MTH 350 – Mathematical Logic, Section 5414

The Theory of Computation

Spring Semester 2012

Professor: Gunter Fuchs
Office: 1S-225
e-mail: gunter.fuchs@csi.cuny.edu
Office Hours:
Tuesday 12:00-1:40p, Thursday 12:00-12:50p
Telephone: 718-982-4182
Website: http://www.math.csi.cuny.edu/~fuchs/

Prerequisites: MTH 233 (Calculus III) or MTH 236 (Accelerated Calculus II)

Meeting Time and Places: Tuesday 10:10-12:05 in 1S-107, Thursday 10:10-12:05 in 1S-219

Special Dates: No class on Feb. 21 – classes follow a Monday schedule that Tuesday.

Required Reading: Michael Sipser. *Introduction to the Theory of Computation.* Second Edition. Course Technology CENGAGE Learning. ISBN-13: 978-0-534-95097-2

Course Description and Objectives: The theme this semester will be the notion of computability. What does it mean to say that a function is computable? To answer, we will study the concepts of finite automata, pushdown automata, Turing machines and other concrete models for the theoretical analysis of the foundations of computing. We will investigate how these theoretical machines work, study the class of functions they compute, and analyze the problems they can solve. Dually, we will provide examples of functions which are not computable, and problems which are not decidable, like the famous halting problem. We will also analyze how fast certain problems can be solved, thereby classifying problems by complexity. We will study the question of whether two complexity classes of problems, P and NP, are equal. This is the most important open question in computer science. There are some fascinating optional topics we may touch upon if time permits. Among these are the basics of information theory, randomness, oracle computations, Turing degrees, or space complexity.
A major goal of this course will also be to develop the ability to explain technical ideas or information in a clear and accurate manner. Therefore, all solutions to problems (unless otherwise specified) will be required to be fully explained and justified, with rigorous proof where this is called for. They should be presented essay style in the form of paragraphs consisting of complete English sentences, with embedded mathematical notation or expressions as required in order to best support the argument or computation at hand. The best solutions will be clear, concise and correct arguments that logically establish whatever claim was made. Problems or exercises that are predominantly computational should also be organized in this form, explaining how and why the computation was carried out. Examples of properly explained arguments can be found in the text or in the lecture.

Homework Assignments: Successful students in this class will be solving a lot of homework problems. Since the best way to learn mathematics is to do mathematics, students are required to complete the homework problems associated with each lecture. Students are expected to keep fully up-to-date with the homework assignments, completing each problem set before the next lecture. Since students can learn much from each other, collaboration on the homework is strongly encouraged, for example in small groups after each class session. Homework solutions will not be collected, but at the beginning of most classes, there will be a quiz which will test the students' understanding of the homework problems. The homework problems are of two types: the exercises review definitions and concepts, and the problems require more ingenuity.

Grading: There will be one mid-term exam and a final exam. In addition, there will be short graded quizzes in the beginning of most classes. Each quiz will get a score between 0 points (a solution with no merit) and 5 points (a perfect solution). The two lowest quiz scores of each student will be dropped. If a student misses a quiz, the result on that quiz will be 0. There will be no make-up quizzes. The grade for the class will be the average of the mid-term exam grade, with weight 35%, the final exam grade with weight 40%, and the average quiz grade, with weight 25%.

Exam Policy: I expect full attendance at the exams. If you know ahead of time that you will miss an exam, you must make prior arrangements with me before the exam. There will be no make-up exams as a general rule.

Exam Date: The tentative date for the mid-term exam is March 20.

Academic Honesty: The CUNY academic integrity policies are available from the web-site http://web.cuny.edu/academics/info-central/policies/academic-integrity.pdf. While working together on any assignment is encouraged, cheating on exams will not be tolerated.

This Spring Semester, we will have 28 sessions, each lasting 2 hours. The following is a tentative plan of the topics covered in each session. Depending on the students' needs and interests, more time might be spent on some topics, and others, towards the end, might be omitted. In particular, modules like the Post Correspondence Problem (Session 20) or Information and Randomness (Session 23), which are not fundamental for the rest of the course, might be omitted, or replaced by other topics.

Session	**Content**	**Pages**
1-2	Sets, functions, relations, languages	1-16
3	Proofs: Constructive, by contradiction and by induction	17-25
4	Finite Automata and Regular Languages	31-43
5-6	Regular Languages, Nondeterminism, and the Equivalence of NFAs and DFAs	44-56
7	Regular Languages and Regular Expressions	58-66
8	Finite Automata and Regular Expressions	66-76
9-10	Nonregular Languages and the Pumping Lemma	77-82
11	Context-free Languages and Grammars	99-108
12	Pushdown Automata	109-122
13	Non-context-free Grammars and the Pumping Lemma for Context-Free Grammars, Review for exam	123-128
14	Mid-term exam	n/a
15	Turing Machines and Computability, Review for exam	137-148
16	Variants of Turing Machines	148-154
17	Decidability	165-172
18	The Halting Problem	173-181

19	Turing Reductions	187-198
20	The Post Correspondence Problem (optional)	199-205
21	The Recursion Theorem	217-224
22	Decidable Theories	224-233
23	Information and Randomness	233-241
24	Complexity: Big-O and small-o notation	245-250
25	Analyzing Algorithms, Dependency on Models of Computation	251-256
26	The class P	256-263
27	The class NP and the P vs. NP problem	264-270
28	NP-completeness	271-283