Griffith UNIVERSITY

COURSE OUTLINE

3515ICT
Theory of Computation

Semester 2 - 2007

Academic Organisation:	School of Information and Communication Technology
Faculty:	Science, Environment, Engineering and Technology
Credit point value:	10
Student Contribution Band:	Band 2
Course level:	Undergraduate
Campus/Location/Learning Mode:	Nathan / On Campus / In Person
Convenor/s:	Prof Rodney Topor (Nathan)
Enrolment Restrictions:	Nil
This document was last updated:	14 June 2007

BRIEF COURSE DESCRIPTION

This course presents the basic theory of regular languages, context- free languages, computability and computational complexity. It describes which computational problems can and can't be solved, which computational problems can and can't be solved efficiently, assuming different models of computation.

Prerequisites: 1002ICT, 1005ICT, 1006ICT or equivalents.

SECTION A – TEACHING, LEARNING AND ASSESSMENT

COURSE AIMS

Course Aims

This course introduces the basic theory of formal languages, computing machines, computability, and computational complexity. It provides basic understanding of the nature and capabilities of different computing models, and develops students' reasoning and problem-solving skills.

LEARNING OUTCOMES

After completing the course, students should understand how real computational problems may be represented in abstract computational models; they should have a good understanding of the relative computational capabilities and limitations of several different abstract models; they should understand and be able to reason about the relationships between languages and models; they should be able to prove that certain languages can or cannot be recognised in different models, and in particular that some languages are uncomputable; they should understand the relative power of deterministic and nondeterministic computation in different models; they should understand how to represent the complexity of computational problems, how to formally capture the difference between tractable and intractable computation, how to compare the complexity of different problems, and how to prove certain problems are intractable. More generally, students should have gained confidence and competence at reasoning carefully about key concepts in the theory of computation, should have developed their problem-solving skills, and should have a good foundation for further study of theoretical computer science.

CONTENT, ORGANISATION AND TEACHING STRATEGIES

The content of the course covers the following topics:

- Introduction to basic concepts of languages, automata and computation theory.
- Regular languages: regular expressions, finite automata, equivalence of deterministic and nondeterministic finite automata, equivalence of regular expressions and finite automata (Kleene's Theorem), uniqueness of minimal finite automata, properties of regular languages, decision problems for regular languages.
- Context-free languages: context-free grammars, pushdown automata, equivalence of context-free languages and nondeterministic pushdown automata, inequivalence of determininistic and nondeterministic pushdown automata, properties of context-free languages, decision problems for context-free languages.
- Theory of computability: Turing machines, equivalence of deterministic and nondeterministic and other variants of Turing machines, equivalence of recursive functions and random access machines to Turing machines, the Church-Turing thesis, existence and examples of uncomputable problems, Rice's Theorem.
- Theory of complexity: Tractable and intractable problems (P *vs* NP), preservation of P under various models of computation, polynomial reducibility, NP-completeness, examples of NP-complete problems, definitions and examples of other cocmplexity classes.
- Applications to string matching, parsing, graph theory, problem solving and other domains.

The course is taught using traditional lectures and tutorials. There are two one-hour lectures and one one-hour tutorial each week. New material is introduced, explained and illustrated during lectures, together with many examples of problem solving techniques. Tutorials are provided for students to reinforce their learning by attempting graded problems and getting feedback on their attempts, and as an opportunity to ask questions about the lecture material. Some software packages which provide self-study opportunities are used to supplement this approach.

The convenor will be available for consultation for two hours each week. A detailed course Web site will serve as an important information source for students. An online discussion forum will be provided as an important mechanism for students to communicate with each other and with the teaching team.

Lectures, tutorials and assessment items (described below) all contribute to developing students reasoning and problem-solving skills.

Contact Summary

Students are required to attend the two lectures each week and to attend and participate in a tutorial each week. Failure to attend or to participate may be taken into consideration by the teaching team if students request out-of-hours assistance, special consideration, deferred examinations, and so on.

CONTENT SUMMARY

The following table indicates what material is covered when. Tutorials will review material covered in lectures in the previous and current week.

Topic	Lecture Content	Weeks	Readings
1.	Introduction	Week 1	Sipser, Chapter 0
2.	Regular languages	Weeks 2–4	Sipser, Chapter 1
3.	Context-free languages	Weeks 5–6	Sipser, Chapter 2
4.	Computability theory	Weeks 7–9	Sipser, Chapters 3–4 and Sections 5.1 and 6.1
5.	Complexity theory	Weeks 10–12	Sipser, Chapter 7 and Sections 8.2 and 8.3
6.	Other topics and review	Week 13	

ASSESSMENT

Summary of Assessment

Item	Assessment Task	Length	Weighting	Total Marks	Relevant Learning Outcomes	Due Day and Time
1.	Four short take-home tests.	Approx. 500 words	10% each	40	Reasoning and problem solving	Monday of weeks 4, 7, 10 and 13
2.	Mid-semester exam	1 hour	20%	20	Ditto	Week 8 or 9
3.	Final exam	3 hours	40%	40	Ditto	End of semester

Assessment Details

The nature of the course content requires that learning be cumulative; each topic builds on previous topics. The frequent assessment is designed to encourage students to keep up to date. The more practice students get in reasoning and problem-solving the better they learn. The take-home tests will be distributed in lectures and on the course Web site, together with assessment criteria and submission instructions.

The final grade is based on a simple sum of marks gained in the assessment items.

Return of Assessment Items

Details of the mechanism for return of marked assignments will be published on the course Web site.

Notification of Availability of Feedback on Assessment

All reasonable efforts will be made to return marked tests within 1 week and to return marked mid-semester exams within 2 weeks.

GRADUATE SKILLS

Graduate Skills	Taught	Practised	Assessed
Effective communication (written)	☐	☒	☒
Effective communication (oral)	☐	☐	☐
Effective communication (interpersonal)	☐	☐	☐
Information literacy	☒	☒	☐
Problem solving	☐	☒	☒
Critical evaluation	☐	☐	☐
Work autonomously	☒	☒	☒
Work in teams	☐	☐	☐
Creativity and innovation	☐	☒	☒
Ethical behaviour in social / professional / work environments	☒	☒	☒
Responsible, effective citizenship	☐	☐	☐

Professional Skills

Take-home tests are required to demonstrate a high level of competence at mathematical and grammatical presentation.

TEACHING TEAM

Course Convenor

Convenor Details	Gold Coast
Campus Convenor	Professor Rodney Topor
Email	r.topor@griffith.edu.au
Office Location	N44 2.20
Phone	3735 5047
Fax	3735 5051
Consultation times	See course Web site.

Additional teaching team members

Details of any additional teaching team members will be published on the course Web site.

COURSE COMMUNICATIONS

A detailed course Web site is provided as an essential mechanism for the teaching team to distribute material to students and to communicate with students. Students should check this Web site regularly. An online discussion forum is provide as an important mechanism for students to communicate with each other and with the teaching team members. Students are encouraged to use this discussion forum frequently. Students may consult the convenor immediately after classes, during the specified consulting hours, and at any time by email.

TEXTS AND SUPPORTING MATERIALS

Students are expected to have their own copy of the **prescribed** textbook:

- **M. Sipser, *Introduction to the Theory of Computation, Second Edition*, Course Technology, 2004.**

The following texts are useful **supplementary** reading:

- D. Harel, *Algorithmics: The Spirit of Computing, Third Edition*, Addison-Wesley, 2004.
- J.E. Hopcroft, R. Motwani and J.D. Ullman, *Introduction to Automata Theory, Language and Computation, Second Edition*, Addison-Wesley, 2001.
- J. Martin, *Introduction to Languages and the Theory of Computation, Third Edition*, McGraw-Hill, 2003.

The software package **JFLAP** available at `http://www.jflap.org` is very useful. This package requires access to a current Java runtime system, which is available on most University systems and most home computers. The software package **GAP** available on dwarf.ict.griffith.edu.au is also very useful.

SECTION B – ADDITIONAL COURSE INFORMATION

Students should refer to the course Web site for further information about the course. A course evaluation survey will be published and students are encouraged to complete it.

SECTION C – KEY UNIVERSITY INFORMATION

ACADEMIC MISCONDUCT

Students must conduct their studies at the University honestly, ethically and in accordance with accepted standards of academic conduct. Any form of academic conduct that is contrary to these standards is academic misconduct, for which the University may penalise a student. Specifically it is academic misconduct for a student to:

present copied, falsified or improperly obtained data as if it were the result of laboratory work, field trips or other investigatory work;

include in the student's individual work material that is the result of significant assistance from another person if that assistance was unacceptable according to the instructions or guidelines for that work;

assist another student in the presentation of that student's individual work in a way that is unacceptable according to the instructions or guidelines for that work;

cheat; (Cheating is dishonest conduct in assessment);

plagiarise (Plagiarism is knowingly presenting the work or property of another person as if it were one's own.)

Visit the University's *Policy on Academic Misconduct* for further details.

PLAGIARISM DETECTION SOFTWARE

In semester 2, 2007 the University is piloting the use of plagiarism detection software. Students should be aware that your Course Convenor may use this software to check submitted assignments. If this course is included in the pilot your Course Convenor will provide more detailed information about how the detection software will be used.

KEY STUDENT-RELATED POLICIES

All University policy documents are accessible to students via the University's Policy Library website at: www.griffith.edu.au/policylibrary. Links to key policy documents are included below for easy reference:

Student Charter

Student Administration Policy

Assessment Policy

Academic Calendar

Health and Safety Policy

Academic Standing, Progression and Exclusion Policy

Policy on Student Grievances and Appeals

Examinations Timetabling Policy and Procedures

Guideline on Student E-Mail

UNIVERSITY SUPPORT RESOURCES

The University provides many facilities and support services to assist students in their studies. Links to information about University support resources available to students are included below for easy reference:

Learning Centres - the University provides access to common use computing facilities for educational purposes. For details visit www.griffith.edu.au/cuse

Learning@Griffith - there is a dedicated website for this course via the Learning@Griffith student portal.

Student Services facilitate student access to and success at their academic studies. Student Services includes: Careers and Employment Service; Chaplaincy; Counselling Service; Health Service; Student Equity Services (incorporating the Disabilities Service); and the Welfare Office.

Learning Services within the Division of Information Services provides learning support in three skill areas: computing skills; library skills; and academic skills. The study skills resources on the website include self-help tasks focusing on critical thinking, exam skills, note taking, preparing presentations, referencing, writing, proof reading, and time management.