Model checking is a technique for verifying finite-state concurrent systems such as sequential circuit designs and communication protocols. It has a number of advantages over traditional approaches that are based on simulation, testing and deductive reasoning. In particular, model checking is automatic and usually quite fast. Also, if the design contains an eror, model checking will produce a counterexample that can be used to pinpoint the source of the error. The method has been used successfully in practice to verify real industrial designs, and several companies are beginning to market commercial model checkers. Among its many academic accomplishments we should mention an ACM Doctoral Dissertation Award for Ken McMillan in 1992, the ACM Paris Kanellakis Award for Theory and Practice in 1999 and the 2007 Turing Award for Edmund Clarke, Alen Emerson and Joseph Sifakis for their work on verification based on temporal logic, the basis for model checking.

The course will discuss verification in general, the basics of model checking and several advanced topics, required for practical uses of the method.

Bibliography:

Clarke, et. al. Model Checking, second edition (Cyber Physical Systems Series) 2nd ed. The MIT Press, 2018.
and class notes

Program:

- Class 1: General info and a bit of history.
- Class 2: Reactive Systemas and Temporal Properties.
- Class 3: Model Checking Overview.
- Class 4: Temporal Logic Model Checking.
- Class 5: Binary Decision Diagrams.
- Class 4: Basic fixpoint theorems.
- Class 5: Symbolic model checking.
- Class 6: SMV.
- Class 7: Compiling source code to graphs.
- Class 8: Verification of Real Time Systems.
- Class 9: Verus
- Class 10: LTL model checking.
- Class 11: Compositional reasoning.
- Class 12: SAT Solvers.
- Class 13: Bounded model checking.
- Class 14: Symbolic trajectory evaluation.
- Class 15: Scheduling problems using SMC.
- Class 16: Real-time systems.