

Master of Science in Computer Science Program

COURSE OUTLINE

School of Engineering and Computing

CSC 610 Mathematical Foundations

COURSE DESCRIPTION

A study of mathematical models of computation and theoretical foundations of computer science. Proof techniques, automata theory, Chomsky hierarchy, decidability and computational complexity are emphasized.

INSTRUCTOR

*P. Peter Dey, Ph.D. Professor, Department of Computer Science and Information Systems, National University, 3678 Aero Court, San Diego, CA 92123-1788
Tel. (858) 309-3421 Fax (858) 309-3420
Office Hours: Mon-Thursday: 1:00PM-3:00 PM, Email: pdey@nu.edu*

TEXTBOOK

John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007). *Introduction to Automata Theory, Languages, and Computation*, 3rd Edition, Addison Wesley

MAIN ALTERNATIVE RESOURCES

Daniel I. Cohen. (1997) *Introduction to Computer Theory*, 2nd Edition, John Wiley & Sons, Inc
(This is as good as the Textbook by Hopcroft, Motwani & Ullman (2007) listed above)

<http://www.asehome.org/mathfoundations/>

(Students found the notes on this site very helpful in the past)

OTHER RESOURCES/REFERENCES

Michael Sipser. (2012) *Introduction to the Theory of Computation*, 3rd Edition, Cengage Learning.
Elaine Rich. (2008) *Automata, Computability and Complexity*, Prentice Hall
R. Gregory Taylor (1998) *Models of Computation and Formal Languages*, Oxford University Press.
Efim Kinber and Carl Smith (2001) *Theory of Computing: A Gentle Introduction*, Prentice Hall.
H. R. Lewis and C. H. Papadimitriou (1998). *Elements of the Theory of Computation*, Prentice Hall
Turing, A.M. (1936). "On Computable Numbers, with an Application to the Entscheidungs problem". *Proceedings of the London Mathematical Society*. 2 (1937) **42**: 230–265

Course Learning Outcomes(CLOs)

Upon successful completion of this course, the students will be able to:

- (CLO₁) Construct a computational model for a given problem and examine its consequences.
- (CLO₂) Describe properties of computational models.
- (CLO₃) Prove: For every Non-deterministic Finite Automaton there is a regular expression.
- (CLO₄) Prove that a given language is Context-Free.
- (CLO₅) Construct a processor for a given Context-Free language.
- (CLO₆) Construct a Turing Machine for a given computational problem.

(CLO7) Prove results of union, concatenation, and complementation of various recursively enumerable sets.

COURSE REQUIREMENTS

Precision Education is a good strategy for this course. The main goal of the course is to learn the precise distinction between what is computable and what is not computable. An goals of the course are to understand the models of computation and complexity of computational problems. As we make progress towards these goals, it would be possible to make precise estimate about what is learnt so far. There are problems that are computationally undecidable (such as integral polynomials with two or more variables) as opposed to comparable computationally decidable problems (such as integral polynomials with one variable). Problem driven teaching will be practiced in this course. Each student is assigned a set of problems at the beginning of the course which must be solved by the due dates (before the end of the course). These problems drive the course. ***Students need to actively participate in the class activities and every student is encouraged to participate in problem solving and classroom discussions.*** Sharing of real world experiences and current events will also be encouraged in the classroom/chats/threaded discussions. Your attendance is very important. Absences, late arrival, or leaving class early will impact your learning and the final grade. You will receive help in your participation activities. However, participation and assignments are the major grading factors. Students are encouraged to discuss problem solving strategies among themselves and learning from each other would be OK. However, plagiarism and copying from each other will NOT be tolerated.

All students are required to meet the minimum requirements for the following items:

- Projects and Assignments
- Quizzes
- Class participation/Discussions
- Midterm examination
- Final examination

COURSE GRADING

Grades will be weighted as follows:

Midterm	20%
Final	40%
Quizzes	15%
Chats/Participation/Discussions	10%
Projects and Assignments	15%

The final course grades are based on the following table:

Letter Grade	GPA	Percentage
A	4.0	95% - 100%
A-	3.7	90% - 94.99%
B+	3.3	87% - 89.99%
B	3.0	84% - 86.99%
B-	2.7	80% - 83.99%
C+	2.3	77% - 79.99%
C	2.0	74% - 76.99%

C-	1.7	70% - 73.99%
D+	1.3	67% - 58.99%
D	1.0	64% - 66.99%
D-	.7	60% - 63.99%
F	0	00% - 59.99%

According to our plan, the following sequence of topics is scheduled for discussions. The plan may be adjusted, if new issues come up.

Session	Readings	Description
1	Chap 1-2-3-4	Course Overview, Introduction to Languages. Finite Automata, Regular Expressions
2	Chap 2-3-4-5-6	Finite Automata, Regular Expressions, Regular Languages, Context-Free Languages.
3	Chap 2-3-4-5-6-7-8	Finite Automata, Regular Expressions, Regular Languages, Context-Free Languages, Pushdown Automata <i>Assignment S1 Due</i>
4	Chap 2-3-4-5-6-7-8	Regular Languages, Finite Automata, Context-Free Languages, Pushdown Automata, Turing Machines <i>Assignments S2, S3, S4, S5 Due</i>
5	Chap 2-3-4-5-6-7-8	Regular Languages, Finite Automata, Context-Free Languages, Pushdown Automata, Turing Machines MIDTERM <i>Assignments S6, S7 Due</i>
6	Chap 2-3-4-5-6-7-8-9	Regular Languages, Finite Automata, Context-Free Languages, Pushdown Automata, Turing Machines, Un-decidability
7	Chap 2-3-4-5-6-7-8-9-10	Regular Languages, Finite Automata, Context-Free Languages, Pushdown Automata, Turing Machines, Un-decidability, Intractable Problems, Database Models. <i>Assignment S8 Due</i>
8	Chap 2-3-4-5-6-7-8-9-10	Regular Languages, Finite Automata, Context-Free Languages, Pushdown Automata, Turing Machines, Undecidability, Intractable Problems, Database Models. <i>Assignment S9 (Project) Due</i>
9	Chap 2-3-4-5-6-7-8-9-10	Final Exam (September 2, 2017)

There will be a quiz in every class except on the days of midterm and final examinations.

The following problems are assigned with due dates:

S₁: Construct a Non-Deterministic Finite Automaton (NFA) for aba^* . Examine if the automaton accepts (1) **abaa**, (2) **baba**, (3) **ab**, (4) **abb**, and (5) **abab** (and answer yes or no for each). In addition, construct a Deterministic Finite Automaton (DFA) for aba^* .

(Due 08-14-2017)

S₂: If L_1 and L_2 are regular languages, then prove that $L_1 \cup L_2$, L_1L_2 , L_1^* are also regular.

(Due 08-16-2017)

S₃: Prove: For every Non-deterministic Finite Automaton there is a regular expression.

(Due 08-16-2017)

S₄: Prove that $L_4 = \{a^n b^n : \text{where } n > 0\}$ is Context-Free. (Due 08-16-2017)

S₅: Construct a Pushdown Automaton or One-Stack Automaton for $L_4 = \{j^n bca^n : \text{where } n \geq 0\}$.

(Due 08-16-2017)

S₆: If L_{61} and L_{62} are sets representing Context-Free languages, then prove that their union, $L_{61} \cup L_{62}$, is also Context-Free. (Due 08-21-2017)

S₇: If L_{71} and L_{72} are sets representing Context-Free languages, then prove that their concatenation, $L_{71}L_{72}$, is also Context-Free. (Due 08-21-2017)

S₈: Construct a Turing Machine for $L_8 = \{ (a^n)^n : \text{where } n > 0 \}$. (Due 08-23-2017)

S₉: (Project) A variant of the following problem based on the last name of the student.

(<http://www.asethome.org/mathfoundations/>)

(Due 08-30-2017)

PART_1. Build the most powerful computing machine that you can think of. Your machine should be able to process complex languages such as $L_{12} = \{c^n dva^n dvj^n : \text{where } n > 0\}$. Demonstrate that a string like ccdvaadvjj would be accepted by the machine. You need to build the machine by defining its elements mathematically. You are not required to deliver the machine with hardware components. If you do not use standard notations provided in the textbook or discussed in the class, then you need to explain your notations. It is known that Finite State Machines or Finite Automata can accept regular expressions. However, Finite Automata cannot process a language like L_{12} , mentioned above, which requires a more powerful machine. You should be able to build such a machine. Explain how your machine will accept strings from L_{12} .

PART_2. You are asked to complete the following three tasks:

1. In the first step, destroy the HALT state(s) or Final states and their incoming transitions of your machine of the assigned problem of PART_1 and examine the consequences.
2. In the second step, destroy the START state and the associated transitions of your machine (in addition to the destructions mentioned in Step 1 and examine the consequences.
3. In this step, reconstruct the machine so that it is distinct from the original machine of PART_1 (may have one or more additional states and/or transitions) and still processes the same language.

(Due 08-30-2017)

Additional help at:

<http://www.asethome.org/mathfoundations/>