Lamar University

COSC 5320, Fall 2002
Formal Languages and Automata

Instructor: Dr. Chung-Chih Li
Office: 69 Maes, tel: (409) 880-8748
URL: http://hal.lamar.edu/∼licc
E-mail: licc@hal.lamar.edu
Office Hours: MWF 11:10 AM ∼ 12:10 PM or by appointment

Class meeting times and place: (Attendance will be taken impulsively)

MWF 10:10 ∼ 11:00 AM at Maes 107

HomePages of the course:

TBA

These webpages contain detailed information about this course, assignments, due dates, and most recent announcements.

Course Description and Purpose:

This course is designed to give students a concrete concept of the underlaying theory of programming languages that plays an indispensable role in the make of modern computers. The applications of automaton theory are not limited to compilers; one can easily find the theory involved in a vast variety of applications, from simple vending machines, to natural language processing, and to sophisticated real-time control systems. Students in this course will be trained to use formal and rigorous mathematical arguments to characterize our objects. We will abstract our computation model away from the real computer, and, instead of focusing on a specific programming language, we will study the general properties of the interested language families.

At the beginning of the semester, we will spend a little while on some mathematical preliminaries that pave the way for being able to comprehend the materials and prove theorems in this course. Then, we will focus on different grammars that specify formal languages at different hierarchies. We will also introduce several variations of automata and examine how different constraints adding to the automata will differ their power. In particular, the topics include Regular Languages, Context-Free Languages, Finite State Automata, Pushdown Automata, Pumping Lemmas, etc. By the end of the semester, if time allows, we will introduce the most powerful computation formalism – Turing Machine – on which the entire theory for modern computation prevails.

Prerequisites:

A certain maturity of mathematical concepts are essential for one to be successful in this course. Students should be familiar with some subjects in Discrete Mathematics such as logics, sets, relations, graphs, trees, mathematical induction and other proof techniques. We will try to

fix and clarify any misconceptions whenever needed in class or in instructor's office hours. However, we cannot satisfy every student who is too much behind the mathematical level we expect one should have before he/she takes this course. For example, we are not going to teach you how to find the gcd of two natural numbers.

Moreover, the programming skill experienced in COSC-2371 (Data Structure and Algorithm Analysis) or equivalent courses is required for doing the programming assignments.

Textbook:

- *An Introduction to Formal Languages and Automata,* by Peter Linz, Jones and Bartlett Publishers

We will cover most of the materials introduced in Chapters 1 to 9 of the textbook.

Reference Books:

1. *Problems on Discrete Mathematics (I, II)*, by Chung-Chih Li and Kishan Mehrotra, Syracuse University Press, 1998.
2. *Introduction to Automata Theory, Languages and Computation*, by John E. Hopcroft and Jeffrey D. Ullman, Addison-Wesley Publishing Company, 1979.
3. *Computability, Complexity, and Languages*, by Martin Davis and Elaine J. Weyuker, Academic Press, 1983.

Note: Students who are lack of mathematical background can quickly pick up the required concepts from the first part of the book listed in 1 above. The book is out of print, but I will reserve a copy of Part I in the main library. Books listed in 2 and 3 are two classic textbooks in the field. Some revisions and up-to-date theory are added to their new editions.

Examinations and Dates: (400 points)

All tests are accumulative, closed book, and indispensable. No makeup test will be given unless a documented absence is authorized by the university.

Midterm 1	100 points	(6th week) Sep. 23, 10:10 $\sim$ 11:00 AM
Midterm 2	100 points	(11th week) Oct. 28, 10:10 $\sim$ 11:00 AM
Final Exam	200 points	(17th week) Dec. 9, 11:00 AM $\sim$ 1:30 PM

- Every student is allowed to bring a self-prepared crib in one letter-sized paper to the test.

Assignments: (120 points)

Three programming assignments will be offered. Each programming assignment carries 40 points towards students' final scores (the perfect score is 600 points). Late works will be graded with penalty; -5 points per day after the due date.

Students are encouraged to discuss assignments and help each other. However, this does not include **either entirely or partially copying or modifying** someone else's programs.

Attendance: (50 points or more)

Each attendance, if taken, contributes 5 points towards students' final scores. In other words, an absence on the day the roll is checked costs 5 points.

Pop quizzes: (50 points or more)

A few 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 if the quiz is given a few minutes before the end of the class. A typical quiz takes about 10 minutes. On the day a quiz is given, the attendance will not be taken. No makeup quiz will be given if missed. A missed quiz due to a university authorized absence on the quiz day will use the average of the rest quizzes as a substitute.

Grading Policy: (600 is considered the perfect score)

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

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.

Special Notes to The Class:

- Backup your works. "My dog ate my disk!!" is not a good excuse.
- Since the more often I take the attendance the more points you may get, you will be better off if I can check the roll efficiently on a daily basis. In order to take the attendance in an efficient way, we will have a seat arrangement for this class. This also helps me to remember your name. Your seat will be fixed to where you sit in the second class of the semester. So, before the second class begins, please decide a seat (negotiate with your classmates, if necessary) where you feel comfortable to sit in for the rest of the semester. However, where you sit in the class will not affect your grade in any way.
- Although there is no team-work project in this course, every student is encouraged to joint a study group for exchanging ideas and discussing materials covered in this course. A typical study group should have 3 to 5 students who can meet conveniently and regularly during the semester outside the classroom. If you can't form a study group by the end of the second week, the instructor will do that for you. None of the activities of the study groups will directly connect to the grade.
- Every student has to prepare a letter-sized Manila folder with the course number and your name written on the tab of the folder. After the study group is determined, write the name list of your group on the page of the folder (facing inside), and give the folder to the instructor. The folder will be used to hold your assignments, tests, attendance, and any personal information regarding the class.

Tentative Topics and Schedules for COSC 5320

Week	Topics	Reading
1: Aug. 21	Introduction to theoretical computer science, mathematics preliminaries: sets, logics, graphs, trees, relations, proofs, and so on.	Lecture note 1.1 $\sim$ 1.3
2: Aug. 26	More on mathematics preliminaries, finite automata, languages, deterministic vs. nondeterministic automata.	2.1$\sim$2.3
3: Sep. 2	Equivalence and reduction of automata, regular expressions, regular languages.	2.3$\sim$3.2
4: Sep. 9	Regular grammars, right- and left-linear grammars.	3.3
5: Sep. 16	Closure properties and membership of regular languages.	4.1$\sim$4.2
6: Sep. 23	**(Midterm 1)**, identifying non-regular languages: the use of the pigeonhole principle and a pumping lemma.	4.2$\sim$4.3
7: Sep. 30	Context-free languages, leftmost and rightmost deviations, deviations trees, sentential forms.	5.1$\sim$5.2
8: Oct. 7	Context-free grammars and programming languages, simplification.	5.3$\sim$6.1
9: Oct. 14	Chomsky normal forms, Greibach normal forms, a parser algorithm.	6.2$\sim$6.3
10: Oct. 21	Pushdown automata and context-free languages, deterministic vs. nondeterministic pushdown automata.	7.1$\sim$7.3
11: Oct. 28	**(Midterm 2),** pumping lemmas for context-free languages and linear languages.	7.4$\sim$8.1
12: Nov. 4	More on pumping lemmas, closure and other properties of context-free languages.	8.1$\sim$8.2
13: Nov. 11	Turing Machines.	9.1$\sim$9.2
14: Nov. 18	Gödel numbers, URM's, and Turing Machines, Church-Turing's thesis.	Lecture note
15: Nov. 25	Catch-up, review for the final. (Thanksgiving Holiday).	
16: Dec. 2	Prepare for the final.	
17: Dec. 9	**Final Examination.**	