Program Equivalence: An Interactive Relational Separation Logic Prover Implemented in Maude

Andrei Alin Corodescu

Scientific Coordinator:
Conf. Dr. Ciobaca Stefan

June 30, 2018

Outline

The Problem

Problem Description

Reducing the gap between theoretical and practical aspects of formal program equivalence verification to increase software quality by making robust methods of verification accessible and easy to use.

The Problem

Problem Description

Reducing the gap between theoretical and practical aspects of formal program equivalence verification to increase software quality by making robust methods of verification accessible and easy to use.

Difficulties

- Representing the theoretical concepts
- Computationally-hard problems
- User experience

Outline

1. The Problem
2. Technologies and theoretical concepts
   - Relational Separation Logic
   - Maude
3. Prover Implementation
   - Goals
   - Axiom Recognition
   - Automatic Proof of Implications

Relational Separation Logic

- Helps reason about how two programs are related
- Hoare Quadruples : $\{R\} \begin{array}{c} C \\ C\prime \end{array} \{S\}$
- Proof rules : $\dfrac{R \Rightarrow R_1 \qquad \{R_1\} \begin{array}{c} C \\ C\prime \end{array} \{S_1\} \qquad S_1 \Rightarrow S}{\{R\} \begin{array}{c} C \\ C\prime \end{array} \{S\}}$

Outline

Maude

- Based on Rewriting logic
- Natural Representation of logics
- Powerful meta language applications

Example

```
rl [Consequence] : { R } C1 -- C2  { S } => ((R => R1) <> ({
    R1} C1 -- C2 {S1})) <> (S1 => S) [nonexec] .
```

Outline

1. The Problem
2. Technologies and theoretical concepts
   - Relational Separation Logic
   - Maude
3. Prover Implementation
   - Goals
   - Axiom Recognition
   - Automatic Proof of Implications

Goals

- The central concept of the prover is Goal, which represents something to be proven.
- New goals are generated by applying proof rules to existing goals.
- A Goal is consumed if it is an matched with an *axiom*, is *manually admitted* by the user, is *automatically proven* (in case of implications) or it is *replaced* by the goals generated by applying a proof rule.
- Goals are stored in a GoalStack structure

Outline

Axiom Recognition

- Simple goals that match axioms are automatically admitted by the prover.
- Equality between variables is interpreted before matching axioms
- Takes place at the meta level

Outline

1. The Problem
2. Technologies and theoretical concepts
   - Relational Separation Logic
   - Maude
3. Prover Implementation
   - Goals
   - Axiom Recognition
   - Automatic Proof of Implications

Automatic Proof of Implications

- Uses the search functionality of Maude
- Searches for a series of rewrites from `R => S` to `true`
- Rewrite rules denoting relation equivalences and implications
- Takes place at the meta level

Demo

Conclusions

- New theoretical concepts
- New, different technology
- Deeper understanding by modelling and applying logics
- Shortcomings of Maude because of it's niche segment and narrow adoption