

Abstract

1 Introduction

- Importance of autonomous systems and their verification
- The state explosion problem, summary of different approaches
- Our idea and contributions

2 Background

2.1 Logics for Multi-Agent Systems

- Multi-Agent Systems and Interpreted Systems
- Temporal Epistemic Logics: (LTL,) CTL, CTLK

2.2 Model Checking

- Explicit Approach to Model Checking
- Symbolic Model Checking and state-space representations

2.3 Knowledge Compilation

- BDDs: BDTs, reduction, canonicity up to reordering
- *A Knowledge Compilation Map*
- Negation Normal Forms and choice of SDDs for the project

2.4 SDDs

- Definitions and Construction
- Syntax and Semantics
- Canonicity
- OBDDs are SDDs

3 Technical Preliminaries

3.1 MCMAS Specifics

- *Some* implementation details (enough to understand the steps needed for replacing BDDs with SDDs).

- Variable allocation and how we keep track of it
- Description of the 4 variable orderings available
- ADDs to represent algebraic expressions

3.2 The SDD Package

- Development, summary of features
- Dynamic minimization and automatic garbage collection
- Algorithm for vtree search (and operations for navigating the space of vtrees)

4 Contributions

4.1 Implementation of a model checker based on SDDs

- Functionality and limitations

4.2 Some heuristics

(This is one of the most important sections - but heavily dependent on the next couple of weeks)

5 Evaluation

5.1 Models

- Description of models used for quantitative analysis

5.2 Comparison without dynamic variable reordering

- Comparison of various variable orderings with corresponding right-linear vtrees
- Comparison of various variable orderings with equivalent *dissections* of these orderings (= vtrees), leading to SDDs non-equivalent to BDDs
- Analysis of heuristics proposed in previous section

5.3 Comparison using dynamic minimization algorithms

- Experiments with different initial orderings and vtrees
- Analysis of heuristics proposed in previous section
- Comparison of different vtree minimization algorithms/settings (The SDD package allows the user to implement their own minimization function, or change the settings in the original functions. It would be interesting to find out what happens if we change the minimization thresholds for time and memory, or if we choose a different algorithm for searching the space of vtrees)

6 Conclusions and further work

6.1 Review

6.2 Future work

I will see what I don't have time to do before the deadline. Potential missing features will be counterexample/witness generation, and checking for deadlock or model overflow. Also being able to check ATL formulas. At some point I would like to have a go at implementing an ADD equivalent for SDDs.

7 Bibliography