Lamar University

COSC 3302-01, Spring 2005

Introduction to Computation Theory

Instructor: Chung-Chih Li, Ph.D.
Office: Mase 69, Tel:(409) 880-8748
URL: `http://hal.lamar.edu/~licc`
E-mail: `licc@hal.lamar.edu`
Office Hours: Tue & Thu 2:00 ∼ 3:30 PM or by appointment

Classroom and meeting time:

Mase 111, Tue & The 12:30 ∼ 1:45 PM
(Attendance will be taken impulsively)

HomePages of the course: `http://hal.lamar.edu/~licc/cosc3302`

From there, you may find important information about assignments, assignment data, due dates, sample programs, or announcements. **Note:** *An announcement made in the class will be considered as an official one, since I may not be able to update every announcement.*

Course Description and Purposes:

This course is designed to give students a general picture of the underlying theory of computation. The topics extend across three major areas in theoretical computer science, which are Automata, Computability, and Complexity. The three areas form indispensable foundations of computer science from both applicative and theoretical aspects, and hence we believe that students majoring in computer science should be familiar with the topics in a certain depth.

Each of the three areas puts different emphases on the notion of computation. The theory of automata deals with the relations between formal languages and computing machines, which directly benefits many applications in computer science such as compilers and natural language processing. We will discuss finite state machines and examine a few language families in the standard hierarch of formal languages. In particular, we will have a close look at regular languages and context-free languages.

As for computability, it is one of the oldest intellectual inquiries, tracing back to ancient Greece. In modern study, the theory systematically characterizes what can be solved by mechanical procedures, and the notion turns out to be the very idea of the modern computer. In this course, we will use Truing Machines as our standard computing formalism to explain the notion of computability. Then, some important concepts such as the well-known Church-Turing thesis, decidability, reducibility, and the recursion theorem will also be discussed.

A theoretical reflection on the study of algorithms gave birth to this relatively new research area called Computational Complexity Theory. Problems that cannot be solved efficiently by computers in many cases are not because of the lack of powerful computers or good programming skill, but because of the inherited nature of the problems. Thus, we need to provide a theoretical framework to characterize computable problems in terms of their "natural difficulties". Classes such as **P**, **NP** and **NP**-complete, **LogSPACE**, **PSPACE**, and **BPP** are some typical and well-studied complexity classes. The topics are covered in Part Three of our textbook, but we are not able to cover all of them in this course due to the time constrain.

Prerequisites:

COSC 1337 (or COSC 1374), MATH 2305.

Textbooks:

Introduction to the Theory of Computation, by Michael Sipser, PWS Publishing Company, 1997.

Reference:

Discrete Structures, Logic, and Computability, by James L. Hein, Jones and Bartlett Publishers, Inc. 2002, Second Edition.

Examinations: (400 points) Two midterms (100 points for each) and one Final Exam (200 points).

- Unless announced otherwise, all tests are accumulative, closed book, and indispensable. No makeup test will be given unless a documented absence is authorized by the university.
- Every student is allowed to bring a self-prepared crib sheet to the test. You can **write** down anything on both sides of **one** letter-sized paper. No circulation during the test.

Midterm I	100 points	Feb. 17, 6^{th} week's Thursday
Midterm II	100 points	Mar. 22, 11^{th} week's Tuesday
Final Exam	200 points	May 10, Tuesday, 11:00 AM $\sim$ 1:30 PM

Assignments: (100 points) 10 homeworks (maybe programming projects) will be given. Students are encouraged to discuss assignments and help each other. However, this does not mean that you can either entirely or partially copy or modify someone else's works.

Any form and any degree of plagiarism will receive 0 point.

Late works will be graded with penalty: -10 points per day after the due date.

Attendance: (50 points) Attendances will be taken impulsively.

Pop quizzes: (100 points)

About 10 pop quizzes will be given impulsively. Each quiz carries 10 points towards students' final scores. The coverage of every quiz is also accumulative, including the materials covered in the class right before the quiz. A typical quiz takes about 10 minutes. No makeup quiz will be given if missed. If you miss a quiz due to a university authorized absence, we will use the average of your rest quizzes; otherwise, you get a 0 for the absent quiz.

Academic Honesty:

Cheating, plagiarism, collusion, abuse of resource materials, and their consequences are defined and described under the section of Academic Affairs in the *Student Handbook*.

Students giving away academic works for assignment offered for credit to other students working on the same assignment will be considered as guilty as academic dishonesty, and will receive the same penalty.

Grading Policy:

Considering 650 points the perfect score, your grade is based on the scheme shown in the table.

I do not curve!!

Points	Grade	
540 ∼ 650	A	Excellent
420 ∼ 539	B	Good
300 ∼ 419	C	Satisfactory
200 ∼ 299	D	Passing
0 ∼ 199	F	Failure

Tentative Topics and Schedule:

Week	**Topics**	**Reading**
1: Jan. 12	(Jan. 12, first class day), mathematical preliminaries	Syllabus
2: Jan. 17	(Jan. 17, no class), mathematical preliminaries: sets, logic, functions, graphs, proofs, mathematical induction	Chapter 0
3: Jan. 24	Languages, regular languages, NFA, DFA	1.1, 1.2
4: Jan. 31	Regular expressions, non-regular languages, pumping lemma	1.3, 1.4
5: Feb. 7	Context-free languages, context-free grammars, push-down automata	2.1, 2.2
6: Feb. 14	Push-down automata, **(Midterm 1, Feb. 17, Thursday)**	2.2
7: Feb. 21	(Feb. 23, last day to drop without penalty), non-context-free languages, Turing machines	2.3, 3.1
8: Feb. 28	Truing machines, nondeterministic Truing machines, Church-Turing thesis, algorithms	3.2, 3.3
9: Mar. 7	Decidability, diagonalization, halting problems	4.1, 4.2
10: Mar. 14	Spring break, no class	
11: Mar. 21	**(Midterm 2, Mar. 22, Tuesday)**, Undecidable problems, reductions, (Mar. 25, good Friday, no class)	5.1
12: Mar. 28	Post corresponding problem, computable functions, mapping reductions	5.2, 5.3
13: Apr. 4	Big-O and small-o notations, algorithm analysis, complexity classes	7.1
14: Apr. 11	Classes **P** and **NP**	7.2, 7.3
15: Apr. 18	**NP**-completeness, Cook reductions	7.4
16: Apr. 25	Intractable problems, **NP**-complete problems	7.5
17: May 2	(May 3, last class day)	
18: May 9	**(Final Exam: May 10, Tuesday, 11:00 AM ∼ 1:30 PM)**	