CO7209 Software Reliability

Credits: *15* **Convenor:** *Dr. A. Kurz* **Semester:** *1st*

Prerequisites:	*none*		
Assessment:	*Coursework: 40%*	*Two hour exam in January: 60%*	
Lectures:	*23 hours*	**Private Study:**	*73.5 hours*
Laboratories:	*16 hours*		

Subject Knowledge

Aims The aims of this module are to (1) present a collection of methods for dealing with software reliability; (2) explore in detail some of the methods and tools that have been developed in recent years for automatic verification of properties of software systems; (3) raise awareness to the limitations of current technologies, and how they may be overcome in the future.

Learning Outcomes This module introduces techniques and tools for verifying that computer systems are reliable in the sense that they have the properties intended. The module covers: languages for modelling systems and their properties; model checking and algorithms; a selection of tools (e.g. SPIN); specification, verification and validation of typical properties of reactive systems; limitations of automatic verification techniques; Relationship to software testing techniques (black-box checking).

Methods Class sessions, tutorials and laboratory sessions together with course notes, recommended reading, worksheets, printed solutions, and some additional hand-outs.

Assessment Assessed coursework, traditional written examination.

Skills

Aims To teach students the role and nature of software reliability methods; to develop in the students the ability to separate concerns during system verification, namely in what concerns modelling software systems, specifying required properties, and applying model-checking based verification techniques.

Learning Outcomes On completion of this module, the student should be able to: use tools (e.g. SPIN) to verify and debug small-scale systems; understand and explain the principles and algorithms behind those tools; understand the application of the tools to different domains; understand the limitations of current verification techniques (e.g. the state explosion problem) and efforts to overcome them.

Methods Class sessions together with worksheets.

Explanation of Prerequisites

Knowledge of Discrete Maths will be helpful.

Course Description

Software reliability methods are now past the stage of ”promising curiosities” and offer techniques and tools that can be employed in a variety of application domains to verify required functional properties against models of system behaviour.

This module covers some of these techniques and tools, namely those which, like SPIN, are based on model-checking. It introduces students to languages for modelling systems and their properties, some of the algorithms that can be employed for automatic verification, and the limitations of current implementations.

Detailed Syllabus

- Modelling Software systems with state transition systems.
- Introduction to propositional and temporal logic.
- Specifying software system properties with temporal logic.

- Automatic verification techniques based on model-checking.
- Relationship of model-checking and black-box testing.

Reading List

[B] Peled, Doron, *Software Reliability Methods,* Springer-Verlag, 2001.

[C] Holzmann, Gerard, *Spin Model Checker: Primer and Reference Manual,* Addison Wesley, 2003.

[C] McMillan, Ken, *Symbolic model checking,* Kluwer Academic Publishers, 1993.

Resources Study guide, worksheets, lecture rooms with data projector, computer laboratory access, tutorial rooms with OHP.

Module Evaluation Course questionnaires, course review.