Capital University of Science and Technology

Department of Computer Science

CS3613 - Theory of Automata and Formal Languages

Course Title: Theory of Automata and Formal Languages (CS3613)

Pre-requisite(s): Discrete Structures (CS2053)

Credit Hours: 3

Instructor(s):

Text Book(s):

Introduction to the Theory of Computation by "Michael Sipser", 3rd Edition, Cengage Learning, 2013.

Introduction to Computer Theory by "Daniel I. A. Cohen", 2nd Edition.

Reference Book(s):

- Introduction to Languages and the Theory of Computation by "John C. Martin", 4th Edition, McGraw Hill, 2011.

Web Reference:

- https://www.tutorialspoint.com/automata_theory/index.htm [Automata Theory Tutorial]

Course Introduction:

This is an introductory course on Theory of Automata. Students are introduced to the concept of Formal Language and Automata. Formal Languages cover recursive definitions of languages, regular grammar, regular expression, context free grammar and language. In Automata they learn about finite automata (deterministic; non-deterministic), transition graphs and pushdown automata (deterministic; non-deterministic). They also learn about fundamental concept of Moore and Mealy machines and Turing machines.

Course Objectives:

At the end of the course, you should be able to:

1. Explain different methods for defining languages

2. Discuss what Finite Automata is
3. Differentiate between Regular Languages and Non-Regular Languages
4. Describe Context-free languages and context-free grammars, parse trees, derivations and ambiguity; Basic concepts of pushdown automata
5. Explain basic definitions and relation to the notion of an algorithm or program

Course Learning Outcomes (CLOs):

At the end of this course, the students should be able to:

CLO1: Describe the role of abstract computational models to define which computational problems are solvable and which are not (C2-Understanding).

CLO2: Interpret formal languages and their description in the form of formal grammars (C3-Application).

CLO3: Construct grammars and models for different languages (C3-Application).

CLO4: Differentiate between deterministic and non-deterministic models and their limitations (C4-Analysis).

CLOs – PLOs Mapping:

	CLO:1	**CLO:2**	**CLO:3**	**CLO:4**
PLO:1 (Academic Education)				
PLO:2 (Knowledge for Solving Computing Problems)	√			
PLO:3 (Problem Analysis)				√
PLO:4 (Design/Development of Solutions)		√	√	
PLO:5 (Modern Tool Usage)				

Course Contents:

Week	Contents
1	Introduction to Theory of Automata, History of Computation, Why this Course and its Application, Basic Definitions i.e. Automata, Letters, Alphabets, Strings and words with examples, Kleene Closure Operator
2	Methods for Defining Languages (Descriptive, Recursive with examples), Regular Expression with Examples and Introduction to finite Automata, States, transitions, input symbols
3	Deterministic Finite Automata with Formal Description and Examples.
4	Non-Deterministic Finite Automata with Formal Description and Examples, Deterministic and Non-deterministic Finite Automata and their differences.
5	NFA DFA Equivalence, Generalized Transition Graphs
6	Examples of GTG and How they relate to Nondeterminism, Kleene's theorem (part I, part II, part III), proof of Kleene's theorem part I
7	Proof of Kleene's theorem part II (method with different steps), particular examples of Transition Graphs to determine corresponding R.E.s. Proof of Kleene's theorem part III (method with different steps), Union of Two Finite Automata
8	Concatenation of Two Finite Automata and Closure of Finite Automata, Pumping Lemma for Regular Languages
Mid-Term Exam	
9	Myhill Nerode Theorem – Regular Languages, Myhill Nerode Theorem – DFA minimization, Mid-term Showing
10	Context Free Languages
11	Context Free Grammars, Derivation Trees and CFG from RE (Regular Expressions) and FA (Finite Automata)

12	Ambiguity, Simplification of CFG, Chomsky Normal Form and Greinbach Normal Form, PDA
13	PDA and its types and Building a PDA, Conversion of CFG to PDA
14	Pumping Lemma for Context Free Languages, Chomsky Hierarchy of languages, Limitations of PDAs
15	Turing machines, Designing Turing machines for complex problems
16	Formal Description of Turing Machine, Designing Turing machines for complex problems

Grading Policy:

S.No	**Grading**	**% of Total Marks**
1	Assignments	20
2	Quizzes	20
3	Mid-term Exam	20
4	Final Exam	40
	Total	**100**