Course Outline

1. Title: Formal Specification and Verification
2. Credit Structure (L-T-P-Cr): 3--- 0 ---0 ---3
3. Course Code: IT 427
4. Program/Semester: BTech, Semester VI
5. Category: Elective
6. Instructor: Rahul Muthu
7. Prerequisite: Discrete Mathematics (SC106), Models of Computation (IT422)
8. Foundation for: Advanced Model Checking
9. Abstract:

Improving the reliability of software systems is widely recognized as one of the major challenges in programming language research and technology today. Early detection of errors in requirements, designs and implementations requires a variety of program analysis and verification techniques. These techniques range from light-weight simulation to medium-weight static analysis or model checking to heavy-weight theorem proving.

In this course we will concentrate on two major formal approaches for verifying the correctness of programs: model checking and deductive verification. We will study the fundamentals of each approach, with a focus on recent advances. We will also study some concrete tools following these approaches, and evaluate them in term of their expressive power, domain of applicability, degree of automation, efficiency, soundness, and completeness.

10. Books and References:

1. Software Reliability Methods, D. A. Peled
2. Systems and Software Verification,
3. Logic in Computer Science, Huth, Ryan

11. Detailed Contents:

1. Proposition Logic, First Order Logic, Linear-time Temporal Logic, Branching Time Temporal Logic (Computational Tree Logic).

2. Modelling the behaviour of systems: Finite state automata , Non-determinism, Extensions to Buchi Automata and other automata on infinite words. Regular languages and ω-regular languages.

3. Expressing properties of the behaviour of systems: Introduction to formal specification, Assertions, Invariants , Safety and liveness properties, Regular expressions, Buchi-automata, and never-claims, Linear Temporal Logic (LTL), Computation Tree Logic (CTL), Specification patterns
4. Model Checking (Or Verification)

Checking safety properties, Checking liveness properties using Buchi automata, Checking LTL properties, CTL Model Checking

5. Relation of Verification with Testing

This module will cover the main difference between these two fundamental ways of improving software reliability. The former is a more rigorous mathematical method, while the latter is more of an empirical method based on typical representative test cases.

11. Outcomes and Objectives:

Students, after taking this course, are expected to know the benefits of formal verification in the software development process, and to understand a variety of formal verification methods and their applicability. Further, students are expected to learn several well developed formal verification methods for both sequential and concurrent software systems and be able to apply them to verify small benchmark systems. In particular, this course will cover the new verification paradigm of modeling checking, which has become the focal research area in the past decade and has become very successful in industrial applications. Several well-established model checking techniques will be studied. A specific model checking tool (perhaps SPIN) will be covered including use in practice by the students.

12. Attendance Policy: Nothing specific, except that there will be some continuous evaluation modules, which are contingent on the students’ general presence in a sizable fraction of the lectures.

13. Evaluation: Partial in-course (class tests, assignments, quizzes, presentations, projects) and partial examination based.