

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  
April 25, 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

<b>1</b>	<b>Introduction</b>	<b>11</b>
1.1	Attestor . . . . .	11
1.2	Related Work . . . . .	11
<b>2</b>	<b>Linear Temporal Logic</b>	<b>12</b>
<b>3</b>	<b>Recursive State Machines</b>	<b>13</b>
<b>4</b>	<b>Hierarchical Model Checking</b>	<b>14</b>
<b>5</b>	<b>RSM Approach</b>	<b>15</b>
5.1	Implementation . . . . .	15
<b>6</b>	<b>On-the-fly Approach</b>	<b>16</b>
6.1	Implementation . . . . .	16
<b>7</b>	<b>Benchmarks</b>	<b>17</b>
<b>8</b>	<b>Conclusion and Future Work</b>	<b>18</b>
8.1	Possible Extensions . . . . .	18



# Chapter 1

## Introduction

### 1.1 Attestor

### 1.2 Related Work

## Chapter 2

# Linear Temporal Logic

## Chapter 3

# Recursive State Machines

**Definition 3.1** (Recursive State Machine). *Ein Spiel  $\mathcal{G}$  ist ein Tupel  $(\mathcal{N}, (\Sigma_i)_{i \in \mathcal{N}}, (U_i)_{i \in \mathcal{N}})$ , wobei  $\mathcal{N} = \{1, \dots, n\}$  die Menge der Spieler beschreibt,  $\Sigma_i$  den Strategieraum eines Spielers  $i \in \mathcal{N}$  und  $U_i : \Sigma_1 \times \dots \times \Sigma_n \rightarrow \mathbb{R}^n$  die Auszahlungsfunktion für einen Spieler  $i \in \mathcal{N}$  in einem Zustand  $S$ . Mit  $S = (S_1, \dots, S_n) \in \Sigma_1 \times \dots \times \Sigma_n$  beschreiben wir den Zustand eines Spieles, in dem Spieler  $i$  Strategie  $S_i \in \Sigma_i$  spielt.*

## Chapter 4

# Hierarchical Model Checking

# Chapter 5

## RSM Approach

### 5.1 Implementation

## Chapter 6

# On-the-fly Approach

### 6.1 Implementation



# Chapter 7

## Benchmarks

## Chapter 8

# Conclusion and Future Work

### 8.1 Possible Extensions