COURSE DESCRIPTION

Course Name:	Verification of Reactive Systems
Course Code:	**CS405**
Credits:	**3-0-0-3**
Prerequisites:	MFCS, FLAT (recommended)
Intended for:	BTech, MTech, MSc, MS/PhD in the area.
Elective/Core:	Discipline elective for BTech CSE and EE, free elective for others
Semester:	even

Preamble:

In the process of software/ system development, usually, more than 50% of time is spent on testing and verification to ensure the correct functioning. While testing essentially means running the software/system with particular data or under specific conditions, verification amounts to verifying properties without such restrictions. A correctness proof, obviously, is the most desirable to achieve. It is particularly important in the case of safety critical or mass critical systems.

This course introduces to techniques used to verify the correctness of reactive, non-terminating systems. Apart from giving the necessary theoretical background, the course familiarizes students with software supporting the verification.

Course Outline:

In the course we explore how reactive, non-terminating systems can be verified, in particular,

- how reactive systems can be abstractly modelled,
- what typical properties of reactive systems look like,
- how formal logics help to express such properties,
- techniques to verify whether a property holds for (the model of) a system,
- tools to do the model checking automatically,
- equivalence notions for reactive systems,
- correctness by design,
- case studies of verifications.

As part of the course work, students will work with widely used model checker tools UPPAAL, SPIN and the proof assistance tool RODIN.

Modules:

Introduction (1 week)

- Introduction and overview of the course
- Revision of predicate logic, undecidable problems, proof systems.
- Basic approaches to verification:

property and equivalence verification, correctness by design

Modelling of Systems (2 weeks)

- Formal models for non-terminating reactive systems (hardware/software) : Buechi Automata, Transition Systems, Process Algebras, Petri Nets
- Dimensions of behavior descriptions :
  - interleaving vs "truly concurrent"
  - linear vs branching time
  - quantitative vs qualitative descriptions
- Property classification: regularity, safety, liveness, fairness
- Equivalence notions:
  - trace equivalence
  - testing equivalence
  - observation equivalence
- Case studies - equivalence checking

Property Verification of Reactive Systems (4 weeks)

- Verification of regular properties
- Temporal and modal logics for property specification: LTL, CTL, CTL*, HML
- Verifying properties: principles of model checking, complexity and limits
- Case studies – with model checker SPIN

Property Verification of Time-Critical and Hybrid Systems (3 weeks)

- Timed automata (theory, reachability analysis)
- Case study – with model checker UPPAAL

Correctness by Design (2 week)

- Stepwise refinement
- Case study – with proof assistant RODIN

(Remaining week will be used as buffer and for doubt clearing.)

Evaluation

- Assignments/Tests: 20 %
- Project: 20 %
- Quiz 1: 15 %
- Quiz 2: 15 %
- Final exam: 30 %

Textbooks:

- Christel Baier and Joost-Pieter Katoen: "Principles of Model Checking", MIT Press, 2008.

References:

- Luca Aceto, Anna Ingolfsdottir, Kim G. Larsen and Jiri Srba: Reactive Systems – Modelling, Specification and Verification, Cambridge textbooks, 2007
- M. Ben-Ari: Principles of the SPIN Model Checker, MIT Press, 2008
- Wan Fokking: Modelling Distributed Systems, Springer Verlag, 2007
- Michael Huth and Mark Ryan: Logic in Computer Science – Modelling and Reasoning about Systems, Cambridge University Press, 2004
- Gerald Holzmann: The SPIN Model Checker - Primer and Reference Manual, Addison Wesley, 2003
- Doron Peled: Software Reliability Methods, Springer Verlag, 2001