langle d, e \rangle \sim \langle b, c \rangle \sim \langle a \rangle | a, b, c, d, e \in \mathbb{Z}\}
\]

...
Construction of the Least Fixed Point

\[ U = (\mathbb{Z} \cup \{ \langle , \rangle , " , " , \sim \})^\infty \]

\[ [A] \xrightarrow{\langle a \rangle} a \in \mathbb{Z} \]

\[ [B] \xrightarrow{\tau} \langle a, b \rangle \sim \tau \ a, b \in \mathbb{Z} \]

\[ \mathcal{R} = \{ [A], [B] \} \]

fortunately \( \hat{\mathcal{R}} \) is continuous

\[ \hat{\mathcal{R}}^0(\emptyset) = \emptyset \]

\[ \hat{\mathcal{R}}^1(\emptyset) = \{ \langle a \rangle | a \in \mathbb{Z} \} \]

\[ \hat{\mathcal{R}}^2(\emptyset) = \{ \langle a \rangle | a \in \mathbb{Z} \} \cup \{ \langle b, c \rangle \sim \langle a \rangle | a, b, c \in \mathbb{Z} \} | a, b, c \in \mathbb{Z} \} \]

\[ \hat{\mathcal{R}}^3(\emptyset) = \{ \langle a \rangle | a \in \mathbb{Z} \} \cup \{ \langle b, c \rangle \sim \langle a \rangle | a, b, c \in \mathbb{Z} \} \cup \{ \langle d, e \rangle \sim \langle b, c \rangle \sim \langle a \rangle | a, b, c, d, e \in \mathbb{Z} \} \]

...
Construction of the Least Fixed Point

\( U = (\mathbb{Z} \cup \{\langle , \rangle, "", "", \sim \})^\infty \)

\[
\begin{align*}
[A] & \quad \langle a \rangle \quad a \in \mathbb{Z} \\
[B] & \quad \frac{\tau}{\langle a, b \rangle \sim \tau} \quad a, b \in \mathbb{Z}
\end{align*}
\]

\( \mathcal{R} = \{[A], [B]\} \)

Fortunately \( \hat{\mathcal{R}} \) is continuous

\( \hat{\mathcal{R}}^0(\emptyset) = \emptyset \)

\( \hat{\mathcal{R}}^1(\emptyset) = \{\langle a \rangle | a \in \mathbb{Z}\} \)

\( \hat{\mathcal{R}}^2(\emptyset) = \{\langle a \rangle | a \in \mathbb{Z}\} \cup \{\langle b, c \rangle \sim \langle a \rangle | a, b, c \in \mathbb{Z}\} | a, b, c \in \mathbb{Z}\} \)

\( \hat{\mathcal{R}}^3(\emptyset) = \{\langle a \rangle | a \in \mathbb{Z}\} \cup \{\langle b, c \rangle \sim \langle a \rangle | a, b, c \in \mathbb{Z}\} \cup \{\langle d, e \rangle \sim \langle b, c \rangle \sim \langle a \rangle | a, b, c, d, e \in \mathbb{Z}\} \)

\[ \ldots \]
Construction of the Least Fixed Point

\[ U = (\mathbb{Z} \cup \{\langle, \rangle, ",", \sim\})^\infty \]

\[
\begin{align*}
[A] & \quad \langle a \rangle \quad a \in \mathbb{Z} \\
[B] & \quad \tau \quad \langle a, b \rangle \sim \tau \quad a, b \in \mathbb{Z}
\end{align*}
\]

\[ \mathcal{R} = \{[A], [B]\} \]

fortunately \( \hat{\mathcal{R}} \) is continuous

\[
\begin{align*}
\hat{\mathcal{R}}^0(\emptyset) &= \emptyset \\
\hat{\mathcal{R}}^1(\emptyset) &= \{\langle a \rangle \mid a \in \mathbb{Z}\} \\
\hat{\mathcal{R}}^2(\emptyset) &= \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \\
& \quad \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\} \cup \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\}
\end{align*}
\]

\[
\begin{align*}
\hat{\mathcal{R}}^3(\emptyset) &= \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \\
& \quad \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\} \cup \{\langle d, e \rangle \sim \langle b, c \rangle \sim \langle a \rangle \mid a, b, c, d, e \in \mathbb{Z}\}
\end{align*}
\]

\[
\ldots
\]
Construction of the Least Fixed Point

\[ U = (\mathbb{Z} \cup \{\langle,\rangle, ,\rangle, \sim\})^\infty \]

\[ \begin{align*}
[A] & \quad a \in \mathbb{Z} \\
\langle a \rangle & \\
[B] & \quad \begin{array}{c}
\tau \\
\langle a, b \rangle \sim \tau
\end{array} \\
a, b \in \mathbb{Z}
\end{align*} \]

\[ \mathcal{R} = \{[A], [B]\} \]

fortunately \( \mathcal{R} \) is continuous

\[ \begin{align*}
\hat{\mathcal{R}}^0(\emptyset) & = \emptyset \\
\hat{\mathcal{R}}^1(\emptyset) & = \{\langle a \rangle \mid a \in \mathbb{Z}\} \\
\hat{\mathcal{R}}^2(\emptyset) & = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \\
& \quad \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\} \mid a, b, c \in \mathbb{Z}\} \\
\hat{\mathcal{R}}^3(\emptyset) & = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \\
& \quad \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\} \cup \\
& \quad \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\} \cup \\
& \quad \{\langle d, e \rangle \sim \langle b, c \rangle \sim \langle a \rangle \mid a, b, c, d, e \in \mathbb{Z}\}
\end{align*} \]
Construction of the Greatest Fixed Point

\[ U = (\mathbb{Z} \cup \{\langle, \rangle, ,, , \sim\})^\infty \]

\[
\begin{align*}
[A] & \quad a \in \mathbb{Z} \\
[B] & \quad \tau \quad \langle a, b \rangle \sim \tau \quad a, b \in \mathbb{Z}
\end{align*}
\]

\[ \mathcal{R} = \{[A], [B]\} \]

fortunately \( \hat{\mathcal{R}} \) is cocontinuous

\[
\hat{\mathcal{R}}^0(U) = U
\]

\[
\begin{align*}
\hat{\mathcal{R}}^1(U) & = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \{\langle a, b \rangle \sim \tau \mid \tau \in U\} \\
\hat{\mathcal{R}}^2(U) & = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \\
& \quad \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\} \cup \\
& \quad \{\langle c, d \rangle \sim \langle a, b \rangle \sim \tau \mid a, b, c, d \in \mathbb{Z}, \tau \in U\}
\end{align*}
\]
Construction of the Greatest Fixed Point

\[ U = (\mathbb{Z} \cup \{\langle , \rangle, ” , ” , \sim \})^\infty \]

\[
\begin{align*}
[A] & \quad \frac{\langle a \rangle}{a \in \mathbb{Z}} \\
[B] & \quad \frac{\tau}{\langle a, b \rangle \sim \tau} a, b \in \mathbb{Z}
\end{align*}
\]

\[ R = \{[A], [B]\} \]

Fortunately \[ \widehat{R} \] is cocontinuous

\[ \widehat{R}^0(U) = U \]

\[ \widehat{R}^1(U) = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \{\langle a, b \rangle \sim \tau \mid \tau \in U\} \]

\[ \widehat{R}^2(U) = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \\
\quad \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\} \cup \\
\quad \{\langle c, d \rangle \sim \langle a, b \rangle \sim \tau \mid a, b, c, d \in \mathbb{Z}, \tau \in U\} \]

\[ \ldots \]
Construction of the Greatest Fixed Point

\[ U = (\mathbb{Z} \cup \{ \langle \cdot, \cdot \rangle, "", "", \leadsto \})^\infty \]

\[ [A] \quad \frac{}{\langle a \rangle} \quad a \in \mathbb{Z} \quad [B] \quad \frac{\tau}{\langle a, b \rangle \leadsto \tau} \quad a, b \in \mathbb{Z} \]

\[ \mathcal{R} = \{ [A], [B] \} \]

Fortunately \( \mathcal{R} \) is cocontinuous

\[ \hat{\mathcal{R}}^0(U) = U \]

\[ \hat{\mathcal{R}}^1(U) = \{ \langle a \rangle \mid a \in \mathbb{Z} \} \cup \{ \langle a, b \rangle \leadsto \tau \mid \tau \in U \} \]

\[ \hat{\mathcal{R}}^2(U) = \{ \langle a \rangle \mid a \in \mathbb{Z} \} \cup \{ \langle b, c \rangle \leadsto \langle a \rangle \mid a, b, c \in \mathbb{Z} \} \cup \{ \langle c, d \rangle \leadsto \langle a, b \rangle \leadsto \tau \mid a, b, c, d \in \mathbb{Z}, \tau \in U \} \]

\[ \ldots \]
Construction of the Greatest Fixed Point

\[ U = (\mathbb{Z} \cup \{\langle , \rangle, , , \rangle \sim \})^\infty \]

\[
\begin{align*}
[A] & \quad a \in \mathbb{Z} & [B] & \quad \tau \quad \langle a, b \rangle \sim \tau \quad a, b \in \mathbb{Z}
\end{align*}
\]

\[ R = \{[A], [B]\} \]

Fortunately \( \hat{R} \) is cocontinuous

\[ \hat{R}^0(U) = U \]

\[ \hat{R}^1(U) = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \{\langle a, b \rangle \sim \tau \mid \tau \in U\} \]

\[ \hat{R}^2(U) = \{\langle a \rangle \mid a \in \mathbb{Z}\} \cup \{\langle b, c \rangle \sim \langle a \rangle \mid a, b, c \in \mathbb{Z}\} \cup \{\langle c, d \rangle \sim \langle a, b \rangle \sim \tau \mid a, b, c, d \in \mathbb{Z}, \tau \in U\} \]

\[ \ldots \]
Induction Principle on Rules

if \( \hat{R}(X) \subseteq X \) then \( \mu(\hat{R}) \subseteq X \)

that means that

for a given \( X \),
if for all rules \( (S, x) \in R \), if \( S \subseteq X \), then also \( x \in X \)
then \( \mu(\hat{R}) \subseteq X \)
Coinduction Principles on Rules

if $X \subseteq \hat{R}(X)$ then $X \subseteq \nu(\hat{R})$

that means that

for a given $X$,
if for all $x \in X$ there is a rule $(S, x) \in R$ with $S \subseteq X$
then $X \subseteq \nu(\hat{R})$
Example

\[
\langle a, b \rangle \rightarrow \langle a - b, b \rangle \text{ if } a \geq b \\
\langle a, b \rangle \rightarrow \langle a, b - a \rangle \text{ if } b \geq a \\
\langle a, b \rangle \rightarrow \langle a + b \rangle \text{ if } a = 0 \lor b = 0
\]

\[U = (\mathbb{Z} \cup \{\langle, \rangle, ,, , \sim\})^\infty\]

\[
\begin{align*}
[A] & \quad a \in \mathbb{Z} \\
[B] & \quad \langle a, b \rangle \sim^\tau \exists \text{ transition from } \langle a, b \rangle \text{ to } hd(\tau)
\end{align*}
\]

The set of finite executions: \[C^+ = \mu [A, B]\]

The set of infinite and finite executions: \[C^\infty = \nu [A, B]\]

The set of infinite executions: \[C^\omega = \nu [B]\]
Proof Trees

finite proof tree

infinite proof tree
A tree is well-founded if the relation on the nodes, which contains a pair of nodes \((n, p)\) if \(p\) is the parent of \(n\), is well-founded.

Remark

Finite proof trees are well-founded.

Proposition

Let \(\mathcal{R}\) be a set of a set of ground rules over \(U\) such that \(\mathcal{R}\) is cocontinuous. Then \(x \in \mu \hat{\mathcal{R}}\) iff there is a well-founded proof tree of \(x\) under \(\mathcal{R}\). Then \(x \in \nu \hat{\mathcal{R}}\) iff there is a proof tree of \(x\) under \(\mathcal{R}\).
Plan

1. Motivation

2. Induction and Coinduction
   - Motivation
   - Theoretical Foundation
   - (Co)Inductive Sets Defined by Ground Inference Systems

3. Reachability Predicates

4. Logical Constrained Rewriting Systems (LCTRSs)

5. A Coinductive Proof System for Reachability Formulas

6. Conclusion
Overview

Semantics First! (J. Goguen)

- transition systems
- state predicate
- derivative (semantically)
- reachability predicate as pairs of state predicates
Transition Systems

\((M, \sim), \text{ with } \sim \subseteq M \times M\)

execution paths:

\[
\begin{align*}
\gamma \in M, \gamma \text{ irreducible} & \quad \frac{\tau}{\gamma_0 \circ \tau} \quad \gamma_0 \sim \text{hd}(\tau)
\end{align*}
\]
State Predicates and their Derivatives

- State predicate: $P \subseteq M$
- Derivative of a state predicate $P$:
  \[
  \partial(P) = \{ \gamma' \mid \gamma \sim \gamma' \text{ for some } \gamma \in P \} 
  \]
Reachability Predicates

- A **reachability predicate** is a pair of state predicates: $P \Rightarrow Q$
- Models: transition systems
- Satisfiability: $(M, \sim) \models \forall P \Rightarrow Q$ iff any execution path $\tau$ starting from $P$ ($hd(\tau) \in P$) is infinite or eventually reaches $Q$
Satisfiability Coinductively

\[(M, \leadsto) \models \forall P \Rightarrow Q\]

iff

\[P \Rightarrow Q \in \nu \DVP\]

where \(\DVP\) consists of the following rules:

\[
\begin{align*}
\text{[Subs]} & \quad \frac{}{P \Rightarrow Q \quad P \subseteq Q} \\
\text{[Step]} & \quad \frac{\partial(P \setminus Q) \Rightarrow Q}{P \Rightarrow Q \quad P \setminus Q \text{ runnable}}
\end{align*}
\]

\(P\) is runnable if \(P \neq \emptyset\) and for all \(\gamma \in P\) there is \(\gamma' \in M\) s.t. \(\gamma \leadsto \gamma'\).
Plan

1. Motivation

2. Induction and Coinduction
   - Motivation
   - Theoretical Foundation
   - (Co)Inductive Sets Defined by Ground Inference Systems

3. Reachability Predicates

4. Logical Constrained Rewriting Systems (LCTRSs)

5. A Coinductive Proof System for Reachability Formulas

6. Conclusion
Overview

(based on IJCAR 2018 paper)

- builtin models (local computations, constraints over builtin model solvable by SMT)
  - state predicates as constrained terms = \langle \text{term} \mid \text{constraint-formulas} \rangle
  - reachability predicates as reachability formulas = pair of constrained terms
  - transition systems represented as set of rewriting rules logically constrained (LCTRSs)
Overview

(based on IJCAR 2018 paper)

- builtin models (local computations, constraints over builtin model solvable by SMT)
- state predicates as constrained terms = \langle \text{term} \mid \text{constraint-formulas} \rangle
- reachability predicates as reachability formulas = pair of constrained terms
- transition systems represented as set of rewriting rules logically constrained (LCTRSs)
Overview

(based on IJCAR 2018 paper)

- builtin models (local computations, constraints over builtin model solvable by SMT)
- state predicates as constrained terms = \langle term | constraint-formulas \rangle
- reachability predicates as reachability formulas = pair of constrained terms
- transition systems represented as set of rewriting rules logically constrained (LCTRSs)
Overview

(based on IJCAR 2018 paper)

- builtin models (local computations, constraints over builtin model solvable by SMT)
- state predicates as constrained terms = \langle \text{term} | \text{constraint-formulas} \rangle
- reachability predicates as reachability formulas = pair of constrained terms
- transition systems represented as set of rewriting rules logically constrained (LCTRSs)
Signatures Modulo a Builtin Model

- a builtin model $M^b$ for a many-sorted builtin signature $\Sigma^b = (S^b, F^b)$.
  
  Example: $S^b = \{\text{Int, Bool}\}$, $F^b = \{+, \times, \wedge, \vee, \ldots\}$.

- an order-sorted signature $(S, \leq, \Sigma)$ including $\Sigma^b$

- $M^b$ is freely extended to a $\Sigma$-model $M^\Sigma$ s. t.:
  
  - each $f \in \Sigma \setminus \Sigma^b$ is interpreted as a term constructor
  
  - the $\Sigma^b$-terms reduced to their values in $M^b$
Signatures Modulo a Builtin Model

- a built-in model $M^b$ for a many-sorted built-in signature $\Sigma^b = (S^b, F^b)$. Example: $S^b = \{\text{Int}, \text{Bool}\}$, $F^b = \{+, \times, \&, \lor, \ldots\}$.
- an order-sorted signature $(S, \leq, \Sigma)$ including $\Sigma^b$

$M^b$ is freely extended to a $\Sigma$-model $M^\Sigma$ s.t.:
- each $f \in \Sigma \setminus \Sigma^b$ is interpreted as a term constructor
- the $\Sigma^b$-terms reduced to their values in $M^b$
Signatures Modulo a Built-in Model

- a built-in model $M^b$ for a many-sorted built-in signature $\Sigma^b = (S^b, F^b)$. Example: $S^b = \{\text{Int}, \text{Bool}\}$, $F^b = \{+, \times, \land, \lor, \ldots\}$.
- an order-sorted signature $(S, \leq, \Sigma)$ including $\Sigma^b$
- $M^b$ is freely extended to a $\Sigma$-model $M^\Sigma$ s. t.:
  - each $f \in \Sigma \setminus \Sigma^b$ is interpreted as a term constructor
  - the $\Sigma^b$-terms reduced to their values in $M^b$
Constrained Terms

- **constrained terms** \( \langle t \mid \phi \rangle \)
  - \( t \) is a \( \Sigma \)-term
  - \( \phi \) is a first-order (with equality) formula

**Example:**
\[
\langle \text{init}(n) \mid \exists u.1 < u < n \land n \text{ mod } u = 0 \rangle
\]
\( \text{init} \in \Sigma_{\text{Int},\text{Cfg}}, \) \( n \) and \( u \) variables of sort \( \text{Int} \)

- semantically, a constrained term defines a state predicate
\[
[\langle t \mid \phi \rangle] \overset{\Delta}{=} \{ \alpha(t) \mid \alpha : X \rightarrow M^\Sigma \text{ s.t. } M^\Sigma, \alpha \models \phi \}.
\]
Constrained Terms

- **constrained terms** \( \langle t \mid \phi \rangle \)
  - \( t \) is a \( \Sigma \)-term
  - \( \phi \) is a first-order (with equality) formula

**Example:**
\[
\langle \text{init}(n) \mid \exists u.1 < u < n \land n \text{ mod } u = 0 \rangle
\]

\( \text{init} \in \Sigma_{\text{Int}, \text{Cfg}} \), \( n \) and \( u \) variables of sort \( \text{Int} \)

- semantically, a constrained term defines a state predicate
\[
\llbracket \langle t \mid \phi \rangle \rrbracket \triangleq \{ \alpha(t) \mid \alpha : X \rightarrow M^\Sigma \text{ s.t. } M^\Sigma, \alpha \models \phi \}.
\]
Constrained Terms

- **Constrained terms** $\langle t \mid \phi \rangle$
  - $t$ is a $\Sigma$-term
  - $\phi$ is a first-order (with equality) formula

- **Example:**
  $\langle \text{init}(n) \mid \exists u.1 < u < n \land n \mod u = 0 \rangle$

  $\text{init} \in \Sigma_{\text{Int},\text{Cfg}}$, $n$ and $u$ variables of sort $\text{Int}$

- Semantically, a constrained term defines a state predicate
  $\llbracket \langle t \mid \phi \rangle \rrbracket \triangleq \{ \alpha(t) \mid \alpha : X \to M^\Sigma \text{ s.t. } M^\Sigma, \alpha \models \phi \}$. 
Specification of Transition Systems as LCTRS

- LCTRS = Logical Constrained Term Rewriting System
- example:

  $\text{init}(n) \rightarrow \text{loop}(n, 2)$ if $\top$, 
  $\text{loop}(i \times k, i) \rightarrow \text{comp}$ if $k > 1$, 
  $\text{loop}(n, i) \rightarrow \text{loop}(n, i + 1)$ if $\neg(\exists k. k > 1 \land n = i \times k)$.

- the general form of a rule: $l \rightarrow r$ if $\phi$
- we may think that $\langle l \mid \phi \rangle$ and $\langle r \mid \phi \rangle$ are two constrained terms, but this is not entirely true (see the next slide)
- a LCTRS $R$ defines a transition relation $\sim R$ between the corresponding instances of $l$ and $r$ satisfying $\phi$
- example: $\text{loop}(5, 2) \sim R \text{loop}(5, 3)$, $\text{loop}(8, 2) \sim R \text{comp}$
Specification of Transition Systems as LCTRS

- **LCTRS** = Logical Constrained Term Rewriting System
- **example:**

  \[
  \text{init}(n) \rightarrow \text{loop}(n, 2) \text{ if } \top,
  \]
  \[
  \text{loop}(i \times k, i) \rightarrow \text{comp} \text{ if } k > 1,
  \]
  \[
  \text{loop}(n, i) \rightarrow \text{loop}(n, i + 1) \text{ if } \neg(\exists k. k > 1 \land n = i \times k).
  \]

- the general form of a rule: \( l \rightarrow r \text{ if } \phi \)
- we may think that \( \langle l | \phi \rangle \) and \( \langle r | \phi \rangle \) are two constrained terms, but this is not entirely true (see the next slide)
- a LCTRS \( \mathcal{R} \) defines a transition relation \( \sim_{\mathcal{R}} \) between the corresponding instances of \( l \) and \( r \) satisfying \( \phi \)
- **example:** \( \text{loop}(5, 2) \sim_{\mathcal{R}} \text{loop}(5, 3), \text{loop}(8, 2) \sim_{\mathcal{R}} \text{comp} \)
Specification of Transition Systems as LCTRS

- LCTRS = Logical Constrained Term Rewriting System
- example:

\[
\begin{align*}
\text{init}(n) & \rightarrow \text{loop}(n, 2) \text{ if } \top, \\
\text{loop}(i \times k, i) & \rightarrow \text{comp} \text{ if } k > 1, \\
\text{loop}(n, i) & \rightarrow \text{loop}(n, i + 1) \text{ if } \neg(\exists k. k > 1 \land n = i \times k).
\end{align*}
\]

- the general form of a rule: \( l \rightarrow r \text{ if } \phi \)
- we may think that \( \langle l \mid \phi \rangle \) and \( \langle r \mid \phi \rangle \) are two constrained terms, but this is not entirely true (see the next slide)
- a LCTRS \( \mathcal{R} \) defines a transition relation \( \sim_{\mathcal{R}} \) between the corresponding instances of \( l \) and \( r \) satisfying \( \phi \)
  example: \( \text{loop}(5, 2) \sim_{\mathcal{R}} \text{loop}(5, 3), \text{loop}(8, 2) \sim_{\mathcal{R}} \text{comp} \)
Reachability Properties of LCTRSs

- a reachability formula is a pair of constrained terms \( \varphi \Rightarrow \varphi' \)
- semantics:

\[ R \models^\forall \varphi \Rightarrow \varphi' \]

iff

\[ (M^\Sigma, \sim_R) \models^\forall \llbracket \sigma(\varphi) \rrbracket \Rightarrow \llbracket \sigma(\varphi') \rrbracket \]

for each \( \sigma : \text{var}(\varphi) \cap \text{var}(\varphi') \rightarrow M^\Sigma \)

- the shared variable must have the same values
Derivatives of Constrained Terms

- the derivatives of state predicates are extended to constrained terms
- the set of derivatives of a constrained term $\varphi \triangleq \langle t | \phi \rangle$ w.r.t. a rule $l \rightarrow r$ if $\phi_{lr}$ is

$$
\Delta_{l,r,\phi_{lr}}(\varphi) \triangleq \{ \langle c[r] | \phi' \rangle | \phi' \triangleq \phi \land t = c[l] \land \phi_{lr},
\quad c[\cdot] \text{ an appropriate context}
\quad \phi' \text{ is satisfiable} \}.
$$

- example:

$$
\Delta_{R}(\langle \text{init}(n) | \exists u.1 < u < n \land n \mod u = 0 \rangle) = \{ \langle \text{loop}(n, 2) | \exists u.1 < u < n \land n \mod u = 0 \rangle \}.
$$
Symbolic Derivatives and the Concrete Ones Agree

Theorem

Let $\varphi \triangleq \langle t \mid \phi \rangle$ be a constrained term, $\mathcal{R}$ a constrained rule system, and $(M^\Sigma, \sim_{\mathcal{R}})$ the transition system defined by $\mathcal{R}$. Then $[[\Delta_{\mathcal{R}}(\varphi)]] = \partial([[\varphi]]).$
Plan

1. Motivation

2. Induction and Coinduction
   - Motivation
   - Theoretical Foundation
   - (Co)Inductive Sets Defined by Ground Inference Systems

3. Reachability Predicates

4. Logical Constrained Rewriting Systems (LCTRSs)

5. A Coinductive Proof System for Reachability Formulas

6. Conclusion
Overview

- lift up the inference system to reachability formulas
- add a circular inference rule to infinite proof trees into finite ones
Overview

- lift up the inference system to reachability formulas
- add a circular inference rule to infinite proof trees into finite ones
Proof System \((\text{DSTEP}(\mathcal{R}))\)

it corresponds to the semantic coinductive definition

\[
\begin{align*}
\text{[axiom]} & \quad \langle t_l \mid \bot \rangle \Rightarrow \langle t_r \mid \phi_r \rangle \\
\text{[subs]} & \quad \langle t_l \mid \phi_l \land \neg(\exists \tilde{x}. t_l = t_r \land \phi_r) \rangle \Rightarrow \langle t_r \mid \phi_r \rangle \\
\text{[der}\forall] & \quad \langle t^j \mid \phi^j \rangle \Rightarrow \langle t_r \mid \phi_r \rangle, j \in \{1, \ldots, n\}
\end{align*}
\]

\(\text{cond}_s, \text{cond}_d\)

where \(\Delta_{\mathcal{R}}(\langle t_l \mid \phi_l \rangle) = \{\langle t^1 \mid \phi^1 \rangle, \ldots, \langle t^n \mid \phi^n \rangle\}\)

The conditions ensure the sound application of the rules.
Example

The infinite proof tree for the reachability formula $\langle \text{init}(n) \mid \psi \rangle \Rightarrow \varphi_r$, where $\psi \triangleq \exists u. 1 < u < n \land n \ mod \ u = 0$ and $\varphi_r \triangleq \langle \text{comp} \mid \top \rangle$:

\[
\begin{align*}
\langle \text{comp} \mid \bot \rangle & \Rightarrow \varphi_r \quad \text{[axiom]} \\
\langle \text{comp} \mid \psi \land \phi_a \rangle & \Rightarrow \varphi_r \quad \text{[subs]} \\
\langle \text{comp} \mid \psi \land \phi_2 \land \phi_b \rangle & \Rightarrow \varphi_r \quad \text{[subs]} \\
\langle \text{loop}(n, 3) \mid \psi \land \phi_2 \rangle & \Rightarrow \varphi_r \\
\langle \text{loop}(n, 2) \mid \psi \rangle & \Rightarrow \varphi_r \\
\langle \text{init}(n) \mid \psi \rangle & \Rightarrow \varphi_r
\end{align*}
\]

\[
\begin{align*}
\langle \text{comp} \mid \bot \rangle & \Rightarrow \varphi_r \quad \text{[axiom]} \\
\langle \text{comp} \mid \psi \land \phi_2 \land \phi_b \rangle & \Rightarrow \varphi_r \quad \text{[subs]} \\
\langle \text{loop}(n, 3) \mid \psi \land \phi_2 \rangle & \Rightarrow \varphi_r \\
\langle \text{loop}(n, 2) \mid \psi \rangle & \Rightarrow \varphi_r \\
\langle \text{init}(n) \mid \psi \rangle & \Rightarrow \varphi_r \quad \text{[der $\forall$]}
\end{align*}
\]

Infinite proof trees cannot be handled in practice, therefore we look for finite representations of them.
Let $G$ be a finite set reachability formulas (goals that we intend to prove). Then the set of rules $\text{DCC}(\mathcal{R}, G)$ consists of $\text{DSTEP}(\mathcal{R})$, together with

\[
\begin{align*}
\langle t_r^c | \phi_l \land \phi \land \phi_r^c \rangle & \Rightarrow \varphi_r, \\
\langle t_l | \phi_l \land \neg \phi \rangle & \Rightarrow \varphi_r \\
\langle t_l | \phi_l \rangle & \Rightarrow \varphi_r
\end{align*}
\]

[circ] \quad \phi \text{ is } \exists \text{var}(t_i^c, \phi_i^c). t_l = t_i^c \land \phi_i^c, \\
\langle t_i^c | \phi_i^c \rangle & \Rightarrow \langle t_r^c | \phi_r^c \rangle \in G.

**Theorem (Circularity Principle)**

Let $\mathcal{R}$ be a constrained rule system and $G$ a set of goals. If $(\mathcal{R}, G) \vdash \forall G$ then $\mathcal{R} \models \forall G$. 

Lucanu (UAIC) 
Proving Reachability by Coinduction 
SYNASC, September 21, 2018 
51 / 55
Example

In order to prove $\langle \text{init}(n) \mid \psi \rangle \Rightarrow \langle \text{comp} \mid \top \rangle$, we choose the following set of circularities

$$G = \left\{ \begin{array}{l}
\langle \text{init}(n) \mid \psi \rangle \Rightarrow \langle \text{comp} \mid \top \rangle, \\
\langle \text{loop}(n, i) \mid 2 \leq i \land \exists u. i \leq u < n \land n \mod u = 0 \rangle \Rightarrow \langle \text{comp} \mid \top \rangle
\end{array} \right\}.$$

The finite proof tree for the first circularity:

$$
\frac{
\frac{\langle \text{comp} \mid \bot \rangle \Rightarrow \varphi_r}{[\text{axiom}]} \\
\langle \text{comp} \mid \psi \land \phi \land \top \rangle \Rightarrow \varphi_r}{[\text{subs}]} \\
\langle \text{loop}(n, 2) \mid \psi \land \neg \phi \rangle \Rightarrow \varphi_r}{[\text{axiom}]} \\
\langle \text{loop}(n, 2) \mid \psi \rangle \Rightarrow \varphi_r}{[\text{circ}]} \\
\langle \text{init}(n) \mid \psi \rangle \Rightarrow \varphi_r}{[\text{der} \forall]} \\
\frac{
\langle \text{init}(n) \mid \psi \rangle \Rightarrow \varphi_r}{[\text{der} \forall]}$
$$
Example

In order to prove \( \langle \text{init}(n) \mid \psi \rangle \Rightarrow \langle \text{comp} \mid \top \rangle \), we choose the following set of circularities
\[
G = \left\{ \langle \text{init}(n) \mid \psi \rangle \Rightarrow \langle \text{comp} \mid \top \rangle , \\
\langle \text{loop}(n, i) \mid 2 \leq i \land \exists u. i \leq u < n \land n \mod u = 0 \rangle \Rightarrow \langle \text{comp} \mid \top \rangle \right\}.
\]

The finite proof tree for the first circularity:
\[
\frac{\langle \text{comp} \mid \bot \rangle \Rightarrow \varphi_r \quad \text{[axiom]}}{\langle \text{comp} \mid \psi \land \phi \land \top \rangle \Rightarrow \varphi_r \quad \text{[subs]}} \quad \frac{\langle \text{loop}(n, 2) \mid \psi \land \neg \varphi \rangle \Rightarrow \varphi_r \quad \text{[axiom]}}{\langle \text{loop}(n, 2) \mid \psi \rangle \Rightarrow \varphi_r \quad \text{[circ]}} \quad \frac{\langle \text{init}(n) \mid \psi \rangle \Rightarrow \varphi_r \quad \text{[der} \forall \text{]} }{\langle \text{init}(n) \mid \psi \rangle \Rightarrow \varphi_r \quad \text{[der} \forall \text{]}}
\]
Example Proof Tree for the Second Circularity

\[
\langle c \mid \perp \rangle \Rightarrow \varphi_r \\
\langle c \mid \psi_i \land \psi_b \land \psi_c \rangle \Rightarrow \varphi_r \\
\langle l(n, i + 1) \mid \psi_i \land \psi_b \land \neg \psi_c \rangle \Rightarrow \varphi_r \\
\langle l(n, i) \mid \psi_i \rangle \Rightarrow \langle c \mid T \rangle \quad [\text{der} \forall]
\]
Plan

1. Motivation
2. Induction and Coinduction
   - Motivation
   - Theoretical Foundation
   - (Co)Inductive Sets Defined by Ground Inference Systems
3. Reachability Predicates
4. Logical Constrained Rewriting Systems (LCTRSs)
5. A Coinductive Proof System for Reachability Formulas
6. Conclusion
Conclusion

- we defined a framework for proving reachability properties consisting of
  - logical constrained rewriting systems, for specifying transition systems (LCTRS)
  - a coinductive proof system consisting of four rules
- LCTRSs are expressive enough to describe semantics of programming languages (K Framework, G. Roșu)
- there is a prototype that implements the proof system (Ciobâcă, Buruiană)
- further work includes
  - unification modulo builtins (a first step in WOLLIC 2018)
  - extension of the proof systems to program equivalence (a first step in FAoC 2015)