CS 246 Software Verification

Instructor: Mohsen Lesani
Email: lesani@cs.ucr.edu
url: https://www.cs.ucr.edu/~lesani/
Prerequisites: CS/Math 111, CS 141, CS 150, or equivalents; or consent of instructor
Units: 4

Catalog description:
A study of techniques to specify and verify the correctness of software systems. Topics include formal description languages, first order theories, type systems, mechanized verification and model checking.

Summary:
Recent studies show that the global cost of software and hardware bugs is hundreds of billion dollars annually. Aircraft, space shuttles, self-driving cars, medical devices and Internet services are a few use cases that require reliable systems. In the last decade, the verification community has brought tools to assist engineers in building reliable computing systems. We are at the beginning of the new era of verified (rather than tested) computing products. The goal of this course is to introduce the students to the theory and practice of program specification and verification.

In this course, we will study propositional logic, first-order logic, and commonly used first-order theories and their satisfiability decision procedures. We will work with the state-of-the-art constraint solving tools. We will also study programming using dependent types and interactive theorem proving using higher order logics. We will work with the state-of-the-art functional programming languages and theorem provers. We will study Hoare logic to prove the partial and total correctness of sequential programs. We also study abstract syntax, operational semantics and basic type systems and their proof of soundness. Finally, we will study the fundamentals of model checking.

Contents
Week 1:
Propositional Logic, First Order Logic
Reading: Text book 1, Chapter 1 and 2

Week 2:
First Order Theories
Reading: Text book 1, Chapter 3

Week 3:
Induction
Reading: Text book 1, Chapter 4

Week 4:
Program Verification by Hoare Logic
Reading: Text book 1, Chapter 5, 6

Week 5:
Typed lambda calculus, Type systems
Reading: Text book 2, Chapter 9
Week 6:  
Type Inference  
Reading: Text book 2, Chapter 22

Week 7:  
Subtyping  
Reading: Text book 2, Chapter 15

Week 8:  
Mehahnized Theorem Proving with Coq, Logics  
Reading: Text book 3, Volume 1. Chapter 1-5

Week 9:  
Mehahnized Theorem Proving with Coq, Program Verification  
Reading: Text book 3, Volume 2. Chapter 1-3

Week 10:  
Model Checking  

Textbooks

Grading
Homeworks (two): 30% of the grade.  
Midterm: 30% of the grade.  
Project and report: 40% of the grade.  
Projects can be performed individually or in groups of two (rarely three) students. A project proposal is required at least one month before the end of classes. Various types of projects can be proposed, for example: (1) a case study of a substantial system using one of the tools studied in class, (2) a comparison of the tools studied in class on a realistic set of benchmarks, (3) implementing a new algorithm, or (4) developing the theory for a new analysis algorithm. A mixture of the above, e.g., (1) and (2) or (3) and (4) might also be quite appropriate, especially in a group project.