ATME® College of Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING –AI & ML

COURSE MODULE OF THE SUBJECT TAUGT FOR THE SESSION 2023-24 (ODD SEMESTER)

<u>Course Syllabus with CO's</u>

AcademicYear:2023-2024

Department: Computer Science and Engineering-AI & ML

Course Code	Course Title	Core/Elective	Prerequisite	Contact Hours			Total Hrs/ Sessions
				L	T	P	
21CS51	Automata Theory and Compiler Design	Core	-	3	-	-	40

Objectives

1. Introduce the fundamental concepts of Automata Theory, Formal Languages and compiler design
2. Principles Demonstrate Application of Automata Theory and Formal Languages in the field of compiler design
3. Develop understanding of computation through Push Down Automata and Turing Machines
4. Introduce activities carried out in different phases of Phases compiler
5. Identify the undecidability problems.

Topics Covered as Per Syllabus

<u>MODULE-1:</u>

Introduction to Automata Theory: Central Concepts of Automata theory, Deterministic Finite Automata(DFA), Non- Deterministic Finite Automata(NFA) ,Epsilon- NFA, NFA to DFA Conversion, Minimization of DFA Introduction to Compiler Design: Language Processors, Phases of Compilers

<u>MODULE-2:</u>

Regular Expressions and Languages: Regular Expressions, Finite Automata and Regular Expressions, Proving Languages Not to Be Regular Lexical Analysis Phase of compiler Design: Role of Lexical Analyzer, Input Buffering , Specification of Token, Recognition of Token

<u>MODULE-3:</u>

Context Free Grammars: Definition and designing CFGs, Derivations Using a Grammar, Parse Trees, Ambiguity and Elimination of Ambiguity, Elimination of Left Recursion, Left Factoring. Syntax Analysis Phase of Compilers: part-1: Role of Parser , Top-Down Parsing

MODULE-4:

Push Down Automata: Definition of the Pushdown Automata, The Languages of a PDA. Syntax Analysis Phase of Compilers: Part-2: Bottom-up Parsing, Introduction to LR Parsing: SLR, More Powerful LR parsers

MODULE-5:

Introduction to Turing Machine: Problems that Computers Cannot Solve, The Turing machine, problems, Programming Techniques for Turing Machine, Extensions to the Basic Turing Machine Undecidability : A language That Is Not Recursively Enumerable, An Undecidable Problem That Is RE. Other Phases of Compilers: Syntax Directed Translation- Syntax-Directed Definitions, Evaluation Orders for SDD's. Intermediate-Code Generation- Variants of Syntax Trees, Three-Address Code. Code Generation- Issues in the Design of a Code Generator

List of Text Books

1. Textbooks 1. John E Hopcroft, Rajeev Motwani, Jeffrey D. Ullman," Introduction to Automata Theory, Languages and Computation", Third Edition, Pearson.
2. Alfred V.Aho, Monica S.Lam,Ravi Sethi, Jeffrey D. Ullman, " Compilers Principles, Techniques and Tools", Second Edition,Perason.

Reference:

1. Elain Rich, "Automata,Computability and complexity", 1st Edition, Pearson Education,2018.
2. K.L.P Mishra, N Chandrashekaran , 3rd Edition , 'Theory of Computer Science",PHI,2012.
3. Peter Linz, "An introduction to Formal Languages and Automata ", 3rd Edition, Narosa Publishers,1998.
4. K Muneeswaran, "Compiler Design", Oxford University Press 2013

List of URLs,TextBooks,Notes,Multimedia Content,etc

Weblinks and Video Lectures (e-Resources):
1. https://nptel.ac.in/courses/106/106/106106049/#
2. https://nptel.ac.in/courses/106/104/106104123/
3. https://www.jflap.org/

Course Outcomes

At the end of the course the student will be able to:

CO 1. Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation

CO 2. Design and develop lexical analyzers, parsers and code generators

CO 3. Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers.

CO 4. Acquire fundamental understanding of the structure of a Compiler and Apply concepts automata theory and Theory of Computation to design Compilers

CO 5. Design computations models for problems in Automata theory and adaptation of such model in the field of compilers

The Correlation of Course Out comes (CO's) and Program Outcomes(PO's)

Subject Code:	21CS51	**Title:** Automata Theory and Compiler Design										
List of Course Outcomes	**Program Outcomes**											**Total**
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO- 5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	
CO-1	2	2	-	-	-	-	-	-	-	-	-	-	**4**
CO-2	2	2	-	-	-	-	-	-	-	-	-	-	**4**
CO-3	2	2	-	-	-	-	-	-	-	-	-	-	**4**
CO-4	2	2	-	-	-	-	-	-	-	-	-	-	**4**
CO-5	2	2	-	-	-	-	-	-	-	-	-	-	**4**
Total	**10**	**10**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**20**

Note:

3=Strong Contribution 2=Average Contribution 1= Weak Contribution 0=No Contribution

The Correlation of Course Outcomes (CO's)and Program Outcomes(PSO's)

Subject Code:	**21CS51**		**Title:** Automata Theory and Compiler Design
List of Course Outcomes	**Program Specific Outcomes**			**Total**
	PSO-1	**PSO-2**	**PSO-3**	
CO-1	1	-	-	1
CO-2	1	-	-	1
CO-3	1	-	-	1
CO-4	1	-	-	1
CO-5	1	-	-	1
Total	5	-	-	5

Note: 3=Strong Contribution 2=Average Contribution 1 =Weak Contribution 0=No Contribution