



Philadelphia University
Faculty of Information Technology
Department of Computer Science
First Semester, 2007/2008

Course Syllabus

Course Title: Theory of Computation	Course code: 751323
Course Level: 3	Course prerequisite(s) and/or corequisite(s): 210104 + 721211
Lecture Time:	Credit hours: 3

Academic Staff Specifics

Name	Rank	Office Number and Location	Office Hours	E-mail Address

Course Description:

This module introduces the theory of computation through a set of abstract machines that serve as models for computation - finite automata, pushdown automata, and Turing machines - and examines the relationship between these automata and formal languages. Additional topics beyond the automata classes themselves include deterministic and nondeterministic machines, regular expressions, context free grammars, undecidability, and the P = NP question.

Course Objectives:

Finite automata are useful models for many important kinds of hardware and software. Here are the most important kinds: Software for designing and checking the behavior of digital circuits; The “lexical analyzer” of a typical compiler, that is, the compiler component that breaks the input text into logical units, such as identifiers, keywords, and punctuation; Software for scanning large bodies of text, such as collections of Web pages, to find occurrences of words, phrases, or other patterns; Software for verifying systems of all types that have a finite number of distinct states, such as communication protocols or protocols for secure exchange of information.

Course Components

- Basic concepts and definitions; Set operations; partition of a set
- Equivalence relations; Properties on relation on set.
- Proving Equivalences about Sets.
- Central concepts of Automata Theory.
- Regular Expressions; Operations on Regular expressions

- Finite Automata and Regular Expressions.
- Conversion from FA and regular expressions.
- Deterministic Finite Automata (DFA); Minimization of DFA.
- Non-Deterministic Finite Automata (NFA).
- Equivalence of Deterministic and Non-Deterministic Finite Automata.
- Equivalence between DFA, NFA, NFA- Λ
- Context-Free Grammars.
- Parse Trees; Ambiguity in Grammars and Languages.
- Standard Forms; Chomsky Normal Forms;
- Greibach normal Forms.
- Minimization of CFG's.
- Pushdown Automata (PDA).
- Deterministic and Non-Deterministic (PDA); Formal definition of NPDA.
- Transition functions of NPDA; NPDA Execution.
- Accepting Strings with NPDA; Equivalence of PDAs and CFG.
- The Turing Machine.
- Programming Techniques for Turing Machines; Formal definition of TM's.
- TM's as acceptors; TM's as transducers; Recognizing Languages with TM's.; Sorting with TM's.; Programming in TM's.
- Multiple Tracks, Subroutines, Complexity issues and analysis

Text book:

Title: **Introduction to Computer Theory**

Author: **Daniel I. A. Cohen**

Publisher: **Prentice-Hall, Second Edition, 1997.**

In addition to the above, the students will be provided with handouts by the lecturer.

Teaching Methods:

Duration: 16 weeks, 64 hours in total

Lectures: 38 hours (2-3 per week),

Tutorials: 10 hours (on average, 1 per week)

Home work: 3 Assignments

Learning Outcomes:

- **Knowledge and understanding**
 - Acquire a full understanding and mentality of Automata Theory as the basis of all computer science languages design
 - Have a clear understanding of the Automata theory concepts such as RE's, DFA's, NFA's, Stack's, Turing machines, and Grammars
- **Cognitive skills (thinking and analysis).**
 - Be able to design FAs, NFAs, Grammars, languages modelling, small compilers basics
 - Be able to design sample automata
- **Communication skills (personal and academic).**
 - Be able to minimize FA's and Grammars of Context Free Languages
- **Practical and subject specific skills (Transferable Skills).**

Assessment Instruments

<u>Allocation of Marks</u>	
Assessment Instruments	Mark
First examination	15%
Second examination	15%
Final Exam (written unseen exam)	50%
Reports, research projects, Quizzes, Home works, Projects	20%
Total	100%

** Make-up exams will be offered for valid reasons only with consent of the Dean. Make-up exams may be different from regular exams in content and format.*

Practical Submissions

The assignments that have work to be assessed will be given to the students in separate documents including the due date and appropriate reading material.

Documentation and academic honesty

Submit your home work covered with a sheet containing your name, number, course title and number, and type and number of the home work (e.g. tutorial, assignment, and project).

Any completed homework must be handed in to my office (room ---) by 15:00 on the due date. After the deadline “zero” will be awarded. You must keep a duplicate copy of your work because it may be needed while the original is being marked.

You should hand in with your assignments:

- 1- A printed listing of your test programs (if any).
- 2- A brief report to explain your findings.
- 3- Your solution of questions.

• Protection by Copyright

1. Coursework, laboratory exercises, reports, and essays submitted for assessment must be your own work, unless in the case of group projects a joint effort is expected and is indicated as such.
2. Use of quotations or data from the work of others is entirely acceptable, and is often very valuable provided that the source of the quotation or data is given. Failure to provide a source or put quotation marks around material that is taken from elsewhere gives the appearance that the comments are ostensibly your own. When quoting word-for-word from the work of another person quotation marks or indenting (setting the quotation in from the margin) must be used and the source of the quoted material must be acknowledged.
3. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.

• Avoiding plagiarism.

1. Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.
2. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and

changing a few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.

3. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.
4. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty the Department will take into account factors such as the year of study, the extent and proportion of the work that has been plagiarized, and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

Course/Module Academic Calendar

Week	Basic and support material to be covered	Homework/reports and their due dates
(1)	Basic concepts and definitions Set operations; partition of a set Equivalence relations; Properties on relation on set; Proving Equivalences about Sets. Central concepts of Automata Theory.	
(2)	Regular Expressions; Operations on Regular expressions Finite Automata and Regular Expressions. Recursive definitions; Conversion from FA and regular expressions; Kleen's Theory; Mealy Moore Machines. Conversion from Mealy to Moore and vice versa.	Tutorial 1
(3)	Deterministic Finite Automata (DFA). Minimization of DFA; Non-Deterministic Finite Automata (NFA).	Tutorial 2
(4)	Equivalence of Deterministic and Non-Deterministic Finite Automata.	Tutorial 3
(5)	Finite Automata with Epsilon-Transition. Equivalence between DFA, NFA, NFA- Λ	Assignment 1
(6)	Pumping Lemma for Regular Languages. Closure Properties of Regular Languages.	Tutorial 4
(7)	Context-Free Grammars; Regular Grammars; Parse Trees.	Tutorial 5
(8)	Ambiguity in Grammars and Languages. Standard Forms; Chomsky Normal Forms; Greibach normal Forms.	Tutorial 6
(9)	Pumping Lemma for Context-Free Languages; Closure Properties of Context-Free Languages; Minimization of CFG's.	
(10)	Pushdown Automata (PDA).	Tutorial 7; Assignment 2
(11)	Deterministic and Non-Deterministic (PDA); Formal definition of NPDA. Transition functions of NPDA; NPDA Execution; Accepting Strings with NPDA; Equivalence of PDAs and CFG.	Tutorial 8
(12)	The Turing Machine.	Tutorial 9, Assignment 3

(13)	Programming Techniques for Turing Machines; Formal definition of TM's. TM's as acceptors; TM's as transducers; Recognizing Languages with TM's.; Sorting with TM's.; Programming in TM's	Tutorial 10
(14)	Multiple Tracks, Subroutines, Complexity issues and analysis	
(15) Specimen examination (Optional)	Equivalence of PDA's and CFG.	
(16) Final Examination	Revision	

Expected workload:

On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.

Attendance policy:

Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Module References

Students will be expected to give the same attention to these references as given to the Module

1. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, 1998
2. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory, Languages, and Computation", Second Edition, Prentice-Hall, 2001
3. Peter Dehning, Jack B. Dennis, "Machines, Languages and Computation", Second Edition, Prentice-Hall, 1978
4. Harry R. Lewis, Christos H. Papadimitriou, "Elements of the theory of computation", Second Edition, Prentice-Hall, 1998

Simulators:

In order to improve the pedagogy of this course, interactive animations of the various automata using available simulators are recommended.

General & Introductory Computer Science. Computer Science Special Topics. Introduction to Computer Theory, 2nd Edition.

Description. About the Author. This text strikes a good balance between rigor and an intuitive approach to computer theory. Covers all the topics needed by computer scientists with a sometimes humorous approach that reviewers found "refreshing." The goal of the book is to provide a firm understanding of the principles and the big picture of where computer theory fits into the field. About the Author.

Daniel Isaac Aryeh Cohen is an American mathematician and computer scientist who is now a professor emeritus at Hunter College.

Permissions. Request permission to reuse content from this site. Table of contents. AUTOM Theory of Computer Science (Automata, Languages and Computation) Third Edition. 434 Pages 2010 16.85 MB 43,951 Downloads. Automata, Languages and

Computation. THIRD EDITION . The enlarged third edition of Thea Python Programming: An Introduction to Computer Science. 261

Pages 2012 1.2 MB 89,551 Downloads. Python Programming: An Introduction to Computer Science John M. Zelle, Ph.D. Version

1.0rc2 Fall Introduction to Insurance Mathematics: Technical and Financial Features of Risk Transfers. 479 Pages 2011 2.19

MB 2,377 Downloads New! . Pitacco, Introduction to Insurance Mathematics,. 1. Introduction to Computer Theory book. Read 12

reviews from the world's largest community for readers. This text strikes a good balance between rigor and ... This text strikes a good

balance between rigor and an intuitive approach to computer theory. Covers all the topics needed by computer scientists with a

sometimes humorous approach that reviewers found "refreshing". It is easy to read and the coverage of mathematics is fairly simple so

readers do not have to worry about proving theorems. Get A Copy.