Bachelor & Master Projects and Theses
Chair for Software and Systems Engineering

Prof. Dr. Stefan Leue

University of Konstanz
Chair for Software Engineering

Stefan.Leue@uni-konstanz.de
http://www.sen.inf.uni-konstanz.de

Summer Term 2018
Problem Description

- ATL allows for model-to-model transformation
  - e.g., SysML to PRISM, UPPAAL, Promela/SPIN
- first transformation system available
  - MSc Thesis Hargurbir Singh
- can we prove that the ATL rules preserve the operational semantics?

Project/Thesis

- literature survey on correctness of model-to-model transformations
- development of proof approach for (parts of) above translation
Problem Statement

- Causality Checking in QuantUM relies on trace computation
  - bottleneck: memory
- LTSmin / PINS is interface to symbolic model checking engine
  - hope: more memory efficient

Project Tasks

- integrate QuantUM/Causality Checking into LTSmin / PINs
- direct encoding of causality checking in symbolic BDD data structures [MP/MT]

Prerequisites

- programming, discrete structures
- for [MP/MT]: (symbolic) model checking an advantage
Checking Real-Time Properties for SysML [BP/BT]

- (N.N.)

- **SysML and Real Time**
  - overview of real-time property expressiveness in SysML / MARTE
  - analysis of real-time primitives in SysML / Papyrus
  - definition of a semantics for selected primitives

- **Analysis**
  - translation into Timed Automata / UPPAAL input language using ATL model transformation technology
  - reverse interpretation of the analysis results
Pattern Mining for Fault Localization [BP/BT/MP/MT]

- **Counterexamples in Model Checking**
  - there can be thousands of counterexamples with hundreds of events

- **Fault Localization**
  - use sequential pattern mining to find patterns in counterexamples that
    - occur frequently in "good" counterex.
    - occur infrequently in "bad" counterex.

- **Goals**
  - some exisitant seq. pattern mining algorithms
  - implementation and link to model checking
  - case studies

- **Collaboration**
  - with PD Dr. Christian Borgelt, Bio-Informatics and Information Mining
Causality Checking for Real Time [MP/MT]

- **Real-Time Systems**
  - correctness depends on relative timing of computation steps
  - timed automata
  - UPPAAL model checker

- **Research Question**
  - how can causality checking be extended to real-time properties?
  - what is a meaningful causality concept in this setting?

- **Project/Thesis Tasks**
  - how do counterexamples in real-time model checking look like?
  - development of a causality concept for real-time properties
  - implementation of this concept, case studies

- **Prerequisites**
  - Advanced Model Checking
Causality Checking for Programs [MP/MT]

Programs
- the assignment of certain values to variables can cause a program to crash
- which variable assignments and which values are causal for a program failure?

Tasks
- selection of a program analysis framework (other than testing)
  - for instance, symbolic execution, static analysis
- development of a causality notion for program executions
- prototype implementation and case study

Prerequisites
- good understanding of logic, program semantics, foundations of computing
Causality Checking for Deadlock Properties [BP/BT] [MP/MT]

- **Deadlocks**
  - circular wait, no more progress
- **Causality Checking for Deadlocks**
  - deadlock is reachability [BP/BT]
    - adopt causality checking to deadlock
    - implement in SpinJa
  - consider different deadlocks
    - extend the algorithmics and implementation of causality checking to multiple deadlocked states
  - implementation and case studies
- **Prerequisites**
  - preferably, one of the model checking courses
  - good programming skills
Run-Time Causality Checking [MP/MT]

- **Runtime Verification**
  - observe and assess running system
  - often: monitoring

- **Run-Time Causality Checking**
  - observe system behavior
  - detect occurrence of events at run-time as causal for undesired system behavior
  - learn for the future

- **Tasks**
  - study various run-time verification approaches, in particular run-time model checking
  - analyze, what causality can mean in this context
  - adapt causality checking

- **Prerequisites**
  - one model checking course
  - advantageous: machine learning, data mining
Model Transformation

- UML-RT models edited in Papyrus Real Time (EMF)
- understand semantics of state machine diagrams, inter-object communication and meta-models
- define transformation rules in ATL model transformation framework
- implementation and case studies

Benefits

- exposure to practically very relevant model based design language
- interest in Papyrus community

Prerequisites

- Software Engineering
- interest in semantics and model transformation
Model Transformation to LTSmin / PINS
- SysML models edited in Papyrus Real Time (EMF)
- understand semantics of state machine diagrams, inter-object communication and meta-models
- define transformation rules in ATL model transformation framework
- implementation and case studies

Benefits
- exposure to practically very relevant model based design language
- interest in Papyrus community

Prerequisites
- Software Engineering
- interest in semantics and model transformation
CHESS

- tool for finding and producing bugs in concurrent programs (C,C++,C#)

Task

- transfer the idea of Causality Checking to testing / code analysis
- apply it to the CHESS tool / environment
- implement prototype
- perform case studies
Develop a graphical representation of EOL (event order logic)

- EOL (event order logic) is a subset of LTL (linear temporal logic) including ordering constraints.
- currently, EOL is represented by formulae.

\[(T_a \land (C_a \land C_c)) \land \langle \neg C_l \land \rangle T_c\]

- graphical representations are better suited for non-specialists, e.g.:

Prerequisite:

- knowledge of LTL, e.g., course: Model Checking of Software and Systems, is an advantage but not strictly required
Formal Requirements, Deliverables

♦ Before you start your work
  ‣ Written proposal (≈ 1 page) containing (deadline: May 2, 2018)
    ‐ the topic you want to choose
    ‐ how well you fit the prerequisites
    ‐ why you are suited for the task.
  ‣ Schedule for the project/thesis
    ‐ What should be achieved at which point in time.

♦ During your preparation of the thesis or the project work
  ‣ Regular consultation with your supervisor.

♦ Deliverables
  ‣ Any models/code/data/binaries you created for the project.
  ‣ Report of ≥ 20 pages (Projects, Bachelor-Thesis) or ≥ 40 pages (Master-Thesis)
    ‐ Discussing state of the art
    ‐ Stating the problem
    ‐ Presenting your approach and results
    ‐ Critical discussion
Interested? Contact...

... either one of us any time!

- Prof. Dr. Stefan Leue
  - Email: Stefan.Leue@uni.kn, Room: PZ 902
- Dr. Georgiana Caltais
  - Email: Georgiana.Caltais@uni.kn, Room: PZ 913
- Martin Kölbl
  - Email Martin.Koelbl@uni.kn, Room: PZ 912