

# IN9100: Introduction to Runtime Verification

Paul Kobialka, Peter Ölveczky

November 5, 2023

## Curriculum

The curriculum consists of two papers:

- [1] Bartocci, E. and Falcone, Y. and Francalanza, A. and Reger, G.: Introduction to runtime verification. In: Lectures on Runtime Verification - Introductory and Advanced Topics. LNCS 10457, pp. 1–33. Springer (2018).
- [2] Roşu, G. and Havelund, K.: Rewriting-based techniques for runtime verification. Automated Software Engineering **12**(2): pp. 151–197 (2005).

The two papers complement each other, [1] gives a general introduction to runtime verification, and [2] presents actual algorithms, i.e., with formalizations in Maude.

## 1 What to know?

For [1]:

- Why are different groups of specification languages presented? What are their benefits? (Generally, with few examples from Section 2.3)
  - Compare with presented features of specification languages (Section 2.2).
- What is the typical monitoring setup?
  - Describe the three elements: Monitor, System, Instrumentation.
- Which design choices are made when setting up a monitor?
  - Focus and elaborate on the setting presented in [2].
- How are monitors employed at the software level?
- Hardware Instrumentation is not part of the curriculum (Section 4.2)
- What properties are monitorable?

- Section 5.2 is not part of the curriculum

For [2]:

- How does the setting considered in [2] relate to settings summarized in [1]? (Section 2)
- Which foundations are implemented in Maude for monitoring? (Sect 3.2)
- What is finite-trace LTL and how is it different from infinite-trace LTL?
- What is the Maude specification for naive runtime verification?
- Section 5 is not part of the curriculum.
- Which improvements are introduced in Section 6?
  - Elaborate also from a Maude perspective.
- Why is the algorithm from Section 6 called asynchronous?
- Proofs in Section 6 are not part of the curriculum.
- Section 7 is not part of the curriculum.
- Why do the authors propose rewriting for monitoring?