# Seminar
# Trends in Computer-Aided Verification
## Introduction

Christina Jansen Tim Lange Thomas Noll Kevin van der Pol

Software Modeling and Verification Group

RWTH AACHEN UNIVERSITY

noll@cs.rwth-aachen.de

Winter Semester 2014/15; 15 October, 2014

# Outline

# Formal Verification Methods

## Formal verification methods

- Rigorous, mathematically based techniques for the specification, development and verification of software and hardware systems
- Aim at improving correctness, reliability and robustness of such systems

# Formal Verification Methods

## Formal verification methods

- Rigorous, mathematically based techniques for the specification, development and verification of software and hardware systems
- Aim at improving correctness, reliability and robustness of such systems

## Classifications

- According to design phase
  - specification, implementation, testing, ...
- According to specification formalism
  - process algebras, timed automata, Markov chains, ...
- According to underlying mathematical theories
  - model checking, theorem proving, static analysis, ...

# Outline

# Goals

## Aims of this seminar

- Independent understanding of a scientific topic
- Acquiring, reading and understanding scientific literature
- Writing of your own report on this topic
- Oral presentation of your results

# Requirements on Report

## Your report

- Independent writing of a report of 15–20 pages
- Complete set of references to all consulted literature
- Correct citation of important literature
- Plagiarism: taking text blocks (from literature or web) without source indication causes immediate exclusion from this seminar
- Font size 12pt with "normal" page layout
- Language: German or English
- We expect the correct usage of spelling and grammar
  - $\geq$ 10 errors per page $\Longrightarrow$ abortion of correction

# Requirements on Talk

## Your talk

- Talk of about 45 minutes
- Focus your talk on the audience
- Descriptive slides:
  - $\leq$ 15 lines of text
  - use (base) colors in a useful manner
- Language: German or English
- No spelling mistakes please!
- Finish in time. Overtime is bad
- Ask for questions

# Final Preparations

## Preparation of your talk

- Setup laptop and projector ahead of time
- Use a (laser) pointer
- Number your slides
- Multiple copies: laptop, USB, web
- Have backup slides ready for expected questions

# Outline

# Important Dates

## Talks

The seminar will be held weekly on Tuesdays at 16:00 (?) starting end of November

- see http://moves.rwth-aachen.de/teaching/ws-1415/cav/

# Important Dates

## Talks

The seminar will be held weekly on Tuesdays at 16:00 (?) starting end of November

- see `http://moves.rwth-aachen.de/teaching/ws-1415/cav/`

## Deadlines

You are requested to adhere to the following firm deadlines:

- immediately: obtain the required literature from the web or library
- eight weeks before your talk: present a table of contents
- six weeks before your talk: preliminary version of your report
- four weeks before your talk: final version of your report
- two weeks before your talk: preliminary version of your slides
- one week before your talk: final version of your slides

# Important Dates

## Talks

The seminar will be held weekly on Tuesdays at 16:00 (?) starting end of November

- see `http://moves.rwth-aachen.de/teaching/ws-1415/cav/`

## Deadlines

You are requested to adhere to the following firm deadlines:

- immediately: obtain the required literature from the web or library
- eight weeks before your talk: present a table of contents
- six weeks before your talk: preliminary version of your report
- four weeks before your talk: final version of your report
- two weeks before your talk: preliminary version of your slides
- one week before your talk: final version of your slides

Missing a deadline causes immediate exclusion from the seminar

# Outline

# Selecting Your Topic

## Procedure

- You obtain(ed) a list of topics of this seminar.
- Indicate the preference of your topics (first, second, third).
- We do our best to find an adequate topic-student distribution.
- Disclaimer: no guarantee for an optimal solution.
- Your topic will be published on our website by 17 October.
- Please give language preference
  - unsure $\Longrightarrow$ German

# Selecting Your Topic

## Procedure

- You obtain(ed) a list of topics of this seminar.
- Indicate the preference of your topics (first, second, third).
- We do our best to find an adequate topic-student distribution.
- Disclaimer: no guarantee for an optimal solution.
- Your topic will be published on our website by 17 October.
- Please give language preference
  - unsure $\Longrightarrow$ German

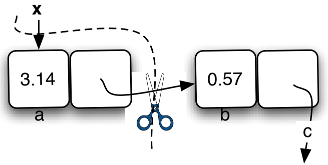
## Withdrawal

- You have up to three weeks to refrain from participating in this seminar.
- Later cancellation (by you or by us) causes a not passed for this seminar and reduces your (three) possibilities by one.

# Outline

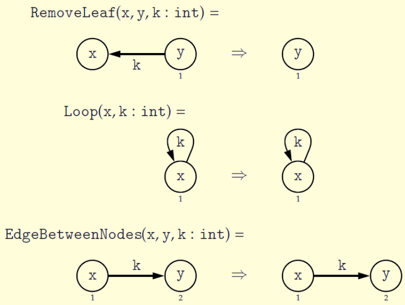
# 1: Permission Accounting in Separation Logic

(Richard Bornat, Cristiano Calcagno, Peter O'Hearn, and Matthew Parkinson)




- verification of concurrent heap-manipulating programs
- heap described by (Separation) logic formulae
- challenge adressed: shared memory
- permissions guarantee data-race freedom

```
main    = RemoveLeaf!; if NotNull then No else Yes.
NotNull = {Loop, EdgeBetweenNodes}.
```




- visual programming: GP
- programming language based on graph transformation
- specification: structural graph properties and label properties

# 3: Implicit Dynamic Frames
**(Jan Smans, Bart Jacobs, and Frank Piessens)**

*We wish to logically represent how the execution of a command changes the state without having to explicitly say how the command does not change the state.*

- verification of OO programs
- pre-/postcondition based
- program state as FO-formula
- enrich pre-/postconditions with access assertions ("the heap part a method is allowed to touch"")

# Outline

# 4: Grammar-Based Shape Analysis

(Oukseh Lee, Hongseok Yang, and Kwangkeun Yi)

**Verification of pointer programs**

- Infinite data domains
- Dynamic storage (de-)allocation

$\Rightarrow$ Requires abstraction techniques

**Heap:**

$\mathtt{x} \rightarrow o_1 \xrightarrow{\mathtt{n}} o_2 \xrightarrow{\mathtt{n}} o_3 \xrightarrow{\mathtt{n}} \mathtt{nil}$

**Shape graph:**

$\{\mathtt{x}\} \xrightarrow{\mathtt{n}} \emptyset \circlearrowleft \mathtt{n}$

**Shape analysis**

- Based on "instrumentation predicates"
- Yields partition of concrete nodes ("summary nodes")

$\Rightarrow$ Information loss in summary nodes

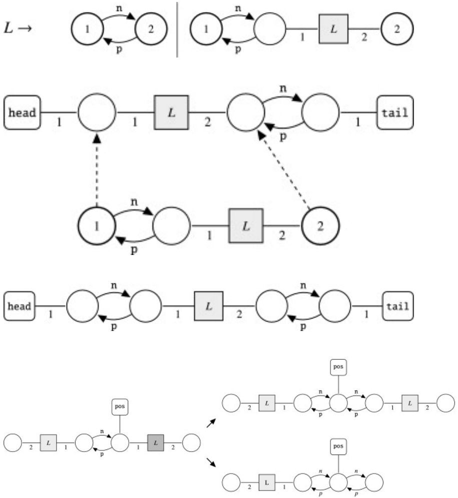
**Refinement: grammar annotations**

- Idea: associate tree grammars with summary nodes
- Applications: binomial heaps, Schorr-Waite tree traversal

# 5: Abstraction by Hyperedge Replacement Grammars
(Jonathan Heinen, Christina Jansen, Joost-Pieter Katoen, and Thomas Noll)

## Approach

- Hyperedge replacement grammars for modelling dynamic data structures
- Abstraction = backward application of rule
- Concretisation = forward application of rule
- Yields finite state space
- Makes standard model checking applicable
- Application: stackless tree traversal



# Outline

# 6: Software Model Checking via IC3
## (Alessandro Cimatti and Alberto Griggio)

### IC3

- Hardware MC approach
- construct inductive reachability sets
- look for violations of induction (CTI)
- trace back from CTIs
  - unreachable: exclude
  - reachable: trace back further

### Software Model Checking

- Adapt Hardware MC approach to Software MC
- Combine with well understood techniques from Software MC
  - ART construction
  - Interpolation

# 7: IC3 Modulo Theories via Implicit Predicate Abstraction

(Alessandro Cimatti, Alberto Griggio, Sergio Mover, and Stefano Tonnetta)

## IC3

- Hardware MC approach
- Finite set of variables
- Binary circuits
- Finite state space
- Excluding states always terminates

## Software Model Checking

- Control + data flow
- Finite set of variables
- Infinite variable domain
- Infinite state space
- Excluding states will not terminate

- Use Predicate Abstraction to project infinite state space to finite one
- Use original Boolean IC3 algorithm

# Outline

# 8: Assume-Guarantee Reasoning
## (Anvesh Komuravelli, Corina S. Păsăreanu, and Edmund M. Clarke)

- Model with multiple components: state space explosion

# 8: Assume-Guarantee Reasoning
## (Anvesh Komuravelli, Corina S. Păsăreanu, and Edmund M. Clarke)

- Model with multiple components: state space explosion
- Model checking: $C_1 \mid\mid C_2 \models P$?

# 8: Assume-Guarantee Reasoning
## (Anvesh Komuravelli, Corina S. Păsăreanu, and Edmund M. Clarke)

- Model with multiple components: state space explosion
- Model checking: $C_1 \mid\mid C_2 \models P$?
- Verification on individual components

# 8: Assume-Guarantee Reasoning
(Anvesh Komuravelli, Corina S. Păsăreanu, and Edmund M. Clarke)

- Model with multiple components: state space explosion
- Model checking: $C_1 \mid\mid C_2 \models P$?
- Verification on individual components

**Assume-Guarantee rule**

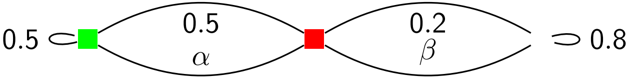
$$\frac{C_1 \models A \qquad\qquad C_2 \models A \Rightarrow P}{C_1 \mid\mid C_2 \models P}$$

- Model with multiple components: state space explosion
- Model checking: $C_1 \mid\mid C_2 \models P$?
- Verification on individual components

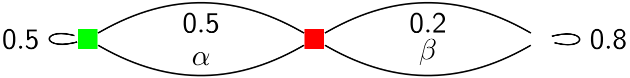
**Assume-Guarantee rule**

$$\frac{C_1 \models A \qquad\qquad C_2 \models A \Rightarrow P}{C_1 \mid\mid C_2 \models P}$$

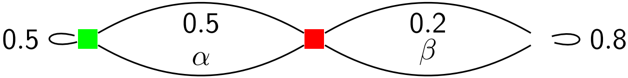
- Problem: find $A$



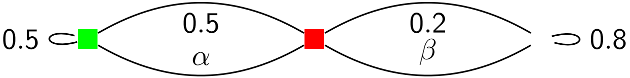
0.5
0.5
α
0.2
β
0.8



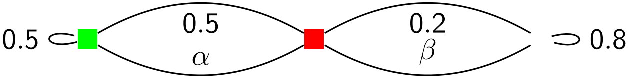

- Multiple-objective Long Run Average



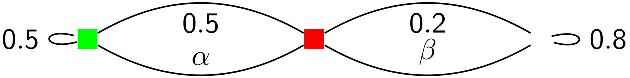

- Multiple-objective Long Run Average
- $\mathcal{M} \models \text{green} \geq 0.5$



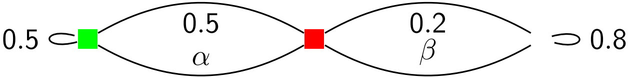

- Multiple-objective Long Run Average
- $\mathcal{M} \models$ red $\leq 0.2$



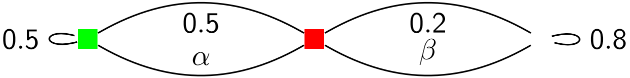

- Multiple-objective Long Run Average
- $\mathcal{M} \models \text{green} \geq 0.5 \wedge \text{red} \leq 0.2$




- Multiple-objective Long Run Average
- $\mathcal{M} \models$ green $\geq 0.5 \wedge$ red $\leq 0.2$
- Trade-off between objectives: Pareto curve



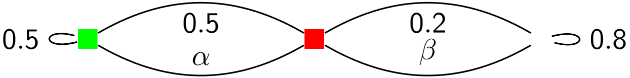

- Multiple-objective Long Run Average
- $\mathcal{M} \models$ green $\geq 0.5 \wedge$ red $\leq 0.2$
- Trade-off between objectives: Pareto curve




- Multiple-objective Long Run Average
- $\mathcal{M} \models \text{green} \geq 0.5 \wedge \text{red} \leq 0.2$
- Trade-off between objectives: Pareto curve

## Assume-Guarantee rule

$$\frac{C_1 \models A \qquad\qquad C_2 \models A \Rightarrow P}{C_1 \mid\mid C_2 \models P}$$




- Multiple-objective Long Run Average
- $\mathcal{M} \models \text{green} \geq 0.5 \wedge \text{red} \leq 0.2$
- Trade-off between objectives: Pareto curve

**Assume-Guarantee rule**

$$\frac{C_1 \models A \qquad\qquad C_2 \not\models A \wedge \neg P}{C_1 \parallel C_2 \models P}$$

# Outline

# Some Final Hints

## Hints

- Take your time to understand your literature.
- Be proactive! Look for additional literature and information.
- Discuss the content of your report with other students.
- Be proactive! Contact your supervisor on time.
- Prepare the meeting(s) with your supervisor.
- Forget the idea that you can prepare a talk in a day or two.

# Some Final Hints

## Hints

- Take your time to understand your literature.
- Be proactive! Look for additional literature and information.
- Discuss the content of your report with other students.
- Be proactive! Contact your supervisor on time.
- Prepare the meeting(s) with your supervisor.
- Forget the idea that you can prepare a talk in a day or two.

We wish you success and look forward to an enjoyable and high-quality seminar!