Acropolis Technical Campus, Indore, 452020, (M.P.)

Computer Science and Engineering

Course Plan

UG

Theory of Computation			
Course Code	CS501	Session: Jul-Dec 2019	Semester: V
Tutor	Prof. Kunal Batra (Section A) Prof.Jayesh Umre (Section B,C)	Revision date :	Branch: CSE
E mail	Kunalbatra.atc@acropolis.in/ jayeshumre.atc@acropolis.in	Mob. No: 9981588900/9993122561	No. of Pages:10

a. Scheme of the Semester Containing the Course

S. No	Subject Code	Subject Name	Maximum Alloted Marks					Total Marks	Hours/Week		Total Credits
			Theory			Practical					
			End Sem	Mid Sem Exam	Quiz/ Assignment	End Sem	Term Work Lab Work & Sessional		L	P	
1	CS - 501	Theory of Computation	70	20	10	30	20	150	3	2	4

b. Course Overview

Formal languages and automata theory deals with the concepts of automata, formal languages, grammar, computability and decidability. The reasons to study Formal Languages and Automata Theory are Automata Theory provides a simple, elegant view of the complex machine that we call a computer. Automata Theory possesses a high degree of permanence and stability, in contrast with the ever-changing paradigms of the technology, development, and management of computer systems. Further, parts of the Automata theory have direct bearing on practice, such as Automata on circuit design, compiler design, and search algorithms; Formal Languages and Grammars on compiler design; and Complexity on cryptography and optimization problems in manufacturing, business, and management. Last, but not least, research oriented students will make good use of the Automata theory studied in this course.

c. Course Learning Objectives (CLOs)

The Learning Objective of Theory of Computation are such that the student will

CO1 : Study various types of Finite Automata.
CO2: Appreciate to prove equivalence of language described by Automata
CO3: Understand the grammar and PDA for a given language.
CO4: Grasp the comprehensive knowledge of Turing Machine.
CO5 Acquire awareness about the concepts of tractability, decidability, NP completeness.

d. Course Outcomes(COs)

1. To understand computability, decidability, and complexity through problem solving.
2. To analyse and design abstract model of computation & formal languages.
3. To understand and conduct mathematical proofs for computation and algorithms.
4. Discuss abstract model of computing machine through Turing Machine and itstypes.

5. Draw to create modern techniques to solve NP completeness problems.

Course Outcome (CO)	CO Statement
CO.501.1	To understand computability, decidability, and complexity through problem solving.
CO.501.2	To analyse and design abstract model of computation & formal languages,.
CO.501.3	To understand and conduct mathematical proofs for computation and algorithms.
CO.501.4	Discuss abstract model of computing machine through Turing Machine and itstypes.
CO.501.5	Draw to create modern techniques to solve NP completeness problems.

e. **Mapping Course Outcomes (COs) leading to the achievement of Programme Outcomes (POs) and Programme Specific Outcomes (PSOs)**

(A) Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Apply** knowledge of mathematics, science, computing and engineering fundamentals to computer science engineering problems.
2. Able to **identify, formulate,** and demonstrate with excellent programming, and problem solving skills.
3. **Design solutions** for engineering problems including design of experiment and processes to meet desired needs **within reasonable constraints** of manufacturability, sustainability, ecological, intellectual and health and safety considerations.
4. Propose and develop effective **investigational** solution of complex problems using research methodology; including design of experiment, analysis and interpretation of data, and combination of information to provide suitable conclusion. synthesis
5. Ability to create, select and use the **modern techniques** and various **tools** to solve engineering problems and to evaluate solutions with an understanding of the limitations.
6. Ability to acquire knowledge of **contemporary issues** to assess societal, health and safety, legal and cultural issues.
7. Ability to evaluate the **impact** of engineering solutions on individual as well as organization in a societal and environmental context, and recognize sustainable development, and will be aware of emerging technologies and current professional issues.
8. Capability to possess leadership and managerial skills, and understand and commit to professional **ethics** and responsibilities.

9. Ability to demonstrate the team work and **function** effectively as an individual, with an ability to design, develop, test and debug the project, and will be able to work with a multi-disciplinary team.
10. Ability to **communicate effectively** on engineering problems with the community, such as being able to write effective reports and design documentation.
11. Flexibility to feel the recognition of the need for, and have the ability to engage in independent and **life- long learning by** professional development and quality enhancement programs in context of technological change.
12. A **practice** of engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and entrepreneurship.

B) Program Specific Outcomes (PSOs)

1. Develop latest solutions for real world problems; applying mathematical, engineering and project management skills through modern infrastructure and tools to benefit society and human.
2. Understand the need for sustainable development and commit to professional ethics to create an intelligent model that understand real world entities and their relationship to one another.
3. Effectively communicate knowledge, thoughts, techniques and processes to community.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO.501.1	---	---	1	---	---	1	---	2	---	----	1	----	2	1	----
CO.501.2	---	1	2	---	1	---	---	1	---	1	----	1	----	---	----
CO.501.3	1	---	---	1	---	2	--	---	1	----	----	2	----	----	1
CO.501.4	---	1	---	1	---	---	2	---	---	1	---	---	---	---	---
CO.501.5	1	---	---	---	1	---	1	---	1	---	1	---	1	1	---
Average	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Enter correlation level 1, 2, 3 as defined below-

1: Slight (Low); 2: Moderate (Medium); 3: Substantial (High) and if there is no correlation, put "----".

C) Measuring Course Outcomes attained through University Examinations

Course Code	% of students securing C grade (Grade Point 5) or more grade in Theory	Attainment Level for Theory Exams (A1)	% of students securing B grade (Grade Point 7) more grade in Practical	Attainment Level for Practical Exams (A2)	Weighted Attainment level for Theory & Practical (A1+A2)/2
CS504(A)					

Criteria for Attainment:

For Theory Exam:

I. Attainment Level 1: If 60% students scoring ≥C Grade

II. Attainment Level 2: If 61-70% students scoring ≥C Grade

III. Attainment Level 3: If 71-100% students scoring ≥C Grade

For Practical Exam:

IV. Attainment Level 1: If 60% students scoring ≥B Grade

V. Attainment Level 2: If 61-70% students scoring ≥ B Grade

VI. Attainment Level 3: If 71-100% students scoring ≥ B Grade

D) PO and PSO Attainment

urse	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
rect tainment	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
direct tainment	1	2	1	2	1	2	2	1	2	1	2	2	1	2	1
) and PSO tainment	1	2	1	2	1	2	2	1	2	1	2	2	1	2	1

PO Attainment level = 80% of direct assessment + 20% of indirect assessment

f. Topic delivery details of "Content beyond the Syllabus" for the attainment of POs and PSOs.

Director
Shivajirao Kadam Inst. of Tech. & MGMT- Technical Campus
INDORE (M.P.)

Sr. No.	Content Beyond syllabus to be taught	Satisfying PO	Satisfying PSO
1.	Discuss about the compiler	8	1

g. Distribution of Course Work as per University Scheme

Slot / Contact Type	Ingredients (per student)	Distribution of periods @ 60min		Distribution of Marks Max. Marks As per University scheme		
		Number of hours per week	Per Semester (15 weeks)	End Sem.	Internal	
					MST/LWS	Q/A
Theory Slot	Lecture (L)	3	45	70	20	10
	Tutorial (T)	--	--			
Practical Slot	Practical Work (P)	2	26	30	10	10

Internal Assessments are based on scheme provided by the university.

(g.1) No. of Theory Lectures Necessary for the course: 45

(g.2) No. of Theory Lectures Unit wise:

UNIT	I	II	III	IV	V	VI	TOTAL
Assigned No. of Lectures per Unit →	9	9	8	11	8	–	45

h. Prerequisite(s)

The students should have a basic idea of models of computation, computable and non-computable functions, space and time complexity, tractable and intractable functions

i. Post Requisites

The Students able to apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.

Director
Shivajirao Kadam Inst. of Tech. & MGMT- Technical Campus INDORE (M.P.)

j. University Syllabus

Theory

Unit I	**
Introduction of Automata Theory: Examples of automata machines, Finite Automata as a language acceptor and translator, Moore machines and mealy machines, composite machine, Conversion from Mealy to Moore and vice versa	13
Unit II	**
Types of Finite Automata: Non Deterministic Finite Automata (NDFA), Deterministic finite automata machines, conversion of NDFA to DFA, minimization of automata machines, regular expression, Arden's theorem. Meaning of union, intersection, concatenation and closure, 2 way DFA	8
Unit III	**
Grammars: Types of grammar, context sensitive grammar, and context free grammar, regular grammar. Derivation trees, ambiguity in grammar, simplification of context free grammar, conversion of grammar to automata machine and vice versa, Chomsky hierarchy of grammar, killing null and unit productions. Chomsky normal form and Greibach normal form	8
Unit IV	**
Push down Automata: example of PDA, deterministic and non-deterministic PDA, conversion of PDA into context free grammar and vice versa, CFG equivalent to PDA, Petrinet model.	8
Unit V	**
Turing Machine: Techniques for construction. Universal Turing machine Multitape, multihead and multidimensional Turing machine, N-P complete problems. Decidability and Recursively Enumerable Languages, decidability, decidable languages, undecidable languages, Halting problem of Turing machine & the post correspondence problem.	8

****No of lecture required**

Tutorials :- For smooth conduction of tutorials we implements following steps

- Doubt Clearing session.
- Quiz Test.
- Discussed about related topics

Practicals

1. To find a process for function that accept 3 one's
2. To find a process for function that accept the string ending with 101
3. To find a process for function that accept the decimal number divisible by 2
4. To find a process for function that accept string starting with a.
5. To find a process for function that accept string ending with bb.
6. To find a process for function that accept string starting with a and ending with bb.
7. To find a process for function that Convert NDFA to DFA.
8. To find a process for function that produce 'A', 'B', 'C' depends on inputs that end with '10' or with '11' else other
9. To find a process for function that produce 'A', 'B', 'C' depends on inputs that end with '10' or with '11' else other
10. Design a PDA for accepting the language L = {a^n b^n | n>=1},
11. Design a turing machine for accepting the language L = {a^n b^n | n>=1},

Director
Shivajirao Kadam Inst. of Tech.
& MGMT- Technical Campus
INDORE (M.P.)

K. Books prescribed by the University

1. Introduction to Automata Theory Language & Computation, Hopcroft& Ullman, Narosa Publication. Edition & year ✓
2. Element of the Theory Computation, Lewis &Christors, Pearson
3. Theory of Computation, Chandrasekhar & Mishra, PHI
4. Theory of Computation, Wood, Harper & Row
5. Introduction to Computing Theory, Daniel I-A Cohen, Wiley

Additional books prescribed by the Tutor

a. Introduction to Automata Theory, Languages, and Computation by John E. Hopcroft,Rajeev Motwani Jeffrey D. Ullman
b. An Introduction To Automata Theory & Formal Languages by Adesh K. Pandey
c. Theory Of Computer Science: Automata, Languages And Computation by K.L.P. Mishra,Nagasubramaniyan Chandrasekaran

L. List of Lab experiments with COs:

S. No.	Problem Statement	Mapping with CO					
		CO1	CO2	CO3	CO4	CO5	CO6
1	To find a process for function that accept 3 one's	√					
2	To find a process for function that accept the string ending with 101	√					
3	To find a process for function that accept the decimal number divisible by 2	√					√
4	To find a process for function that accept string starting with a			√			
5	To find a process for function that accept string ending with bb.				√		
6	To find a process for function that accept string starting with a and ending with bb.			√	√		
7	To find a process for function that Convert NDFAto DFA			√		√	
8	To find a process for function that produce 'A', 'B', 'C' depends on inputs that end with '10' or with '11' else other		√				√
9	To find a process for function that produce 'A', 'B', 'C' depends on inputs that end with '10' or with '11' else other		√				

10	Design a PDA for accepting the language L = {a^n b^n | n>=1},			√			
11	Design a turing machine for accepting the language L = {a^n b^n | n>=1},		√				

m. Lecture Plan

Acropolis Technical Campus, Indore, (M.P.) Pin- 452020 Computer Science and Engineering **Lecture Plan** UG			
Internet and Web Technology			
Course Code	CS- 501	**Session**: July-Dec2019	**Semester**: V
Tutor	Prof. Kunal Batra(A) Prof.Jayesh Umre (B,C)	**Revision date** :-	**Branch**: CSE
E-Mail	Kunalbatra.atc@acropolis.in / jayeshumre.atc@acropolis.in	**Mob. No**: 9981588900 / 9993122561	**No. of Pages**: 3

Sr. No.	Unit No.	Topic to Cover / Content	Aim (CO)	Ref. no. [page to page].
1	U1	**Automata** • Introduction • Basic Machine • FSM • FSA • Transition Graph • Transition Matrix	CO1	2[1- 3]
2	U1	**Automata** • Define Deterministic Finite State Automata (DFA)	CO1	2[1-3]
3	U1	**Automata** • DFA with Example	CO1	2[4-5]
	U1	**Automata** • Solve problem of DFA	CO1	2[5-6]
	U1	**Automata** • Non-Deterministic Automata (NFA)	CO1	2[7-8]
	U1	**Automata** • Non-Deterministic Automata (NFA)	CO1	2[8-9]
	U2	**Automata** • Equivalence of DFA and NDFA	CO2	2[9-12]
	U2	**Automata** • Transformation of NFA to DFA	CO2	2[10-12]
	U2	**Automata**	CO2	2[12-18]

Director
Shivajirao Kadam Inst. of Tech.
& MGMT- Technical Campus
INDORE (M.P.)

		• Conversion of NFA with e- Transition to NFA without e-Transition		
10	U2	**Automata** • Minimization of DFA's	CO2	2[18-20]
11	U2	**Automata** • Minimization of finite automata	CO2	2[20-21
12	U1	**Automata** • Two-way finite automata	CO2	2[21-25]
13	U2	**Automata** • Mealy & Moore machines	CO2	2[26-29]
14	U2	**Automata** • Mealy & Moore machines	CO2	2[30-34]
15	U3	**Regular Sets and Regular Grammars** • Alphabet, Words, operation, Regular Sets, Finite Automata	CO2	2[222-225]
16	U3	**Regular Sets and Regular Grammars** • Regular Expression,	CO2	2[12-18]
17	U3	**Context Free Grammar** • Introduction to CFG	CO2	2[267-268]
18	U3	• Regular Grammars	CO2	2[47-49]
19	U3	**Context Free Grammar** • Derivation trees and Ambiguity	CO2	2[47-49]
20	U3	**Context Free Grammar** • Derivation trees and Ambiguity	CO2	2[50-53]
21	U3	**Context Free Grammar** • Simplification of Context free grammars	CO2	2[267-268]
22	U3	**Context Free Grammar** • Simplification of Context free grammars	CO2	c, Internet Notes, 2[180-185]
23	U3	**Context Free Grammar** • Simplification of Context free grammars	CO2	2[263]
24	U3	**Context Free Grammar** • Normal Forms	CO2	2[88-87]
25	U3	**Context Free Grammar** • Normal Forms	CO2	2[180-185]
26	U3	**Pushdown Automata** • Definition of PDA	CO3	2[59-66]
27	U4	• Deterministic Pushdown Automata	CO3	2[59-66]
28	U4	**Pushdown Automata** • Non-Dctcrministic Pushdown Automata	CO3	2[51-52]
29	U4	**Pushdown Automata** • PDA corresponding to given CFG	CO3	2[87-88]
30	U4	**Pushdown Automata** • CFG corresponding to a given PDA	CO3	2[263-266]
31	U4	**Pushdown Automata** • CFG corresponding to a given PDA	CO3	2[263-266]
32	U4	**Context Free Languages** • Pumping Lemma for CFL's,	CO3	2[51-52]
33	U5	**Context Free Languages** • Closure Properties of CFL's	CO3	2[608-613]
34	U5	**Context Free Languages** • Decision Problem involving CFL's	CO3	2[608-613]

5	U5	**Turing Machines** • Introduction	CO3	2[750-752], 2[631-632]
6	U5	• TM model	CO3	2[633-635], 2[682-686]
37	U5	**Turing Machines** • Representation and languages acceptability of TM Design of TM	CO3	2[679-689]
38	U5	**Turing Machines** • Universal TM & Other modification	CO3	2[889-893]
39	U5	**Turing Machines** • Properties of recursive & recursively enumerable Languages,	CO3	2[323-327]
40	U5	**Turing Machines** • Church's hypothesis	CO3	2[76-77], c Internet Notes
41	U5	**Turing Machines** • Composite and Iterated TM • Turing Machine as Enumerators	CO3	c, Internet Notes
42	U5	**Tractable and Untractable Problems** • P • NP • NP Complete	CO3	2[702-703],
43	U5	**Tractable and Untractable Problems** • NP Hard Problem • Example of NP Hard Problem	CO3	2[702-703],
44	U5	**Tractable and Untractable Problems** • Satisfy Ability problem • Vertex Cover Problem	CO3	2[705-706]
45	U5	Hamiltonian Path Problem	CO3	2[705-706]

Prepared by:

Prof. Kunal Batra [CS-A]

Prof. Jayesh Umre [CS-B,C]