UNIVERSITATEA ”ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ

TITLU PROVIZORIU

ANDREI-ALIN CORODESCU

Sesiunea: *iulie, 2018*

Coordonator științific
Conf. Dr. Ciobâcă Ștefan

Contents

1 Introduction

Comparing programs or code fragments and studying their equivalence is part of every software engineer's activities when they are testing an alternative implementation for an existing solution, fixing bugs, launching new product versions, etc . Naturally, for every process completed by persons there are efforts being made in order to make it more efficient, less error-prone and, in the end, automate the process all together. Once such as a task is automated in software engineering, it can be included in the flow of any research or development phase. An example benefiting from a formal proof of program equivalence is compiler optimization, where the optimized code needs to be equivalent to the input one .

The present paper describes the development of an interactive tool for arguing how to programs are related, based on studied and previously used theoretical concepts and technologies which facilitate the implementation of those concepts .

The tool represents an implementation of Hoare Logic - which allows formal reasoning about a program - , along with 2 of its extensions, namely the Separation Logic (named Separation Logic from now on) and Relational Separation Logic (named Relational Logic from now on). The 2 extensions simplify the Hoare Logic proofs, mainly using the "*" connector, allowing for local reasoning of effects of statements in a program . The tool has been implemented in Maude, a high performance logical framework with powerful metalanguage applications which facilitate the implementations of executable environments for logics.

The tool is built as a CLI which helps argue how two programs are related using Relational Separation Logic specifications. As a consequence of the dependency of Relational Separation Logic on Separation Logic, proofs about single programs using the latter are also supported by the tool. The tool has been developed with extensibility in mind, the main desired extensions being concurrent programs support and automatic proofs.

The rest of the paper is organized as follows:

- Section 1 will describe the language supported by the tool and a brief description of the logics utilized for reasoning .
- Section 2 will describe in detail the building process of the tool and how it can be used, with an accent on the features offered by the Maude language.

- Section 3 will present some of the research done on subjects related to the theme of the paper and discuss how those could be integrated into the current solution.
- Section 4 will present the conclusions of the paper, regarding both the theoretical and technical aspects of the paper.

2 Contributions

Personal contributions to the realization of the project :

- Modelled the Relational Logic and Separational Logic using Maude equational and rewriting logic specifications .
- Executable semantics of a simple programming language using Maude
- Developed an interactive tool for reasoning about program behaviour using the aforementioned logics.
- Automation of some tasks which makes the tool more convenient to use .

3 Theoretical Foundations

În acest capitol voi prezenta pe scurt fundamentele teoretice ale prezentei lucrări.

3.1 Language

3.1.1 Storage Model

3.1.2 Syntax and Semantics of the language

3.2 Hoare Logic

3.3 Separational Logic

3.4 Relational Separation Logic

3.5 Rewriting Logic

4 Relational Separation Logic Interactive Prover

Ordinea subsectiunilor mai poate fi schimbata ulterior. Capitolul cel mai semnificativ din lucrare, va contine toate detaliile de implementare ale solutiei curente