A Constructor-Based Reachability Logic for Rewrite Theories

Stephen Skeirik, Andrei Stefanescu, Jose Meseguer

October 10th, 2017

Outline

Introduction

Origins of Reachability Logic

Reachability Logic (RL) was originally proposed by Rosu et. al to verify programs in $\mathbb{K}$

- based on a rewriting logic (RWL) definition of language $\mathscr{L}$'s semantics
- generalizes both Hoare Logic and Separation Logic
- language-generic: a prover can be generated for each language $\mathscr{L}$ from its rewriting logic semantics $\mathcal{R}_{\mathscr{L}}$

Introduction

From Language-generic to Theory-generic RL

This work addresses the following open problems:

1. Can we develop a reachability logic for general rewrite theories, i.e. $RL(\mathcal{R})$ generalizing $RL(\mathcal{R}_{\mathscr{L}})$?

   ...will allow us to move from verifying *code* to verifying *distributed system designs*

2. How can we *maximize* automation in RL proofs?

3. How can we use RL to prove *invariants* (*invariant paradox*)?

Introduction

From Language-generic to Theory-generic RL

We address questions (1)-(3) via:

1. developing a new RWL-theory based RL *semantics* and very simple *proof system*,
2. utilizing RWL concepts (e.g. *constructors*, *variants*) in our proof system as well as a *equational-theory-generic* SMT solver as a backend based on *variant satisfiability*
3. and applying an appropriate RWL *theory transformation* to prove *invariants*.

The mutual exclusion protocol QLOCK has five rewrite rules:

$$
\begin{array}{lllllllll}
n2w: & < n\ i \mid w & \mid c & \mid q & > & \rightarrow & < n & \mid w\ i \mid c & \mid q\ ;\ i > \\
w2c: & < n & \mid w\ i \mid c & \mid i\ ;\ q > & & \rightarrow & < n & \mid w & \mid c\ i \mid i\ ;\ q > \\
c2n: & < n & \mid w & \mid c\ i \mid i\ ;\ q > & & \rightarrow & < n\ i & \mid w & \mid c \mid q > \\
join: & < n & \mid w & \mid c & \mid q & > & \rightarrow & < n\ i & \mid w \mid c \mid q > \ if\ \phi \\
exit: & < n\ i & \mid w & \mid c & \mid q & > & \rightarrow & < n & \mid w \mid c \mid q >
\end{array}
$$

where $\phi \equiv dup(n\ w\ c\ i) \neq tt$. QLOCK's specification is $\mathcal{R}_{\mathsf{QLOCK}} = (\Sigma, E \cup B, R)$ with R the above rules, B the axioms ACU for _ _ and A for _;_ and, E the equation $dup(s\ s\ s') = tt$.

Outline

Reachability Logic Semantics

Constrained Constructor Patterns

Definition

Let $(\Sigma, B, \vec{E})$ be sufficiently complete w.r.t. constructors Ω.
A *constrained constructor pattern* is a pair: $u \mid \varphi$
such that $u \in T_\Omega(X) \wedge \varphi \in QFForm(\Sigma)$

The set $PatPred(\Omega, \Sigma)$ contains $\bot$ and all constrained constructor patterns, and is closed under $(\vee)$ and $(\wedge)$

The *semantics* of predicate A is $[\![A]\!] \subseteq C_{\Sigma/E,B}$ where:

1. $[\![\bot]\!] = \emptyset$
2. $[\![u \mid \varphi]\!] = \{[(u\rho)!]_{B_\Omega} \in C_{\Sigma/E,B} \mid \rho \in [X \to T_\Omega] \wedge E \cup B \models \varphi\rho\}$.
3. $[\![A \vee B]\!] = [\![A]\!] \cup [\![B]\!]$
4. $[\![A \wedge B]\!] = [\![A]\!] \cap [\![B]\!]$

Definition

Given patterns $u \mid \phi$ and $v_i \mid \psi_i$, a *reachability formula* has form:

$$u \mid \varphi \longrightarrow^{\circledast} \bigvee_i v_i \mid \psi_i$$

Example.

1. $< n \mid w \mid c \mid q > \mid dup(n\ w\ c) \neq tt$
2. $< n \mid w \mid c \mid q > \mid dup(n\ w\ c) \neq tt \longrightarrow^{\circledast}$
   $< n' \mid w' \mid c' \mid q' > \mid dup(n'\ w'\ c') \neq tt$

Reachability Logic Semantics

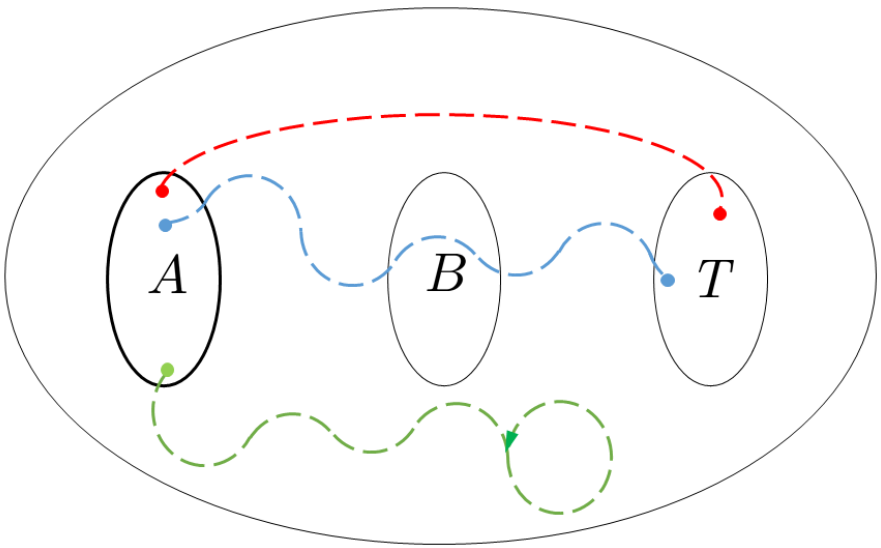
Intuitive Semantics

Q: What does the relation $A \longrightarrow^{\circledast} B$ mean?

A: Suppose we have:

(1) a rewrite theory $\mathcal{R}$
(2) pattern fomulas A, B
(3) and terminating states T

Then $A \longrightarrow^{\circledast} B$ means:
for each state $[t] \in [\![A]\!]$
and rewrite path p from $[t]$,
either: (1) p crosses $[\![B]\!]$ or
(2) p is infinite



- - - indicates counterex.
- - - satisfies $A \rightarrow^{\circledast} B$
- - - *vacuously* satisfies

Reachability Logic Semantics

Formal Semantics

Definition

Let $\mathcal{R} = (\Sigma, E \cup B, R)$ have (a) good *executability conditions*, (b) *constructor* subsignature Ω, (c) and *top sort* $State$ of states. Let $\mathcal{C}_{\mathcal{R}}$ denote the *canonical reachability model*.

$\mathcal{R} \models A \longrightarrow^{\circledast} B$ iff:

For each *concrete state* $[u_0] \in \mathcal{C}_{\mathcal{R},State} \cap [\![A]\!]$ and;
terminating sequence $[u_0] \rightarrow_{\mathcal{R}} [u_1] \cdots [u_{n-1}] \rightarrow_{\mathcal{R}} [u_n]$
There exists $0 \leq j \leq n$ with $[u_j] \in [\![B]\!]$

N.B.: expressible as LTL formula $A \rightarrow (\Box enabled) \vee \Diamond B$

Outline

The Invariant Paradox

Introduction

Recall our example theory QLOCK.

Q: How to express *mutual exclusion* invariant by $A \longrightarrow^{\circledast} B$?

A: Since:

1. $A \longrightarrow^{\circledast} B$ just means $A \rightarrow (\Box enabled) \vee \Diamond B$,
2. and QLOCK is *never terminating*,

then *all* formulas $A \longrightarrow^{\circledast} B$ are satisfied, so we cannot.

(Paradox!!).

The Invariant Paradox

Solving the Invariant Paradox (I)

Recall the structure of the rewrite rules in QLOCK:

$$n2w : < n\ i \mid w \mid c \mid q > \rightarrow < n \mid w\ i \mid c \mid q\ ;\ i >$$

Each rule is topped with a $State$ constructor $< \_ >$. Let's add rule:

$$stop : <\ n \mid w \mid c \mid q\ > \rightarrow [\ n \mid w\ i \mid c \mid q\ ]$$

Note that the $stop$ rule can *terminate* from any state in the combined theory QLOCK-stop.

If $B = \langle \ldots \rangle \mid \varphi$, let $[B]$ denote the predicate $[B] = [\ldots] \mid \varphi$.

Fact. B is an *invariant* from initial states S_0 in QLOCK iff $S_0 \longrightarrow^{\circledast} [B]$ holds in QLOCK-stop.

The Invariant Paradox

Solving the Invariant Paradox (II)

Let $\mathcal{R}$ be a rewrite theory; assume a single $State$ constructor $\langle \_, \ldots, \_ \rangle : w \to State$ and all rules have terms of sort $State$.

Let $\mathcal{R}_{stop}$ extend $\mathcal{R}$ by adding: (i) *fresh* $[\_, \ldots, \_] : w \to State$, and (ii) a *stop rule* $\langle \vec{x} : w \rangle \to [\vec{x} : w]$. Then:

Theorem

B is an invariant for $\mathcal{R}$ from S_0 iff $S_0 \longrightarrow^{\circledast} [B]$ holds in $\mathcal{R}_{stop}$.

Corollary

If $[\![S_0]\!] \subseteq [\![B]\!]$ and $B \longrightarrow^{\circledast} [B]$ holds in $\mathcal{R}_{stop}$, then B is an invariant for $\mathcal{R}$ from initial sates S_0.

Example. Mutual exclusion in QLOCK can be given by $Mutex = \langle n \mid w \mid i \mid i\,;q \rangle \,|\, dup(n\ w\ c\ i) \neq tt \vee \langle n \mid w \mid \emptyset \mid q \rangle \,|\, dup(n\ w\ c) \neq tt$.

Prove: (i) $[\![\langle n \mid \emptyset \mid \emptyset \mid \mathit{nil} \rangle]\!] \subseteq [\![Mutex]\!]$ (ii) $Mutex \longrightarrow^{\circledast} [Mutex]$.

Outline

Inference System

Introduction (I)

Q: Then given RWL theory $\mathcal{R}$, how do we prove $A \longrightarrow^{\circledast} B$?

A: Perhaps surprisingly, two proof rules are enough:

- A rule that traces *rewrite steps* of *symbolic* states in $\mathcal{R}$
- A rule that internalizes *terminating-path-length induction* on $\mathcal{R}$

We call these two rules:

- *Step+Subsumption*
- *Axiom*

Inference System

Introduction (II)

The key ideas are:

1. Proving $A \longrightarrow^{\circledast} B$ may require some *auxiliary lemmas*;
   Let $\mathcal{C}$ denote the formula $A \longrightarrow^{\circledast} B$ plus these lemmas
2. For each formula in $\mathcal{C}$, start with labeled sequents:
   $[\emptyset,\ \mathcal{C}]\ \vdash_T\ u \mid \varphi \longrightarrow^{\circledast} \bigvee_i v_i \mid \psi_i$
3. 1^{st} part ($\emptyset$) is formulas to be assumed as *axioms* (none);
4. 2^{nd} part ($\mathcal{C}$) is formulas to prove that *cannot yet be assumed*
5. the *Step+Subsumption* rule allows us to *inductively assume* $\mathcal{C}$ after a rewrite step with rules $R = \{l_j \to r_j\ if\ \phi_j\}$.

Reachability Logic

Proof Rules (I): Step+Subsumption Rule

$$\frac{\bigwedge_{(j,\alpha)\in \text{UNIFY}(u|\varphi', R)} [\mathcal{A}\cup\mathcal{C},\ \emptyset]\ \vdash_T\ (r_j \mid \varphi' \wedge \phi_j)\alpha \longrightarrow^{\circledast} \bigvee_i (v_i \mid \psi_i)\alpha}{[\mathcal{A},\ \mathcal{C}]\ \vdash_T\ u \mid \varphi \longrightarrow^{\circledast} \bigvee_i v_i \mid \psi_i}$$

with $\varphi' = \varphi \wedge \bigwedge_{(i,\beta)\in \text{MATCH}(u,\{v_i\})} \neg(\psi_i\beta)$ and $R = \{l_j \to r_j \ \mathit{if}\ \phi_j\}$

Note.

- proof rule performs all possible *narrowing* steps with rules R
- goals $u \mid \phi \longrightarrow^{\circledast} B$ with unsatisfiable ϕ are *implicitly removed*

Reachability Logic

Proof Rules (II): The Axiom Rule

$$\frac{\bigwedge_j [\{u' \mid \varphi' \longrightarrow^{\circledast} \bigvee_j v'_j \mid \psi'_j\} \cup \mathcal{A},\ \emptyset] \ \vdash_T \ v'_j\alpha \mid \varphi \wedge \psi'_j\alpha \longrightarrow^{\circledast} \bigvee_i v_i \mid \psi_i}{[\{u' \mid \varphi' \longrightarrow^{\circledast} \bigvee_j v'_j \mid \psi'_j\} \cup \mathcal{A},\ \emptyset] \ \vdash_T \ u \mid \varphi \longrightarrow^{\circledast} \bigvee_i v_i \mid \psi_i}$$

where $\exists \alpha$ with $u =_{E_\Omega \cup B_\Omega} u'\alpha$ and $\mathcal{T}_{\Sigma/E \cup B} \models \varphi \Rightarrow \varphi'\alpha$

Theorem

(Soundness) Let $\mathcal{R}$ be a rewrite theory, and $\mathcal{C}$ a finite set of reachability formulas. If $\mathcal{R}$ proves $[\emptyset,\ \mathcal{C}]\ \vdash_T\ \mathcal{C}$ then $\mathcal{R} \models^{\forall}_T \mathcal{C}$

Outline

Implementation and Case Studies

Reflective Implementation

The proof system has been implemented in *Maude*. Some notes:

1. RWL is *reflective*, Maude's *META-LEVEL* library support was used which supports reasoning over RWL *theories* and *terms*
2. Maude's built-in support for *narrowing* modulo axioms was used to compute successors in the RWL theory $\mathcal{R}$
3. An implementation of a *variant satisfiability*-based, *theory-generic* SMT solver was used to discharge satisfiability and validity proof obligations

Implementation and Case Studies

Case Studies

Example	Description of the System/Property
Choice	Nondeterministically throws away elements from a multiset/eventually only one element left
Comm. Protocol 1	Simple communication protocol/received data is always a prefix of the data to be sent
Comm. Protocol 2	Fault-tolerant communication protocol/all data is eventually received in-order
Dijkstra	Dijkstra's mutual exclusion alg./mutual exclusion
Fixed-Size Token Ring	2-Token ring mutual exclusion alg./mutual exclusion
QLOCK	QLOCK mutual exclusion alg./mutual exclusion
Readers/Writers	Readers-writers mutual exclusion alg./mutual exclusion
Lamport's Bakery	Unbounded Lamport's bakery/mutual exclusion
Thermostat	Open system that dynamically responds to temperature/temperature remains in preset bounds

Implementation and Case Studies

Example Proof Fragment

$$T_1 \equiv \left\{ \frac{}{[\mathcal{C}, \emptyset] \vdash_{[]} [n^3 \mid w^3 \mid \emptyset \mid q^3] \mid dup(n'' \, w' \, p) \neq tt \wedge dup(n^3 \, w^3) \neq tt \rightarrow^{\circledast} [Mutex_1] \vee [Mutex_2]} \; \mathsf{sub}(P_1, \alpha) \right.$$

$$\frac{\dfrac{T_1 \qquad T_2}{\ldots \; [\mathcal{C}, \emptyset] \vdash_{[]} < n'' \mid w' \; p \mid \emptyset \mid q' > \mid dup(n'' \, w' \, p) \neq tt \rightarrow^{\circledast} [Mutex_1] \vee [Mutex_2] \; \ldots} \; \mathsf{axiom}(G_2, \alpha)}{[\emptyset, \mathcal{C}] \vdash_{[]} < n' \mid w' \mid \emptyset \mid q' > \mid dup(n' \; w') \neq tt \rightarrow^{\circledast} [Mutex_1] \vee [Mutex_2]} \; \mathsf{step}(n2w, \theta)$$

where $G_i \equiv Mutex_i \longrightarrow^{\circledast} [Mutex]$, $C \equiv \{G_1, G_2\}$

Outline

Conclusions

We have presented our new theory and implementation of a RL semantics and inference system where:

1. our system is *rewrite-theory-generic*, so it can be applied to analyze distributed system designs
2. our implementation uses a *theory-generic*, *variant satisfiability* SMT solver underneath
3. we applied RWL theory *transformations* in order to specify and verify *invariants*

Future Work

At this point, there are a two clear directions for future work:

1. our *variant satisfiability* implementation currently supports rewrite theories whose equational fragment is decidable—we are developing heuristics for *undecidable* theories
2. we are developing *larger*, *more interesting* case studies to provide further validation for our reachability logic tool

The End

Any Questions?