Course Outline

Course Details

Course Code	COMP3153/9153
Course Title	Algorithmic Verification
Units of Credit	6
Course Website	https://www.cse.unsw.edu.au/~cs3153

Staff

Name	Role	Contact
Dr. Liam O'Connor	Lecturer in Charge	liamoc@cse.unsw.edu.au
Dr. Miki Tanaka	Lecturer	
Prof. Rob van Glabbeek	Lecturer	

Course Summary

It is virtually impossible to guarantee correctness of a system, and in turn the absence of bugs by standard software engineering practice such as code review, systematic testing and good software design alone. The complexity of systems is typically too high to be manageable by informal reasoning or reasoning that is not tool supported. The formal methods community has developed various rigorous, mathematically sound techniques and tools that allow the automatic analysis of systems and software. The application of these fully automatic techniques is typically called algorithmic verification. The course will describe several automatic verification techniques, the algorithms they are based on, and the tools that support them. We will discuss

examples to which the techniques have been applied, and provide experience with the use of several state-of-the-art analysis tools.

The topics covered by the lectures will educate students on the foundations of automata theory and temporal logics, LTL and CTL model checking techniques and model checking tools, the application of static analysis techniques to program verification, and modern advanced verification techniques for timed and probabilistic systems.

Course Timetable

Day	Time	Location
Tuesday	10:00-12:00	Mathews 309
Friday	10:00-12:00	UNSW Business School 119

In week 8 there is a public holiday on the friday. We have a make-up lecture in Week 11 in the normal Tuesday time but located in UNSW Business School 119.

Outcomes

After completing this course, you should:

- Be able to distinguish between the various automatic formal methods available, weighing up their advantages and disadvantages.
- Be able to develop formal models of software systems, amenable to automatic verification.
- Be able to specify software using program logics such as temporal logic.
- Develop a scientific understanding of algorithmic verification techniques, including the underlying concepts for state-of-the-art tools.
- Understand the role of abstraction in formal methods and its uses to simplify models and make verification feasible.
- Be able to use various algorithmic verification tools.

Assumed Knowledge

You need to have successfully completed the core programming, algorithm, and discrete mathematics courses.

The course makes use of a number of discrete mathematics concepts. You may find the course very difficult without MATH1081 or equivalent discrete mathematics background.

Teaching Strategies

The learning focus in this course is primarily on lectures and homework assignments. While marks are assigned to the homework, their primary purpose is to give you concrete tasks with deadlines to help you structure your learning. `

Lectures

The lectures will introduce you to new material, which is being reinforced and practised in the assignments. The course draws on several textbooks for material, listed later in this document. More reading material covering specific topics will be identified throughout the course. Students are required to study reading material as advised during the lecture and/or on the course web page.

There are three to four hours of lectures each week. Generally, lectures will be highly interactive. Recordings have traditionally not been available for this course, and while I may choose to make them available this time around, the course relies heavily on board work, and this may make the recordings difficult to use as a learning resource.

Attendance at all lectures is expected for this course.

Homework Assignments

There are five homeworks released throughout the course. They will be a series of questions that keep you up to date with course content and structure your learning. Feedback will be provided by the lecturer to help you identify problems in your learning.

It's expected that you will do this homework alone, to maximise its value to you as a learning resource. Don't cheat yourself by plagiarising. More information on plagiarism is provided below.

Assessment

Assessment is the aggregation of your homework and final exam, weighted 60/40 in favour of the final exam.

Final Examination

The final exam is a two hour written exam. Requests for a supplementary exam will only be considered where students (a) have completed all other course components to a satisfactory standard, (b) have been absent from the final exam, (c) and have submitted a fully documented request for special consideration to student lifecycle within three working days of the final exam.

Student Conduct

The **Student Code of Conduct** (Information, Policy) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students **inform themselves** of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer, Grievance Officer, or one of the student representatives.

Plagiarism is defined as using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity
- UNSW Plagiarism Procedure

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism

The pages below describe the policies and procedures in more detail:

- Student Code Policy
- Student Misconduct Procedure
- Plagiarism Policy Statement
- Plagiarism Procedure

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students

Course Schedule

Available on its own dedicated page (see the sidebar).

Resources for Students

Recommended Textbooks

- Christel Baier and Joost-Pieter Katoen. **Principles of model checking**. MIT Press, 2008. ISBN 978-0-262-02649-9
- Edmund Clarke, Orna Grumberg and Doron Peled. **Model Checking**. MIT Press, 2000.

- Michael Huth and Mark Ryan. **Logic in Computer Science (2nd edition)**. Cambridge University Press, 2004.
- Beatrice Berard et al. **Systems and Software Verification: Model-Checking Techniques and Tools**. Springer, 2001

Further Reading

- Rajeev Alur. **Techniques for Automatic Verification of Real-Time Systems**. PhD thesis, Stanford University, 1991
- Thomas Ball, Vladimir Levin, Sriram K. Rajamani. **A decade of software model checking with SLAM**. Communication of the ACM 54(7):68-76, 2011.
- Armin Biere, Marijn Heule, Hans van Maaren, Toby Walsh. **Handbook of Satisfiability**. Frontiers in Artificial Intelligence and Applications 185, IOS Press, 2009.
- Edmund Clarke, Daniel Kroening, Natasha Sharygina. **Predicate Abstraction of ANSI-C Programs Using SAT Formal Methods in System Design**, 25, 105—127, Kluwer Academic Publishers, 2004.
- Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. **Counterexample-guided abstraction refinement**. In Computer Aided Verification, pages 154Đ169, 2000.
- Dexter Kozen. **Automata and Computability**. Springer, 1997. ISBN 978-0-387-94907-9
- Flemming Nielson, Hanne Riis Nielson and Chris Hankin. **Principles of Program Analysis**. Springer, 1999.
- Michael Sipser. **Introduction to the Theory of Computation (3rd edition)**. Cengage Learning, 2013. ISBN-13: 978-1-133-18781-3

`Verification tools

- The nuXmv model checker
- The SPIN model checker
- The CBMC model checker
- The UPPAAL real-time model checker

Getting Help

Questions regarding the course material, assignments, exercises, and general administrative questions should be asked on the course forum (accessible from the course web page), where answers benefit the whole class. Alternatively, approach the lecturer after class. To discuss matters concerning your personal performance, please

send an email to the course account `cs3153@cse.unsw.edu.au`. For identification purposes, if you wish to send email concerning the course, you must send the mail from your CSE or UNSW student account (not from GMail, Yahoo, Bigpond or similar), and include your student id and your full name.

Course Evaluation and Development

This course is being continuously improved and we will conduct a survey at the end of session to obtain feedback on the quality of the various course components. Your participation in the survey will be greatly appreciated.

Student feedback over the last years has generally been positive.