Objective:

The goal is to provide the basics of formal methods applied in contemporary software and system engineering practice. Classical schematics based methods alone do not guarantee the quality of software products regarding safety, security and functional correctness requirements. The aims at providing theoretical knowledge and practical skills needed for applying formal techniques and tools in system development automation.

Description:

As a prerequisite the course presumes basic knowledge in algorithmic, logics, data structures and programming. The main body of the course covers:

a. Software models and their semantics: state vs event-based, deterministic vs non-deterministic, synchronous vs asynchronous, safety, liveness, real time and concurrency.
b. Correctness preserving model transformation- and constraint solving based software synthesis
c. Verification algorithms in model checking and in deductive verification.
d. Verification techniques: state space reduction, aspect- oriented, and component models, compositional and hierarchical verification, abstractions.

Learning outcome:

The course provides basic knowledge and skills for

a. composing and verifying software requirements specifications
b. performing and validating the correctness of development steps by using model checking and deductive verification tools
c. using model based development techniques and tools for automated test generation and execution.

Readings: