Rheinisch-Westfälische Technische Hochschule Aachen
Chair for Software Modeling and Verification
Prof. Dr. Ir. Dr. h. c. Joost-Pieter Katoen

Master Thesis

Comparing Hierarchical and On-The-Fly Model Checking for Java Pointer Programs

Sally Chau
Matriculation Number 370584
May 2, 2019

First Reviewer: apl. Prof. Dr. Thomas Noll
Second Reviewer: Prof. Dr. Ir. Dr. h. c. Joost-Pieter Katoen
Supervisor: Christoph Matheja

Acknowledgement

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt und durch meine Unterschrift, dass die vorliegende Arbeit von mir selbstständig, ohne fremde Hilfe angefertigt worden ist. Inhalte und Passagen, die aus fremden Quellen stammen und direkt oder indirekt übernommen worden sind, wurden als solche kenntlich gemacht. Ferner versichere ich, dass ich keine andere, außer der im Literaturverzeichnis angegebenen Literatur verwendet habe. Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Bonn, den 28. September 2015, Sally Chau

Abstract

Contents

Chapter 1

Introduction

1.1 Attestor

1.2 Related Work

Chapter 2

Preliminaries

2.1 Notation

2.2 Hypergraphs

2.2.1 Hyperedge Replacement Grammars

2.3 Model Checking

2.3.1 Linear Temporal Logic

Linear Temporal Logic (LTL) can be used to describe properties of paths in a transition system S. LTL formulae mainly consist of three components: the boolean operators *negation* ($\neg$) and *conjunction* ($\wedge$), the temporal operators *next* ($\bigcirc$) and *until* (U), and a set of atomic proposition AP, which contains the state labels of a transition system. A state label is an assertion about the value of a state, e.g., "$i = 1$".

Definition 2.1 (Syntax of LTL [2])**.** *Given a set AP of atomic propositions with $a \in AP$,* LTL formulae *are recursively defined as*

$$\psi ::= \texttt{true} \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U} \varphi_2.$$

Definition 2.2 (Positive Normal Form [2])**.** *Given a set AP of atomic propositions with $a \in AP$, LTL formulae in* positive normal form *(PNF) are defined as*

$$\varphi ::= \texttt{true} \mid \texttt{false} \mid a \mid \neg a \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U} \varphi_2 \mid \varphi_1 \mathbf{R} \varphi_2.$$

2.3.2 LTL Model Checking

2.4 Recursive State Machines

Definition 2.3 (Recursive State Machine [1]). *A* recursive state machine *(RSM) A over a finite alphabet Σ is given by a tuple $(A_1, ..., A_k)$, where each* component state machine *(CSM) $A_i = (N_i \cup B_i, Y_i, En_i, Ex_i, \delta_i)$, $1 \leq i \leq k$, consists of*

- *a set N_i of* nodes *and a (disjoint) set B_i of* boxes*,*
- *a* labeling $Y_i : B_i \mapsto \{1, ..., k\}$ *that assigns to every box an index $j \in \{1, ..., k\}$ referring to one of the component state machines $A_1, ..., A_k$,*
- *a set of* entry nodes $En_i \subseteq N_i$*,*
- *a set of* exit nodes $Ex_i \subseteq N_i$*, and*
- *a* transition relation *δ_i, where transitions are of the form (u, σ, v), where*
  - *the source u is either a node of N_i or a pair (b, x), where b is a box in B_i and x is an exit node in Ex_j for $j = Y_i(b)$,*
  - *the label σ is in Σ, and*
  - *the destination v is either a node in N_i or a pair (b, e), where b is a box in B_i and e is an entry node in En_j for $j = Y_i(b)$.*

2.4.1 Semantics

This section describes the global relation between component state machines A_i of an RSM $A = (A_1, ..., A_k)$ in order to define its execution. A *global state* of an RSM consists of boxes and nodes of its CSMs.

Definition 2.4 ((Global) State [1]). *A* (global) state *of an RSM $A = (A_1, ..., A_k)$ is a tuple $(b_1, ..., b_r, u)$, where $b_1, ..., b_r$ are boxes and u is a node. The set Q of global states of A is B^*N, where $B = \bigcup_i B_i$ and $N = \bigcup_i N_i$. A state $(b_1, ..., b_r, u)$ with $b_i \in B_{j_i}$ for $1 \leq i \leq r$ and $u \in N_j$ is* well-formed *if $Y_{j_i}(b_i) = j_{i+1}$ for $1 \leq i < r$ and $Y_{j_r}(b_r) = j$.*

A well-formed state $(b_1, ..., b_r, u)$ of an RSM $A = (A_1, ..., A_k)$ corresponds to a path through the components A_j of A, where we enter component A_j via box b_r of component A_{j_r}.
In order to transition between global states of an RSM A, we require the notion of a *global transition relation* δ which enables us to not only transition between states within a CSM A_j as defined by its transition relation δ_j, but also between pairs of CSMs.

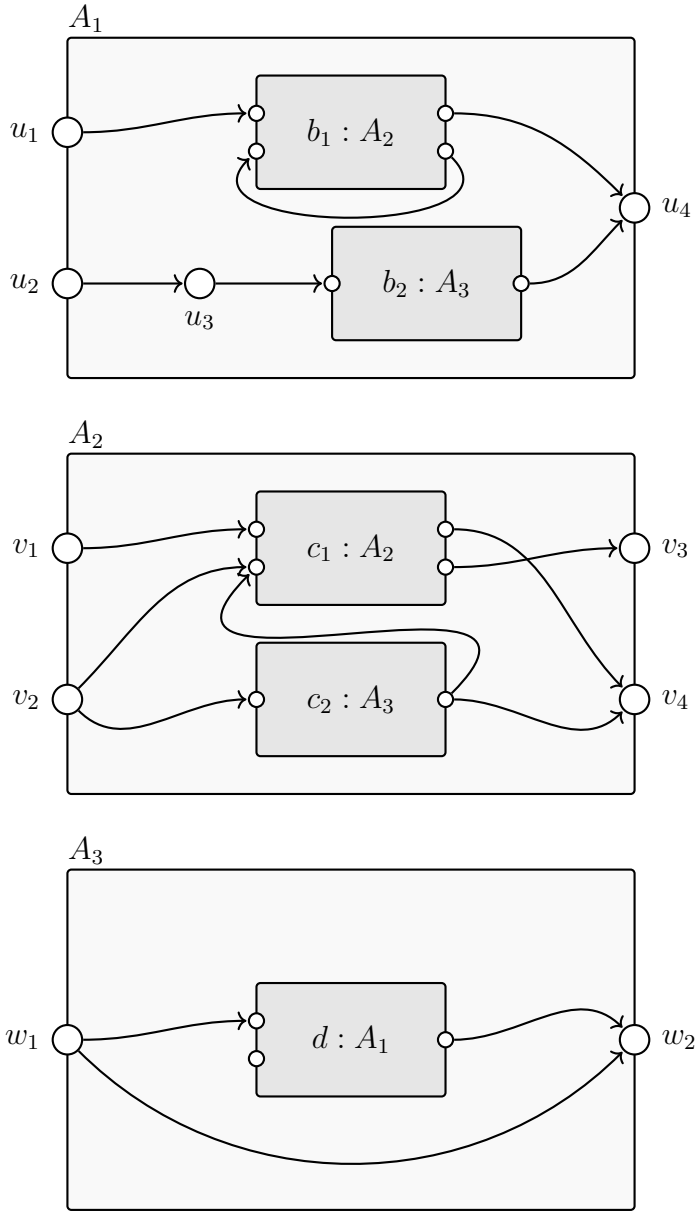

Figure 2.1: A sample recursive state machine. Adopted from [1].

Definition 2.5 ((Global) Transition Relation [1])**.** *Let $s = (b_1, ..., b_r, u) \in Q$ be a state with $u \in N_j$ and $b_r \in B_m$ for an RSM $A = (A_1, ..., A_k)$. A* (global) transition relation *δ for A defines $(s, \sigma, s') \in \delta$ if and only if one of the following holds:*

1. *$(u, \sigma, u') \in \delta_j$ for a node u' of A_j and $s' = (b_1, ..., b_r, u')$.*

2. *$(u, \sigma, (b', e)) \in \delta_j$ for a box b' of A_j and $s' = (b_1, ..., b_r, b', e)$.*

3. *u is an exit-node of A_j, $((b_r, u), \sigma, u') \in \delta_m$ for a node u' of A_m, and $s' = (b_1, ..., b_{r-1}, u')$.*

4. *u is an exit-node of A_j, $((b_r, u), \sigma, (b', e)) \in \delta_m$ for a box b' of A_m, and $s' = (b_1, ..., b_{r-1}, b', e)$.*

Definition 2.5 defines the possible kinds of transitions between global states $s, s' \in Q$ of an RSM A. Case 1 describes the scenario where the source and the destination states are both within the same component A_j, while case 2 depicts that a new component is entered via a box b' of A_j. Thus, the current node of the destination state s' is the entry-node e. Case 3 and 4 are both exiting component A_j via the exit-node u. While case 3 returns to component A_m, from where we entered A_j before, case 4 directly enters a new component via box b' of component A_m.
After defining the terms of global states and the global transition relation for an RSM A, we can summarize these components together with the finite alphabet Σ within the concept of a *labeled transition system* T_A, which encodes the execution of A.

Definition 2.6 (Labeled Transition System [1])**.** *For an RSM $A = (A_1, ..., A_k)$, the* labeled transition system *(LTS) $T_A = (Q, \Sigma, \delta)$ consists of*

- *a set of global states Q,*
- *a finite alphabet Σ, and*
- *a global transition relation δ.*

The LTS of an RSM A is also called the unfolding *of A.*

Chapter 3

Hierarchical Model Checking with Recursive State Machines

3.1 Algorithm

3.2 Implementation

3.3 Evaluation

Chapter 4

On-The-Fly Hierarchical Model Checking

4.1 Algorithm

4.2 Implementation

4.3 Evaluation

Chapter 5

Benchmarks

5.1 Experimental Setup

Describe Technical details here

5.2 Instances

Describe code examples and properties here

5.3 Result

Table of values

Chapter 6

Conclusion

6.1 Discussion

6.2 Outlook

- hierarchical failure trace and counter example generation, spuriosity - hybrid method between on-the-fly and RSM

Bibliography

[1] Alur, Rajeev, Kousha Etessami and Mihalis Yannakakis: *Analysis of recursive state machines*. In *International Conference on Computer Aided Verification*, pages 207–220. Springer, 2001.

[2] Baier, Christel and Joost-Pieter Katoen: *Principles of model checking*. MIT press, 2008.