# Summer 2009 :: COM SCI 181, Sec.1 :: Introduction to Formal Languages and Automata Theory :: Syllabus/Info

## Class Information

| | | |
|---|---|---|
| Location, day, time:<br>*from the Registrar | **Section** | TR 4:00 PM - 5:50 PM BOELTER 5249 |
| | **Discussion 1A** | F 4:00 PM - 5:50 PM BOELTER 5249 |
| Final Exam Code:<br>*from the Registrar | *Currently unavailable* | |
| Course description: | **Lecture, four hours; discussion, two hours; outside study, six hours. Requisites: course 32, and Mathematics 61 or 180. Designed for junior/senior Computer Science majors. Grammars, automata, and languages. Finite-state languages and finite-state automata. Context-free languages and pushdown story automata. Unrestricted rewriting systems, recursively enumerable and recursive languages, and Turing machines. Closure properties, pumping lemmas, and decision algorithms. Introduction to computability. Letter grading.** | |

## Instructor Information


| | |
|---|---|
| Name: | Mark Burgin (markburg@cs.ucla.edu) |
| Office location: | Boelter Hall, 4532M |
| Office hours: | T: 3 - 4 PM, Th: 3 - 4 PM |
| Phone: | 310 206 2099 |

# Introduction to Formal Languages and Automata Theory

0. Basic concepts and constructions:
    a. Mathematics
    b. Linguistics
    c. System theory
    d. Information theory

## 1. Subrecursive algorithms

1.1. Finite automata and regular languages

1.2. Pushdown automata and context-free languages

## 2. Recursive algorithms

2.1. Turing machines

2.1.1. Computability

2.1.2. Complexity and Tractability

2.2. Partial recursive functions

## 3. Superrecursive algorithms

3.1. Inductive Turing machines

3.1.1. Computability

3.1.2. Complexity and Tractability

3.2. Grid automata

Hopcroft, John, Motwani, Rajeev, and Ulman, Jeffrey, *Introduction to Automata Theory, Languages and Computation*, Third Edition, Addison-Wesley, 2007

Burgin, Mark, *Superrecursive algorithms*, Springer Verlag, 2005

Cohen, D.I.A. *Introduction to Computer Theory*, Second Edition, John Wiley & Sons, 1997

## Summer 2009 :: COM SCI 181, Sec.1 :: Introduction to Formal Languages and Automata Theory :: Accreditation > Information

| | | | |
|---|---|---|---|
| **Course:** | COM SCI 181 | **Effective Term:**<br>**Discontinued Term:** | 08S |
| **Title:** | Introduction to Formal Languages and Automata Theory | **Units:** | 4.0 |
| **Description:** | Lecture, four hours; discussion, two hours; outside study, six hours. Requisites: course 32, and Mathematics 61 or 180. Designed for junior/senior Computer Science majors. Grammars, automata, and languages. Finite-state languages and finite-state automata. Context-free languages and pushdown story automata. Unrestricted rewriting systems, recursively enumerable and recursive languages, and Turing machines. Closure properties, pumping lemmas, and decision algorithms. Introduction to computability. Letter grading. | | |
| **Instructors in charge:** | • Sheila Greibach<br>• Rafail Ostrovsky<br>• Amit Sahai | | |
| **Requirement:** | • Computer Science degree - REQUIRED<br>• Computer Science & Engineering degree - REQUIRED | | |
| **Objective:** | | | |
| **Undergraduate Program Objectives:** | http://www.seasoasa.ucla.edu/program/cs_obj.htm<br>http://www.seasoasa.ucla.edu/program/ce_obj.htm | | |

**Textbook:**

| Author/Title | Publisher/Year | Req |
|---|---|---|
| GREIBACH<br>APS INTRODUCTION TO FORMAL LANGUAGE AND THEORY | ACADEMIC PUBLISHING SERVICE | |
| HOPCROFT<br>INTRO AUTOMATA THEORY (3RD) : LANGUAGES & COMPUTATION | ADDISON<br>2007 | |
| SIPSER<br>INTRO THEORY OF COMPUTATION (2ND) | CENGAGE LEARNING<br>2005 | |

| | Course Outcomes (4) | Program Outcomes |
|---|---|---|
| a) | **Understanding of finite state systems, their specifications and properties and building them to meet a task** | |
| b) | **Understanding of regular expressions and their connection to finite state machines** | |
| c) | **Understanding of context-free languages** | |
| d) | **Basic understanding of Turing machines and undecidable problems** | |

**Entire Course**
**0=none or insignificant, 1=some, 2=moderate, 3=strong**
**Yellow boxes indicate highlighted objectives**

**Accreditation Category Content**

| | |
|---|---|
| **Engineering Science Units:** | 4 |

**Accreditation Category Content**

| | |
|---|---|
| **Engineering Design Units:** | 0 |
| **Laboratory Units:** | 0 |

Note: Engineering Science and Engineering Design units must add up to the total number of course units.
Note: Laboratory units can be any number from zero up to the total number of course units.

Last updated 2009-11-03 14:48:41

# CS 181 : Formal Languages and Automata Theory, Winter 2008

Professor Rupak Majumdar
Computer Science
University of California at Los Angeles

## Course Information

### Lectures

Mondays and Wednesdays 12:00-1:50 in 5249 Boelter Hall

### Discussion Sections

- Friday 12:00-1:50 Boelter 5280 Brian Tagiku
- Friday 2:00-3:50 Boelter 5280 Kunal Bindal

### Course Calendar

First lecture: January 7
Mid Term: February 11
Last lecture: March 12
Final exam: 8:00-11:00 on Wednesday, March 19

### Instructor

Rupak Majumdar
4531E Boelter Hall, 825-8127, r u p a k @ cs . ucla . edu
Office hours: Mondays 2:00-3:00 in 4531E Boelter Hall, and by appointment

### Teaching Assistants

Brian Tagiku
Office hours: Tue:1-3pm 4428 Boelter Hall and by appointment (b t a g i k u @ gmail.com )
Kunal Bindal
Office hours: Mon:2-4pm 4428 Boelter Hall and by appointment (k b i n d a l @ cs. ucla. edu)

The CS 181 staff is always glad to help you. Whenever you get lost, desire help, or just want to talk, please see the instructor or the TA--don't let yourself fall behind! If you can't come to one of our office hours, send us email to set up an individual appointment.

### Prerequisites

CS 32, and Math 61 (Discrete Structures) or Math 113 (Combinatorics), or permission of the instructor. Knowledge of algorithms at the CS180 level will be useful but not necessary.

## Textbooks

The required text is

*Introduction to the Theory of Computation, Second Edition* by Michael Sipser, Course Technology, 2005.

This is a new edition of the book. If you already have the old edition of the book:

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

it should still be ok. The 2nd edition is nice because it has more examples and exercises, and better explanations. I strongly recommend getting the text book, and preferably the new edition.

You may find the following books useful as supplementary references.

*Introduction to Automata Theory, Languages, and Computation* by John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, Addison-Wesley, 2000. [Equivalent to Sipser in the material covered.]

*Computational Complexity* by Christos H. Papadimitriou, Addison-Wesley, 1994. [This is a more advanced book, and suitable for further study.]

## Syllabus

CS 181 presents some of the most fundamental results in theoretical Computer Science. These results attempt to answer, in a precise mathematical sense, the following two questions, which are of obvious practical as well as philosophical interest:

- Can a given problem be solved by computation?
- How efficiently can a given problem be solved by computation?

Thus, unlike CS 180, we focus on *problems* rather than on specific *algorithms* for solving problems. To answer both questions mathematically, we need to start by formalizing the notion of "computer" or "machine". So the course outline breaks naturally into three parts:

1. Models of computation (*Automata Theory*)
   - Finite automata
   - Push-down automata
   - Turing machines
2. What can we compute? (*Computability Theory*)
3. How efficiently can we compute? (*Complexity Theory*)

## Course Requirements

You are expected to be familiar with all the material covered in the lectures and in the relevant sections of the textbook. Attending lectures is highly advisable, because some of the lectures and homework problems will be drawn from material not in the text, and you will be responsible for this material. There will be 8 weekly homework problem sets, a midterm exam, and a final exam.

## Reading Assignments

In general there will be no specific reading assignments. However, the relevant sections from the textbook will be pointed out in class. It is recommended that you make yourself familiar with the material prior to class, and it is a good strategy to read the pertinent sections again right after class to make sure you understand them.

## Homework Assignments

A problem set will be handed out each Wednesday and is due *at the beginning of class* on the following Wednesday. If you cannot attend a lecture, you may turn in your homework in the drop box A? in 4428 BH before 12:00 noon on the due date. Late homeworks will not be accepted.

The solution to each homework problem should be written legibly on a separate sheet of paper that contains your name, the homework number, the problem number, Discussion Section Number, and "CS 181 -- Winter 2008". We will not attempt to grade messy and unreadable solutions. If a problem can be interpreted in more than one way, clearly state the assumptions under which you solve the problem.

In writing up your homework you are allowed to consult any book, paper, or published material. If you do so, you are required to cite the complete bibliographical data of your source(s). Simply copying a proof is not sufficient; you are expected to write it up in your own words, and you must be able to explain it if you are asked to do so. Your proofs may refer to previous course material and to previous homeworks. Except for this, all results you use must be proved explicitly.

Our goal is to get across a maximum amount of understanding in a minimum amount of time. Since you have other courses, we will try to monitor the time you spend on this course. It is easy, however, to misjudge the time required to solve a problem, so we ask you to indicate with each problem set how much time you spent studying and how much time you spent completing the assignment. Roughly, you are expected to spend one hour reading and two hours problem solving for every hour of lecture.

Model solutions will be handed out and discussed during the discussion sections on the day after a homework is due. The graded problem sets will be returned a week later also during the discussion sections. If you have questions about the grading, please talk to the TA. Graded problem sets that are not picked up in the discussion sections will be kept in the TA's office. Copies of all handouts, problem sets, and model solutions will be available on the web at

http://www.cs.ucla.edu/~rupak/Courses/winter08/CS181.

It is extremely important that you continuously stay on top of the material, because every new topic and every new homework builds on previous results. If you don't understand the material at the beginning, it will be difficult to catch up later. If you encounter problems, you are encouraged to talk to the course staff as soon as possible. Please do not wait until the last moment to do your homework--start thinking about the problems on the day they are handed out!

## Examinations

There will be an in-class midterm, scheduled on February 11. The final exam will take place during the regularly scheduled exam period on March 19 from 8 am to 11 am.

The final exam will be comprehensive. In all exams you will be allowed to bring any book or notes of your choice, that is, exams are open book. However, you may not use internet access on any device during an exam.

## Grading Policy

Your final grade will be based on the sum of your problem set, prelim, and final exam scores.

- Each of the 8 weekly problem sets will be worth 30 points. The two lowest homework scores will be dropped. You may also skip handing in up to two problem sets; in this case, these will be the scores that are dropped. Because of the policy of dropping the lowest homework score, *excuses for late or missed homeworks will not be accepted.*
- The midterm will count 120 points.
- The final exam will be scored out of 200 points.

Accordingly, out of a possible maximum of 500 points, the homework will count for 36% of the final grade, the midterm will count for 24%, and the final exam will count for 40%.

## Academic Integrity

The work you submit in this course must be the result of your individual effort. You may discuss homework problems and general proof strategies or algorithms with other students in the course, but you must not collaborate in the detailed development or actual writing of problem sets. This implies that one student should never have in his or her possession a copy of all or part of another student's homework. It is your responsibility to protect your work from unauthorized access. In writing up your homework you are allowed to use any book, paper, or published material. However, you are not allowed to ask others for specific solutions, either in person or by using electronic forums such as newsgroups. Of course, during the administration of exams any form of cooperation or help is forbidden.

Academic dishonesty has no place in a university; it wastes our time and yours, and it is unfair to the majority of students. Any dishonest behavior will be severely penalized and may lead to failure in the course.

---

*http://www.cs.ucla.edu/~rupak/Courses/winter08/CS181/info.html*

*Last updated on January 7, 2008.*