Title: Bounded model checking of liveness properties of TLA+ specifications

Topic: Computer science – formal verification
City: Nancy, France
Team: Veridis, Inria Nancy – Grand Est

Advisors: Igor Konnov (igor.konnov@inria.fr), Stephan Merz (stephan.merz@inria.fr)

Context of the topic

TLA+ [1] is a language that was designed for the formal specification of systems such as concurrent and distributed algorithms. It is currently supported by the explicit-state model checker TLC and the interactive TLA+ Proof System TLAPS. In the ongoing project APALACHE [2,3], we are developing a symbolic model checker for TLA+ as a complement to TLC, which relies on state enumeration. In contrast, the bounded model checker generates constraints describing counter-examples to claimed properties that are then discharged by an SMT solver.

Objective of the internship

The overall procedure of symbolic model checking works as follows: (1) it identifies symbolic transitions in a TLA+ specification [4], and (2) it encodes bounded executions and safety properties as constraints in the input language of the SMT solver. The current version verifies whether a bounded execution reaches a state that violates a system invariant provided by the user. These properties belong to the class of safety properties: The system is doing nothing "bad".

It is well understood that safety property alone does not guarantee that a system is doing anything useful. In order to prove that the system is going to achieve a "good" state, one has to prove liveness of the system. For instance, one might want to prove that the system eventually reaches a state in which all the system components have finished their computations. In this work, we propose to extend the APALACHE model checker with techniques for bounded model checking of temporal properties. The intern is expected to implement the well-known technique by Biere et. al. [5] in the model checker for TLA+, which currently supports only finite-state systems. If successful, an extension with the technique recently suggested by Padon et. al. [6] can be envisaged, which, in principle, applies to infinite-state systems.

References

[1] Leslie Lamport. Specifying Systems. Addison Wesley, 2002.
URL: http://lamport.azurewebsites.net/tla/tla.html.

[2] Igor Konnov, Jure Kukovec, Thanh-Hai Tran. BMCMT: Bounded Model Checking of TLA+ Specifications with SMT. TLA+ Community Meeting, FLoC 2018. URL: http://tla2018.loria.fr/contrib/konnov.pdf

[3] APALACHE model checker. URL: http://forsyte.at/research/apalache/

[4] Jure Kukovec, Thanh-Hai Tran, Igor Konnov. Extracting Symbolic Transitions from TLA+ Specifications. Abstract State Machines, Alloy, B, TLA, VDM, and Z, pages 89-104, 2018.

[5] Armin Biere, Keijo Heljanko, Tommi A. Junttila, Timo Latvala, Viktor Schuppan. Linear Encodings of Bounded LTL Model Checking. Logical Methods in Computer Science 2(5), 2006.

[6] Oded Padon, Jochen Hoenicke, Kenneth L. McMillan, Andreas Podelski, Mooly Sagiv, and Sharon Shoham. Temporal Prophecy for Proving Temporal Properties of Infinite-State Systems. FMCAD, 2018.
URL: https://www.cs.tau.ac.il/~odedp/temporal-prophecy.pdf

Prerequisites

The student should have some familiarity with formal modeling and verification techniques. Knowledge of TLA+ is not expected. Some basic knowledge of SMT or constraint solving would be a plus. The APALACHE model checker is implemented in Scala, and elementary knowledge of Scala and Java is required.