Computer-Aided Verification and Synthesis

Course Description:
This course will introduce the fundamental theory in computer-aided verification and synthesis for building provably dependable computing systems. The topics covered include logic specifications, modeling formalisms, verification techniques, and inductive synthesis strategies. A special focus of this course is on the interplay between deductive reasoning (logical inference and constraint solving) and inductive inference (learning from data). We will also survey applications of these techniques to a wide range of problems in hardware, software, cyber-physical systems, robotics and biology.

Prerequisites:
Familiarities of propositional logic, basic probability theory and basic graph algorithms, and experience with one programming language (e.g., C++, Python) are assumed. An undergraduate course in algorithms (e.g., EC330 or equivalent) is required. If you are unsure whether you have sufficient background for the course, please contact the instructor.

Instructor: Wenchao Li <wenchao@bu.edu>

Course Schedule: Two lectures per week. Each lecture will be 1 hour and 45 minutes.

Grading (tentative): Homework (20%), Midterm (20%), Participation (10%), Project (50%).

Course Topics (tentative):
Specification/Logic: (2.5 weeks)
- Review of propositional and first-order logic
- Temporal logic (LTL, CTL, PCTL, STL, etc.)
- Hyperproperties
- Specification mining and inference

Model/Automata: (2.5 weeks)
- Finite-state automata
- Timed automata
- Probabilistic automata and Markov chains
- Automata learning

Constraint solving with SAT and SMT: (3 weeks)
- SAT solving
- Satisfiability Modulo Theories (SMT)
- ∃∀ problems

Model Checking: (4 weeks)
- Explicit-state model checking
- Symbolic model checking
- Abstraction; Refinement; Interpolation
- Probabilistic model checking

Synthesis: (3 weeks)
- Syntax-guided synthesis and learning strategies

- Game-theoretic formalisms and techniques

Applications: (spread out across the course)

- Cyber-physical systems
- Robotics and control
- Computer security
- Software engineering
- Networked and distributed systems
- CAD for integrated circuits
- Synthetic biology
- A.I. safety

Project: The project will be a theoretical and/or experimental investigation related to topics covered in class. A list of suggested project topics will be announced towards the middle of the semester. You are also encouraged to pick topics of your own, but you will need to consult with the instructor. The projects should be done in groups of 2 or 3 students. Projects will culminate in a final paper, presentation, and possibly a demo.

Reference Books:

There is no required textbook for this course. The following books are recommended references. Copies of relevant chapters will be provided.

1. Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled, "Model Checking," MIT Press, January 2000.
2. Christel Baier and Joost-Pieter Katoen, "Principles of Model Checking," MIT Press, April 2008.
3. Michael Huth and Mark Ryan, "Logic in Computer Science: Modelling and Reasoning about Systems," Cambridge University Press, June 2004.
4. Edward A. Lee and Sanjit A. Seshia, "Introduction to Embedded Systems: A Cyber-Physical Systems Approach," MIT Press, 2017. http://leeseshia.org/
5. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach," MIT Press, December 2009.