Course Syllabus: CS 215: Theory of Computation
Winter, 2004

Course Description: Automata theory, formal languages, Turing machines, computability theory and reducibility, computational complexity, determinism, nondeterminism, time hierarchy, space hierarchy, NP-completeness, selected advanced topics.

Prerequisite: CS 150 or equivalent. CS 141 and Math 112 would be very helpful. The students are expected to have a strong background in discrete mathematics including the basics of symbolic logic, set theory, number theory, graph theory, combinatorics, and general proof techniques including mathematical induction and proof by contradiction, and a knowledge of data structures and some programming language. You should be able to follow the proofs presented in the class as well as create your own.

Instructor: Tao Jiang, Surge 330, phone: 787-2991, email: `jiang@cs.ucr.edu`. Office Hours: Tu and Th, 3:00-4:00pm.

Textbooks:
Michael Sipser, *Introduction to the Theory of Computation*, First Edition, 1997, PWS Publishing Company.

Lecture Notes: Copies of transparencies used in lectures (courtesy of Mitsu Ogihara) will be available on the class homepage `www.cs.ucr.edu/~jiang/215-homepage.html`

Reference Books (available at the library):
Christos Papadimitriou, *Computational Complexity*, 1994 , Addison-Wesley.
John Hopcroft and Jeffrey Ullman, *Introduction to Automata Theory, Languages, and Computation*, 1979, Addison-Wesley. (or the second edition)
Tao Jiang, Ming Li, and Bala Ravikumar, Formal models and Computability, in *Handbook of Computer Science*, CRC Press, 1996.

Evaluation:
Homeworks (and possibly quizzes) — 50%
Final examination — 50%

Reading assignment: You are expected to review, in advance, the material to be covered in each class. There will be class handouts from time to time.

Academic dishonesty: Many students find it helpful to consult their peers while doing assignments. This practice is legitimate and to be expected. However, it is not acceptable practice to pool thoughts and produce common answers. To avoid this situation, it is suggested that students not write anything down during such talks, but keep mental notes for later development of their own. Major occurrences of academic dishonesty, such as the submission of work that is not the student's own, will be dealt with according to the departmental policy on academic dishonesty which can be found at webpage `http://www.cs.ucr.edu/curriculum/acad_hone` Students who allow their files or assignments to be copied are as guilty of academic dishonesty as those who copy and will be treated accordingly. Each student is responsible for taking reasonable precautions to ensure that his/her work is not available for unauthorized use.

Tentative Schedule:

```
Preliminaries (4 lectures)
     Overview and fundamental concepts
     Regular languages and finite automata
     Context-free languages and pushdown automata
     Pumping lemmas
```

Computability Theory (6 lectures)

- Turing machines
- Variants of turing machine, nondeterministic TM
- Decidability
- Reducibility
- PCP and mapping reducibility
- Recursion Theorem

Time Complexity (5 lectures)

- Time complexity classes and hierarachy
- The classes P and NP
- Intractability and NP-completeness
- Examples of NP-completeness proofs

Space Complexity (3 lectures)

- Space complexity classes and hierarachy
- Savitch’s Theorem
- PSPACE and PSPACE-cpmpleteness
- L, NL, and NL-completeness