

Bachelor-Seminar: Static Analysis - Tools and Techniques

Master-Seminar: Static Analysis - Mastering Concurrency

Julian Erhard Michael Schwarz Sarah Tilscher
{julian.erhard, m.schwarz, sarah.tilscher}@tum.de

Chair for Formal Languages, Compiler Construction, Software Construction
Department of Informatics, Technical University of Munich

Winter Term 23/24

Writing programs is hard.

Writing correct programs is very hard.

Testing

- ▶ Widely successful
- ▶ Can be automated to some extent
- ▶ Can only show that there are bugs, not their absence

Machine-verified proof (e.g. Isabelle)

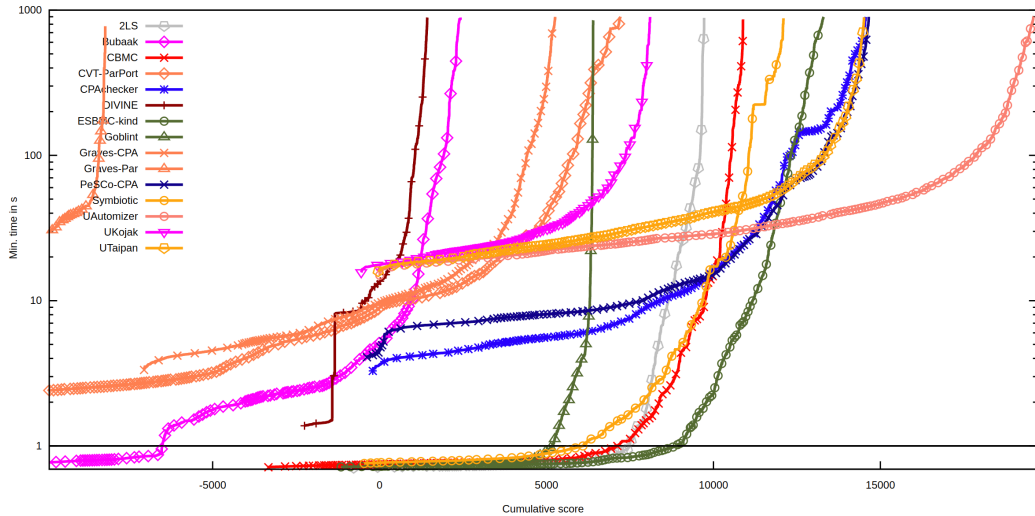
- ▶ Can show bugs & their absence
- ▶ A highly manual process requiring highly trained people
- ▶ Problem with proof and implementation diverging

Static Analysis

- ▶ Often fully automated
- ▶ Can show absence or presence of certain classes of bugs (depending on technique)
- ▶ Runs directly on the input program
- ▶ Abstract Interpretation, Model Checking, Symbolic Execution ...

Wide landscape of tools with different strengths & weaknesses.





Bachelor-Seminar: Static Analysis - Tools and Techniques

- ▶ Focus on concrete tools, diving deeply into the underlying approaches, and do a live demo
- ▶ Possible starting points: CPAchecker, Symbiotic, Frama-C WP, Ultimate Automizer, FB Infer, CBMC and its descendants, Divine, Predator

Master-Seminar: Static Analysis - Mastering Concurrency I

- ▶ Zoning in on one especially challenging aspect namely the analysis of concurrent programs
- ▶ investigate recent (and not so recent) approaches from the literature

Master-Seminar: Static Analysis - Mastering Concurrency II

- ▶ Possible topics include
 - ▶ Miné: Relational thread-modular static value analysis by abstract interpretation. VMCAI '14
 - ▶ Farzan et al.: Stratified Commutativity in Verification Algorithms for Concurrent Programs. POPL '23
 - ▶ He et al.: Satisfiability modulo ordering consistency theory for multi-threaded program verification. PLDI '21.
 - ▶ Sharma and Sharma: Thread-modular Analysis of Release-Acquire Concurrency. SAS '21
 - ▶ Jeannet: Relational interprocedural verification of concurrent programs. Software & Systems Modeling '13
 - ▶ Gotsman et al.: Thread-modular shape analysis. PLDI '07
 - ▶ S. et al.: Clustered Relational Thread-Modular Abstract Interpretation with Local Traces. ESOP '23

Organisation

- ▶ mini-conference
- ▶ reviewing of draft papers among students
- ▶ talks en-block (mid of February)

Task

- ▶ write a Latex paper (8-10 pages)
- ▶ review 2 drafts of your fellow students
- ▶ 20-25 minutes presentation + 10 minutes discussion for each topic
- ▶ the language (for talks and paper) is English

Schedule

Pre-course meeting	today
Kick-off meeting	First week of lectures
Topic distribution	Second week of lectures
Individual meeting	
Draft paper submission	TBA
Submit reviews	TBA
Final paper submission	TBA
Talks	Mid of February

Questions?

