UNIVERSITY OF NEBRASKA AT OMAHA
COURSE SYLLABUS/DESCRIPTION

Department and Course Number	CSCI 4660
Course Title	Automata and Formal Languages
Course Coordinator	Hassan Farhat
Total Credits	3
Date of Last Revision	April 22, 2015

1.0 Course Description

1.1 Overview of content and purpose of the course (Catalog description).
This course is designed to provide students with the theoretical aspects of Computer Science in the area of formal languages. In particular, topics covered include: formal languages definitions; deterministic finite automata, non-deterministic finite automata; conversion between automata's; regular languages; closure properties of regular languages; the pumping lemma; context free languages; context free grammars and different forms; conversion between forms of context free grammars; closure properties of context free grammars; pushdown automata; and Turing machines.

1.2 For whom course is intended
The course is intended for upper division undergraduate CS majors

1.3 Prerequisites of the course (Courses)
CSCI 3320, CSCI 3660, MATH 2030 or permission from instructor

1.4 Prerequisites of the course (Topics)
- 1.4.1 Knowledge of Discrete Mathematics topics (sets, counting techniques, functions, etc.)
- 1.4.2 Algorithmic knowledge found in Data Structures
- 1.4.3 Introductory knowledge of theory of computation

1.5 Unusual circumstances of the course
None

2.0 Objectives

2.1 Study the theoretical aspect of computer Science in the context of formal languages.
2.2 Study the different properties of formal languages.
2.3 Relate finite automata and context free languages to computer and programming language design.

3.0 Content and Organization

		Contact hours
3.1	Definition of alphabet and formal languages	1.5
3.2	Deterministic Finite automata (DFA)	6

- Definition

- Construction of automata
- Different forms (Mealy and Moore machines)

3.3 Non-Deterministic Finite Automata (NDFA) 6
- Definition
- Construction of automata
- Conversion from NDFA to DFA

3.4 Regular Languages 5
- Definition
- Regular expressions
- Construction of NDFA from regular expressions
- Construction of a regular expression from NDFA
- Closure properties of regular languages

3.5 The Pumping Lemma for Regular Languages 5

3.6 Context Free Grammars 5
- Definition
- Construction of context free grammars
- Context free languages
- Types of context free grammars (recursive, linear, etc.)
- Normal forms of context free grammars
- Conversion to normal forms

3.7 Pushdown automata 5
- Definition
- Different forms of acceptance
- Conversion from grammar to automata
- Conversion from automata to grammar

3.8 Closure properties of context free languages 3

3.9 The pumping lemma of context free languages 2

3.10 Turing machines 5
- Definition
- Acceptance
- Universal Turing machines and relation to computers
- The halting problem and computable and non-computable problems.

4.0 Teaching Methodology

4.1 Methods to be used.
The primary teaching methods will be lecture, in-class demonstrations, and in-class exercises.

4.2 Student role in the course.
The student will attend lectures and demonstration, participate in discussion on assigned readings, complete assigned homework, and complete required examinations.

4.3 Contact hours.
Three hours per week

5.0 Evaluation

5.1 Type of student projects that will be the basis for evaluating student performance, specifying distinction between undergraduate and graduate, if applicable. For Laboratory projects, specify the number of weeks spent on each project).

Students will complete a sequence of six assignments desired to provide hands-on exposure to the theoretical topics discussed in class, and take two examinations.

5.2 Basis for determining the final grade (Course requirements and grading standards) specifying distinction between undergraduate and graduate, if applicable.

Component	Grading
Exams	80%
Assignments	15%
Participation	5%

5.3 Grading scale and criteria.

Points	*Grade*
97-100%	A+
93-96%	A
90-92%	A-
87-89%	B+
83-86%	B
80-82%	B-
77-79%	C+
73-76%	C
70-72%	C-
67-69%	D+
63-66%	D
60-62%	D-

6.0 Resource Material

6.1 Textbooks and/or other required readings used in course.

6.1.1 Introduction to Automata Theory, Languages, and Computation, 3rd edition, by John E. Hopcroft, Rajeev Motwani, & Jeffrey D. Ullman, Addison Wesley, 2007

6.2 Other suggested reading materials, if any.

6.3 Other sources of information.

6.4 Current bibliography of resource for student's information.

6.4.1 Languages and Machines: An Introduction to the Theory of Computer Science, 3rd edition, by Thomas A. Sudkamp, Addison Wesley, 2005.

6.4.2 Automata, Computability and Complexity: Theory and Applications, by Rich, Prentice Hall, 2008.

6.4.3 An Introduction to Formal Languages and Automata, by Peter Linz, Jones and Bartlett Publishing, 2000.

6.4.4 Languages and Machines: An Introduction to the Theory of Computer Science, 3rd edition, by Sudkamp, Addison Wesley, 2006.
6.4.5 Theory of Computing: A Gentle Introduction, by Efim Kimber & Carl A. Smith, Prentice Hall, 2000.
6.4.6 Theory of Computation: Formal Languages, Automata, and Complexity, by J. Glenn Brookshear, Benjamin Cummings, 1989.
6.4.7 Elements of the Theory of Computation, by Harry R. Lewis & Christos Papadimitriou, Prentice Hall, 1997.
6.4.8 Introduction to the Theory of Computation, by Michael Sipser, Brooks/Cole Publishing Company, 1996.
6.4.9 Logic and Language Models for Computer Science, by Hamburger & Richards, Prentice Hall, 2002.
6.4.10 Introduction to Computability, by Frederick C. Hennie, Addison Wesley, 1977.
6.4.11 Computability and Complexity: From a Programming Perspective, by Neil D. Jones, MIT Press, 1997.
6.4.12 Language and Automata Theory and Applications: 9th International Conference, LATA 2015, Nice, France, March 2-6, 2015.
6.4.13 Elements of Computation Theory, by Arindama Singh, Springer 2014.
6.4.14 Finite Automata, Their Algebras and Grammars: Toward a Theory of Formal Expressions, By Richard Buchi and Dirk Stiefkes, Springer 2012.

1.0 Computing Accreditation Commission Category Content (class time in hours):

CSAB Category	*Core*	*Advanced*
Data Structures		
Computer Organization and Architecture		
Algorithm and Software Design		
Concepts of Programming Languages		42

2.0 Oral and Written Communications

Every student is required to submit at least __0___ written reports (not including exams, tests, quizzes, or commented programs) to typically _____ pages and to make ___0__ oral presentations of typically _____ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

3.0 Social and Ethical Issues

No coverage.

4.0 Theoretical content

The course is very theoretical, the construction of the machines to recognize different languages and the study of closure properties are well founded in the theoretical aspects of Computer Science. An estimated 40 hours of the material presented is theoretical.

5.0 Problem analysis

The course is mathematical in nature and, as a result, requires considerable understanding of the methods used to solve mathematical problems in the course. This involves problem detailed problem reading to understand what is given and what is required. This is followed by analysis applied to special cases for better understanding of the problem requirements.

6.0 Solution design

The solution design includes verification and proof of correctness. In particular proof by contradiction, and inductive proofs are used extensively.

CHANGE HISTORY

Date	*Change*	*By whom*	*Comments*
09/25/2002	Initial ABET version	Farhat	
06/13/2003	Formatting cleanup	Wileman	
12/8/2008	Updated prerequisites and references	Farhat	
12/10/2008	Formed mapping	Farhat	
4/22/2015	Updated References	Farhat	