Proving Reachability Modulo Theories

Ștefan Ciobâcă

Faculty of Computer Science
Alexandru Ioan Cuza University, Iași, Romania
stefan.ciobaca@info.uaic.ro

July 8th, 2017

(joint work with Dorel Lucanu)
Summary

Given

\[ S = \{ \begin{array}{c} l_1 \rightarrow r_1 \text{ if } \phi_1, \\ \vdots, \\ l_n \rightarrow r_n \text{ if } \phi_n \end{array} \}, \]

how to prove

\[ S \vdash u \rightarrow^* v \text{ if } \phi? \]
Summary

Given

\[ S = \{ l_1 \rightarrow r_1 \text{ if } \phi_1, \]
\[ \ldots, \]
\[ l_n \rightarrow r_n \text{ if } \phi_n \}, \]

how to prove

\[ S \vdash u \rightarrow^* v \text{ if } \phi? \]

Example. Given:

\[ S = \{ \text{init}(n) \rightarrow \text{loop}(0, n) \text{ if } \text{true}, \]
\[ \text{loop}(s, i) \rightarrow \text{loop}(s + i, i - 1) \text{ if } i > 0 \]
\[ \text{loop}(s, 0) \rightarrow \text{done}(s) \text{ if } \text{true} \]

prove

\[ S \vdash \text{init}(n) \rightarrow^* \text{done}(n(n + 1)/2) \text{ if } n \geq 0. \]
Plan

Summary

A Coinductive Definition of Reachability Properties

Constrained Terms

Constrained Rule Systems

Proving Reachability Properties

Tool Demo

Conclusions
Transition Systems

Let \((M, \leadsto)\) be a transition system, where \(\leadsto \subseteq M \times M\). We write \(\gamma \leadsto \gamma'\) for \((\gamma, \gamma') \in \leadsto\). An element \(\gamma \in M\) is \textit{irreducible} if there is no \(\gamma' \in M\) such that \(\gamma \leadsto \gamma'\).
Transition Systems

Let $(M, \leadsto)$ be a transition system, where $\leadsto \subseteq M \times M$. We write $\gamma \leadsto \gamma'$ for $(\gamma, \gamma') \in \leadsto$. An element $\gamma \in M$ is irreducible if there is no $\gamma' \in M$ such that $\gamma \leadsto \gamma'$.

**Definition (Execution Path)**

The set of (complete) execution paths is coinductively defined by the following rules:

\[
\begin{align*}
\Gamma & \quad \gamma \in M, \gamma \text{ irreducible} \\
\Gamma & \quad \gamma_0 \circ \tau \leadsto hd(\tau)
\end{align*}
\]

where the function $hd$ is itself defined coinductively by $hd(\gamma) = \gamma$ and $hd(\gamma_0 \circ \tau) = \gamma_0$. 
Definition (State and Reachability Predicates)

A state predicate is a subset $P \subseteq M$. A reachability predicate is a pair $P \Rightarrow Q$, where $P$ and $Q$ are state predicates. A state predicate $P$ is runnable if for all $\gamma \in P$ there exists $\gamma' \in M$ such that $\gamma \leadsto \gamma'$. 
Definition (Derivative of a State Predicate)

Given a state predicate $P$, the derivative of $P$ is the state predicate $\partial(P)$, defined by $\partial(P) = \{ \gamma' \mid \gamma \sim \gamma' \text{ for some } \gamma \in P \}$. 
Satisfaction of Reachability Predicates

Definition (Satisfaction of a Reachability Predicate)

An execution path \( \tau \) satisfies a reachability predicate \( P \Rightarrow Q \), written \( \tau \models \forall \ P \Rightarrow Q \), iff \( \langle \tau, P \Rightarrow Q \rangle \in \nu \text{EPSRP} \), where EPSRP consists of the following rules:

\[
\begin{align*}
\langle \tau, P \Rightarrow Q \rangle & \quad \text{hd}(\tau) \in P \cap Q \\
\langle \tau, P \Rightarrow Q \rangle & \quad \langle \gamma_0 \circ \tau, P \Rightarrow Q \rangle \quad \gamma_0 \in P, \gamma_0 \rightsquigarrow \text{hd}(\tau).
\end{align*}
\]

Definition (Demonically Valid Predicates, Coinductively)

We say that \( P \Rightarrow Q \) is demonically valid, and we write \( (M, \rightsquigarrow) \models \forall \ P \Rightarrow Q \), iff \( P \Rightarrow Q \in \nu \text{DVP} \), where DVP consists of the following rules:

\[ J \quad \text{Subsumption} \]
\[ K \quad \text{Step} \]

\[ P \Rightarrow Q \subseteq Q \]
\[ P \Rightarrow Q \quad \text{P \Rightarrow Q \ runnable} \]
Satisfaction of Reachability Predicates

Definition (Satisfaction of a Reachability Predicate)

An execution path $\tau$ satisfies a reachability predicate $P \Rightarrow Q$, written $\tau \models \forall P \Rightarrow Q$, iff $\langle \tau, P \Rightarrow Q \rangle \in \nu \text{EPSRP}$, where EPSRP consists of the following rules:

$$\frac{hd(\tau) \in P \cap Q}{\langle \tau, P \Rightarrow Q \rangle} \quad \frac{\langle \tau, \partial(P) \Rightarrow Q \rangle}{\langle \gamma_0 \circ \tau, P \Rightarrow Q \rangle} \quad \gamma_0 \in P, \gamma_0 \rightsquigarrow hd(\tau).$$

Definition (Demonically Valid Predicates, Coinductively)

We say that $P \Rightarrow Q$ is demonically valid, and we write

$$(M, \rightsquigarrow) \models \forall P \Rightarrow Q,$$

iff $P \Rightarrow Q \in \nu \text{DVP}$, where DVP consists of the following rules:

$$\text{[Subsumption]} \quad \frac{P \Rightarrow Q \quad P \subseteq Q}{P \Rightarrow Q} \quad \text{[Step]} \quad \frac{\partial(P \setminus Q) \Rightarrow Q}{P \Rightarrow Q} \quad P \setminus Q \text{ runnable.}$$
Plan

Summary

A Coinductive Definition of Reachability Properties

Constrained Terms

Constrained Rule Systems

Proving Reachability Properties

Tool Demo

Conclusions
**Builtins**

**Definition (Builtin Signature)**

A *builtin signature* $\Sigma^b \triangleq (S^b, F^b)$ is any many-sorted signature that includes the following distinguished objects:

- a sort $\text{Bool}$ together with two constants $\top$ and $\bot$ of sort $\text{Bool}$,
- the propositional operations $\neg : \text{Bool} \to \text{Bool}$, $\wedge, \vee, \rightarrow : \text{Bool} \times \text{Bool} \to \text{Bool}$, and
- an equality predicate $= : s \times s \to \text{Bool}$ for each sort $s \in S^b$. 
Builtins

Definition (Builtin Signature)
A *builtin signature* $\Sigma^b \triangleq (S^b, F^b)$ is any many-sorted signature that includes the following distinguished objects:
- a sort $\text{Bool}$ together with two constants $\top$ and $\bot$ of sort $\text{Bool}$,
- the propositional operations $\neg : \text{Bool} \to \text{Bool}$,
  $\land, \lor, \to : \text{Bool} \times \text{Bool} \to \text{Bool}$, and
- an equality predicate $\equiv : s \times s \to \text{Bool}$ for each sort $s \in S^b$.

Definition (Builtin Model)
A *builtin model* $M^b$ is a model of a builtin signature $\Sigma^b$, where the interpretation of the distinguished objects of the builtin signature is fixed as follows: $M^b_{\text{Bool}} = \{\top, \bot\}$, $M^b_\top = \top$, $M^b_\bot = \bot$, $M^b_{\equiv}(a, b) = \top$ iff $a = b$, $M^b_\land(\top) = \bot$, $M^b_\land(\bot) = \top$, $M^b_\land(\top, b) = M^b_\land(b, \top) = b$, $M^b_\land(\bot, b) = M^b_\land(b, \bot) = \bot$, and so on.
Definition (Signature Modulo a Builtin Model)

A signature modulo a builtin model is a tuple $\Sigma \triangleq (S, \leq, F, M^b)$ consisting of

- an order-sorted signature $(S, \leq, F)$, and
- a builtin $\Sigma^b$-model $M^b$, where $\Sigma^b \triangleq (S^b, F^b)$ is a builtin subsignature of $(S, \leq, F)$.

We further assume that the only builtin constants in $\Sigma$ are the elements of the builtin model, i.e., $F^b_{\varepsilon, s} = M^b_s$. $\Sigma^b$ is called the builtin subsignature of $\Sigma$ and $\Sigma^c = (S, \leq, (F \setminus F^b) \cup \bigcup_{s \in S^b} F^b_{\varepsilon, s})$ the constructor subsignature of $\Sigma$. 
**Model Modulo Builtins**

**Definition (Model $M^\Sigma$ Generated by a Signature Modulo a Builtin Model)**

Let $\Sigma \triangleq (S, \leq, F, M^b)$ be a signature modulo a builtin model. The model $M^b$ is extended in a protected way to a $(S, \leq, F)$-model $M^\Sigma$ defined as follows:

- $M^\Sigma_s = T^\Sigma_{c,s}(M^b)$ for each $s \in S \setminus S^b$, i.e. $M^\Sigma_s$ includes the constructor terms;
- $M^\Sigma_f = M^b_f$ for each $f \in F^b$;
- $M^\Sigma_f$ is the term constructor $M^\Sigma_f(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$ for each $f \in F \setminus F^b$. 
Constraint Formulae

Definition (Constraint Formulae)

The set $\text{CF}(\Sigma, X)$ of constraint formulae over variables $X$ is inductively defined as follows:

$$\phi ::= b \mid t_1 = t_2 \mid (\exists x)\phi' \mid \neg\phi' \mid \phi_1 \land \phi_2$$

where $b$ ranges over $T_{\Sigma,\text{Bool}}(X)$, $t_i$ over $T_{\Sigma,s_i}(X)$ such that $s_1$ and $s_2$ are in the same connected component, and $x$ ranges over all variables.
Constraint Formulae

Definition (Constraint Formulae)
The set \( \text{CF}(\Sigma, X) \) of constraint formulae over variables \( X \) is inductively defined as follows:

\[
\phi ::= b \mid t_1 = ? t_2 \mid (\exists x)\phi' \mid \neg \phi' \mid \phi_1 \land \phi_2
\]

where \( b \) ranges over \( T_{\Sigma, \text{Bool}}(X) \), \( t_i \) over \( T_{\Sigma, s_i}(X) \) such that \( s_1 \) and \( s_2 \) are in the same connected component, and \( x \) ranges over all variables.

Definition (Semantics of Constraint Formulae)
The satisfaction relation \( \models \) is inductively defined over the model \( M^\Sigma \), valuations \( \alpha : \text{Var} \to M^\Sigma \), and formulae \( \phi \in \text{CF}(\Sigma, X) \), as follows:

- \( M^\Sigma, \alpha \models b \) iff \( \alpha(b) = \text{true} \), where \( b \in T_{\Sigma, \text{Bool}}(X) \);
- \( M^\Sigma, \alpha \models t_1 = ? t_2 \) iff \( \alpha(t_1) = \alpha(t_2) \);
- \( M^\Sigma, \alpha \models (\exists x)\phi \) iff there exists \( a \in M_s \) (where \( x \in X_s \)) such that \( M^\sigma, \alpha[x \mapsto a] \models \phi \);
Constrained Terms

Definition (Constrained Terms)

A constrained term $\varphi$ of sort $s \in S$ is a pair $(t \mid \phi)$ with $t \in T_{\Sigma,s}(X)$ and $\phi \in CF(\Sigma, X)$.
Constrained Terms

Definition (Constrained Terms)
A *constrained term* \( \varphi \) of sort \( s \in S \) is a pair \( (t \mid \phi) \) with \( t \in T_{\Sigma,s}(X) \) and \( \phi \in CF(\Sigma, X) \).

Definition (Valuation Semantics of Constrained Terms)
The *valuation semantics* of a constrained term \( (t \mid \phi) \) is the set of valuations\( \| (t \mid \phi) \| \triangleq \{ \alpha : X \rightarrow M^\Sigma \mid M^\Sigma, \alpha \models \phi \} \).
Constrained Terms

Definition (Constrained Terms)
A constrained term $\varphi$ of sort $s \in S$ is a pair $(t \mid \phi)$ with $t \in T_{\Sigma,s}(X)$ and $\phi \in CF(\Sigma, X)$.

Definition (Valuation Semantics of Constrained Terms)
The valuation semantics of a constrained term $(t \mid \phi)$ is the set of valuations $\llbracket (t \mid \phi) \rrbracket \triangleq \{ \alpha : X \rightarrow M^\Sigma \mid M^\Sigma, \alpha \models \phi \}$.

Definition (State Predicate Semantics of Constrained Terms)
The state predicate semantics of a constrained term is defined by

$$\llbracket (t \mid \phi) \rrbracket \triangleq \{ \alpha(t) \mid \alpha \in \llbracket (t \mid \phi) \rrbracket \}.$$
Plan

Summary

A Coinductive Definition of Reachability Properties

Constrained Terms

Constrained Rule Systems

Proving Reachability Properties

Tool Demo

Conclusions
Constrained Rule Systems

Definition (Constrained Rule Systems)

A constrained rule is a tuple \((l, r, \phi)\), often written as \(l \rightarrow r \text{ if } \phi\), where \(l, r\) are terms in \(T_\Sigma(X)\) having sorts in the same connected component, and \(\phi \in \text{CF}(\Sigma, X)\).

A constrained rule system \(\mathcal{R}\) is a set of rules. \(\mathcal{R}\) defines a transition relation \(\rightsquigarrow_\mathcal{R}\) on \(M^\Sigma\) as follows: \(t \rightsquigarrow_\mathcal{R} t'\) iff there exist a rule \(l \rightarrow r \text{ if } \phi\) in \(\mathcal{R}\), a context \(c[\cdot]\), and a valuation \(\alpha : X \rightarrow M^\Sigma\) such that \(t = \alpha(c[l]), t' = \alpha(c[r])\) and \(M^\Sigma, \alpha \models \phi\).
Reachability Properties

Definition (Reachability Properties of Constrained Rule Systems)

A *reachability formula* is a pair of constrained terms written as \( \varphi \Rightarrow \varphi' \), which may share some variables. We say that a constrained rule system \( \mathcal{R} \) *demonically satisfies* \( \varphi \Rightarrow \varphi' \), and write

\[
\mathcal{R} \models^\forall \varphi \Rightarrow \varphi'
\]

iff \((M^\Sigma, \simto_{\mathcal{R}}) \models^\forall \,[\sigma(\varphi)] \Rightarrow [\sigma(\varphi')]\) for each substitution \(\sigma : \text{var}(\varphi) \cap \text{var}(\varphi') \rightarrow M^\Sigma\).
Derivatives of Constrained Terms

Definition (Derivatives of Constrained Terms)

The set of derivatives of a constrained term $\varphi \triangleq (t \mid \phi)$ w.r.t. a rule $l \rightarrow r$ if $\phi_{lr}$ is

$$\Delta_{l,r,\phi_{lr}}(\varphi) \triangleq \{(c[r] \mid \phi') \mid M^\Sigma \models \phi' \leftrightarrow (\phi \land t = ? c[l] \land \phi_{lr}),$$

$$c[\cdot] \text{ an appropriate context,}$$

$$\phi' \text{ is satisfiable}\} \quad (1)$$

where the variables in $l \rightarrow r$ if $\phi$ are renamed such that $\text{var}(l \rightarrow r \text{ if } \phi)$ and $\text{var}(\varphi)$ are disjoint.

If $\mathcal{R}$ is a set of rules, then $\Delta_{\mathcal{R}}(\varphi) = \bigcup_{(l,r,\phi_{lr}) \in \mathcal{R}} \Delta_{l,r,\phi_{lr}}(\varphi)$. A constrained term $\varphi$ is $\mathcal{R}$-derivable if $\Delta_{\mathcal{R}}(\varphi) \neq \emptyset$. 
Derivatives of Constrained Terms

Definition (Derivatives of Constrained Terms)

The set of derivatives of a constrained term \( \varphi \triangleq (t \mid \phi) \) w.r.t. a rule \( l \rightarrow r \) if \( \phi_{lr} \) is

\[
\Delta_{l,r,\phi_{lr}}(\varphi) \triangleq \{ (c[r] \mid \phi') \mid M^\Sigma \models \phi' \iff (\phi \land t =^? c[l] \land \phi_{lr}), \\
c[\cdot] \text{ an appropriate context}, \\
\phi' \text{ is satisfiable} \}
\]  

where the variables in \( l \rightarrow r \) if \( \phi \) are renamed such that \( \text{var}(l \rightarrow r \text{ if } \phi) \) and \( \text{var}(\varphi) \) are disjoint.

If \( \mathcal{R} \) is a set of rules, then \( \Delta_{\mathcal{R}}(\varphi) = \bigcup_{(l,r,\phi_{lr}) \in \mathcal{R}} \Delta_{l,r,\phi_{lr}}(\varphi) \). A constrained term \( \varphi \) is \( \mathcal{R} \)-derivable if \( \Delta_{\mathcal{R}}(\varphi) \neq \emptyset \).

Let \( \varphi \triangleq (t \mid \phi) \), \( \mathcal{R} \) a constrained rule system, and \( (M^\Sigma, \leadsto_{\mathcal{R}}) \) the transition system defined by \( \mathcal{R} \). Then \( \lfloor \Delta_{\mathcal{R}}(\varphi) \rfloor = \partial(\lfloor \varphi \rfloor) \).
Example

\[ S = \{ \begin{align*} 
    & \text{init}(n) \rightarrow \text{loop}(0, n) \text{ if true,} \\
    & \text{loop}(s, i) \rightarrow \text{loop}(s + i, i - 1) \text{ if } i > 0 \\
    & \text{loop}(s, 0) \rightarrow \text{done}(s) \text{ if true} \end{align*} \} \]
Example

\[
S = \{ \\text{init}(n) \rightarrow \text{loop}(0, n) \text{ if true}, \\
\text{loop}(s, i) \rightarrow \text{loop}(s + i, i - 1) \text{ if } i > 0 \\
\text{loop}(s, 0) \rightarrow \text{done}(s) \text{ if true} \}
\]

\[
\Delta_R((\text{loop}(s, n) \mid \text{true})) = \{ (\text{loop}(s + n, n - 1) \mid n > 0), \\
(\text{done}(s) \mid n = 0) \}
\]
Plan

Summary

A Coinductive Definition of Reachability Properties

Constrained Terms

Constrained Rule Systems

Proving Reachability Properties

Tool Demo

Conclusions
Proof System for Symbolic Execution

Figure: The DSTEP(\( \mathcal{R} \)) Proof System

\[ (t \mid \phi) \Rightarrow \varphi' \]  
\[ (t'' \mid \phi'' \land \neg \phi''') \Rightarrow (t' \mid \phi') \]
\[ (t \mid \phi) \Rightarrow (t' \mid \phi') \]
\[ (t'' \mid \phi'') \Rightarrow \varphi' \equiv (t \mid \phi) \Rightarrow \varphi', \text{ and} \]
\[ M^\Sigma \models \phi''' \leftrightarrow (\exists X)(t'' =? t' \land \phi'), \text{ and} \]
\[ X \triangleq \text{var}(t', \phi') \setminus \text{var}(t'', \phi'') \]

\[ \{(t^i \mid \phi^j) \Rightarrow \varphi' \mid (t^i \mid \phi^j) \in \Delta_\mathcal{R}((t'' \mid \phi''))\} \]
\[ (t \mid \phi) \Rightarrow \varphi' \]

\[ (t \mid \phi) \ \mathcal{R}-\text{derivable, and} \]
\[ (t'' \mid \phi'') \Rightarrow \varphi' \equiv (t \mid \phi) \Rightarrow \varphi', \text{ and} \]
\[ \phi'' \land \land \left\{ \neg (\exists Y)\phi^j \mid (t^i \mid \phi^j) \in \Delta_\mathcal{R}((t'' \mid \phi'')), \right. \]
\[ Y \triangleq \text{var}(t^j, \phi^j) \setminus \text{var}(t'', \phi'') \} \]
\[ \text{not satisfiable} \]
Let $\mathcal{R}$ be a constrained rule system. Then $\mathcal{R} \models \forall \nu \text{DSTEP}(\mathcal{R})$. 
Soundness

Let $\mathcal{R}$ be a constrained rule system. Then $\mathcal{R} \models \forall \nu \overline{\text{DSTEP}}(\mathcal{R})$. Not terribly useful!
Circularity

Definition (Demonic circular coinduction)

Let $G$ be a finite set reachability formulae. Then the set of rules $DCC(R, G)$ consists of $DSTEP(R)$ together with

\[
\frac{(t'_{\varsigma} \mid \phi'_{\varsigma} \land \phi \land \phi'')}{
(t \mid \phi \land \neg \phi'') \Rightarrow \phi'}
\]

\[
\frac{(t \mid \phi) \Rightarrow \phi'}{
\text{[circ]} \ M^\Sigma \models \phi' \iff (\exists \text{var}(t_{\varsigma}, \phi_{\varsigma}))(t = ? t_{\varsigma} \land \phi_{\varsigma})
\text{(t}_{\varsigma} \mid \phi_{\varsigma}) \Rightarrow (t'_{\varsigma} \mid \phi'_{\varsigma}) \in G
\}
\]

where the variables in $(t_{\varsigma} \mid \phi_{\varsigma}) \Rightarrow (t'_{\varsigma} \mid \phi'_{\varsigma})$ are renamed such that $\text{var}(t_{\varsigma}, \phi_{\varsigma}) \cap \text{var}(t, \phi) = \emptyset$. 
Circularity

**Definition (Demonic circular coinduction)**

Let $G$ be a finite set reachability formulae. Then the set of rules $\text{DCC}(\mathcal{R}, G)$ consists of $\text{DSTEP}(\mathcal{R})$ together with

\[
\begin{align*}
(t'_c | \phi'_c \land \phi \land \phi'') & \Rightarrow \varphi', \\
(t | \phi \land \neg \phi'') & \Rightarrow \varphi' \quad \text{[circ]} \\
(t | \phi) & \Rightarrow \varphi' \\
M^\Sigma \models \phi'' & \iff (\exists \text{var}(t_c, \phi_c))(t = t'_c \land \phi_c) \\
(t_c | \phi_c) & \Rightarrow (t'_c | \phi'_c) \in G
\end{align*}
\]

where the variables in $(t_c | \phi_c) \Rightarrow (t'_c | \phi'_c)$ are renamed such that $\text{var}(t_c, \phi_c) \cap \text{var}(t, \phi) = \emptyset$.

**Definition**

Let $PT$ be a proof tree of $\varphi \Rightarrow \varphi'$ under $\text{DCC}(\mathcal{R}, G)$. A [circ] node in $PT$ is *guarded* iff it has as ancestor a [der\forall] node. $PT$ is *guarded* iff all its [circ] nodes are guarded.
Soundness of Circularity

Definition
We write $(R, G) \vdash^\forall \varphi \Rightarrow \varphi'$ iff there is a proof tree of $\varphi \Rightarrow \varphi'$ under $DCC(R, G)$ that is guarded. If $F$ is a set of reachability formulae, we write $(R, G) \vdash^\forall F$ iff $(R, G) \vdash^\forall \varphi \Rightarrow \varphi'$ for all $\varphi \Rightarrow \varphi' \in F$. 
Soundness of Circularity

**Definition**
We write \((\mathcal{R}, G) \vdash_{\forall} \varphi \Rightarrow \varphi'\) iff there is a proof tree of \(\varphi \Rightarrow \varphi'\) under \(\text{DCC}(\mathcal{R}, G)\) that is guarded. If \(F\) is a set of reachability formulae, we write \((\mathcal{R}, G) \vdash_{\forall} F\) iff \((\mathcal{R}, G) \vdash_{\forall} \varphi \Rightarrow \varphi'\) for all \(\varphi \Rightarrow \varphi' \in F\).

[Circularity Principle]
Let \(\mathcal{R}\) be a constrained rule system and \(G\) a set of goals. If \((\mathcal{R}, G) \vdash_{\forall} G\) then \(\mathcal{R} \vdash_{\forall} G\).
Demo

http://github.com/ciobaca/rmt/
Conclusions

1. cleaner semantics of constrained rewrite systems;
2. coinductive approach to reachability properties;
3. coinductive proof system for reachability formulae inspired from our previous approaches to partial program correctness [1, 2];
4. implementation.
