COURSEPACK

SCHEME

The scheme is an overview of work-integrated learning opportunities and gets students out into the real world. This will give what a course entails.

<table>
<tr><td>Course Title</td><td colspan="3">Theory of Computation</td><td colspan="3">Course Type</td><td colspan="3">Elective</td></tr>
<tr><td>Course Code</td><td colspan="3">E2UC501T</td><td colspan="3">Class</td><td colspan="3">B.Tech(Sem-V)</td></tr>
<tr><td rowspan="6">Instruction delivery</td><td>Activity</td><td>Credits</td><td>Weekly Hours</td><td colspan="4" rowspan="2">Total Number of Classes per Semester</td><td colspan="2" rowspan="2">Assessment in Weightage</td></tr>
<tr><td>Lecture</td><td>3</td><td>3</td></tr>
<tr><td>Tutorial</td><td>0</td><td>0</td><td rowspan="3">Theory</td><td rowspan="3">Tutorial</td><td rowspan="3">Practical</td><td rowspan="3">Self-study</td><td rowspan="3">CIE</td><td rowspan="3">SEE</td></tr>
<tr><td>Practical</td><td>0</td><td>0</td></tr>
<tr><td>Self-study</td><td>0</td><td>6</td></tr>
<tr><td>Total</td><td>3</td><td>3</td><td>45</td><td>0</td><td>0</td><td>90</td><td>50%</td><td>50%</td></tr>
<tr><td>Course Lead</td><td colspan="2">Mr. Mithlesh Kumar Yadav</td><td>Course Coordinator</td><td colspan="6">Mr. Raj Kumar Parida</td></tr>
<tr><td rowspan="2">Names of Course Instructors</td><td colspan="3">Theory</td><td colspan="6">Practical</td></tr>
<tr><td colspan="3">Dr. Pooja
Mr. V. Gokul Rajan
Mr. Mithlesh Yadav
Mr. Raj Kumar Parida
DR. Shruti Sachdeva
Mr. Sheo Kumar
Dr. Anil Gankotiya
Dr. Ragini Kumari
Dr. Arvind Dagur
Dr. Pradeep Bedi
Dr. P Sudhakar
Dr. Tarun Kumar
Mr. Pradeep Chauhan
Dr. Aanjey Mani Tripathi</td><td colspan="6"></td></tr>
</table>

COURSE OVERVIEW

This course emphasizes computability and computational complexity theory. Topics include regular language, context-free languages, decidable and undecidable problems, reducibility, recursive function theory, completeness, hierarchy theorems, inherently complex problems, and interactive proof systems.

PREREQUISITE COURSE

PREREQUISITE COURSE REQUIRED	YES	
If, yes please fill in the Details	**Prerequisite course code**	**Prerequisite course name**
		Discrete Mathematics

COURSE OBJECTIVE

This Course will introduce students to Design automata, regular expressions and context free grammars for accepting or generating a certain language. To classify machines by their power to recognize languages and comprehend the hierarchy of problems.

COURSE OUTCOMES (Cos)

After the completion of the course, the student will be able to:

E2UC501T.1	Prove Mathematical Induction problems.
E2UC501T.2	Construct regular expressions for different languages and create Deterministic Finite Automata (DFA) and Non Deterministic Finite Automata Machine.
E2UC501T.3	Model Push Down Automata (PDA) for Context Free Languages (CFLs).
E2UC501T.4	Design Turing Machine for various recursive enumerable languages.

BLOOM’S LEVEL OF THE COURSE OUTCOMES

CO No.	Remember **KL1**	Understand **KL 2**	Apply **KL 3**	Analyze **KL 4**	Evaluate **KL 5**	Create **KL 6**
E2UC501T.1			√			
E2UC501T.2			√			
E2UC501T.3			√			
E2UC501T.4			√			√

PROGRAM OUTCOMES (POs):

PO1 Computing Science knowledge: Apply the knowledge of mathematics, statistics, computing science and information science fundamentals to the solution of complex computer application problems.

PO2 Problem analysis: Identify, formulate, review research literature, and analyze complex computing science problems reaching substantiated conclusions using first principles of mathematics,natural sciences, and computer sciences.

PO3 Design/development of solutions: Design solutions for complex computing problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4 Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern computing science and IT tools including prediction and modeling to complex computing activities with an understanding of the limitations.

PO6 IT specialist and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional computing science and information science practice.

PO7 Environment and sustainability: Understand the impact of the professional computing sciencesolutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and normsof the computing science practice.

PO9 Individual and team work: Function effectively as an individual, and as a member or leader indiverse teams, and in multidisciplinary settings.

PO10 Communication: Communicate effectively on complex engineering activities with the IT analyst community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 Project management and finance: Demonstrate knowledge and understanding of the computing science and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOME (PSO):

PSO1: Have the ability to work with emerging technologies in computing requisite to Industry 4.0.

PSO2: Demonstrate Engineering Practice learned through industry internship and research project tosolve live problems in various domains.

COs#/ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
E2UC501T.1	2	1	1											
E2UC501T.2	2	2	1											
E2UC501T.3	2	2	2											
E2UC501T.4	2	2	2		1							1		

COURSE ARTICULATION MATRIX

Note: 1-Low, 2-Medium, 3-High

COURSE ASSESSMENT

The course assessment patterns are the assessment tools used both in formative and summative examinations.

Type of Course (T)	CIE			Total Marks		Final Marks CIE*0.5+SEE*0.5
	IA1# (Assignment)	MTE	IA2# (Assignment)	CIE	SEE	
THEORY	25	50	25	100	100	100

*Assignment, Quiz, Class test, SWAYAM/NPTEL/MOOCs and etc.

COURSE CONTENT

THEORY

Content
FINITE AUTOMATA (FA): Introduction, Proof Techniques, Deterministic Finite Automata (DFA) -Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) **REGULAR EXPRESSIONS (RE):** Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions. REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, applications, and Closure properties of regular languages. **CONTEXT FREE GRAMMER (CFG):** Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL (Proof's omitted). **PUSHDOWN AUTOMATA**: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA. TURING MACHINES (TM): Formal definition and behaviour, Languages of a TM, TM as accepters, and TM as a computer of integer functions, Types of TMs. **RECURSIVE AND RECURSIVELY ENUMERABLE LANGUAGES (REL):** Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), un-decidability of PCP.

LESSON PLAN FOR THEORY COURSES (THEORY AND TUTORIAL CLASSES)

FOR THEORY 15 weeks * 3 Hours = 45 Classes (1credit = 1 Lecture Hour)

L-No	Topic for Delivery	Tutorial/ Practical Plan	Skill	Competency
1	Introduction	NA	Write and Understand Mathematical Proof	CO1
2	Background on Sets, Relations and Graphs			
3	Background on different types of mathematical proofs: Deductive, Contradiction, Induction, Contrapositive.			
4	Formal Definition of Deterministic Finite Automata	NA	Formal Definition Of DFA	CO2
5	State transition Diagram			
6	Examples of languages accepted by DFA Using JFlap			
7	NFA formal definition	NA	Construction of State Transition Diagram	
8	Language accepted by NFA Using JFlap			
9	NFA with epsilon transition			
10	Elimination of epsilon transition			
11	Equivalence of DFA and NFA Using JFlap	NA	Minimization of Automata	
12	Minimization of DFA Using JFlap			
13	Minimization of DFA and Myhill Nerode Theorem	NA	Construct equivalent automata with possibly fewer states	
14	FA with output Moore and Mealy machine	NA	Constructing automata with output	
15	Interconversion of Moore and Mealy machine			
16	Regular Expression Introduction	NA	Conversion from regular expression to DFA	
17	Regular Expression Introduction			
18	DFA to Regular Expression			
19	DFA to Regular Expression			
20	Regular Expression to DFA			
21	Regular Expression to DFA			

22	Algebraic Laws for Regular Expressions			
23	Pumping Lemma for Regular Languages			
24	Pumping Lemma for Regular Languages			
25	Closure Properties of Regular Languages			
26	Decision Properties of Regular Languages	NA	Analyze Properties of Regular Languages	
27	Equivalence and Minimization of Automata			
28	Context-Free Grammars Introduction, Leftmost and Rightmost Derivation			
29	Language of a Grammar			
30	Parse Trees	NA	Derivation of CFG	
31	Inference, Derivation and Parse Trees	NA	Derivations and Parse trees for CFG	
32	Ambiguous Grammar and removing ambiguity	NA		
33	Applications of Context Free Language YACC parser	NA	Applications of CFG	
34	Push Down Automata (PDA) Introduction			
35	Language of PDA			
36	DPDA and CFG	NA	Designing PDA	
37	Eliminating epsilon, useless and unit production		Conversion to normal form	
38	Chomsky Normal Form	NA		
39	Pumping Lemma for CFL	NA	Deciding which languages are Not CFL	
40	Closure and Decision Properties of CFL	NA	To formally describe computations by a Turing Machine	CO3

41	Turing Machine		To recognize undecidable languages	
42	Recursive and Recursive enumerable sets			
43	Undecidability			
44	Undecidability	NA		
45	Post's correspondence problem (PCP)	NA	Analysis of complexity of languages	CO4

BIBLIOGRAPHY

Text Book

Introduction to Automata Theory, Languages and Computation 3rd Edition by John E Hopcroft, Rajeev Motwani and Jeffrey D. Ullman [Availaible Online]

Reference Books

Introduction to the Theory of Computation Second Edition by Michael Sipser [Availaible Online]

Introduction to Formal Languages, Automata Theory and Computation by Kamala Krithivasan and Rama R

Journals/Magazines/Govt.Reports/Gazette/Industry Trends

https://theory.report

https://thmatters.wordpress.com/tcs-blogs/

ALGORITHMICA, Springer Publications [Journal]

THEORY OF COMPUTATION, Elsevier [Journal]

Webliography (Two electronic documents or websites that relate to the Course)

https://math.mit.edu/~sipser/18404/

https://www.math.ias.edu/avi/book

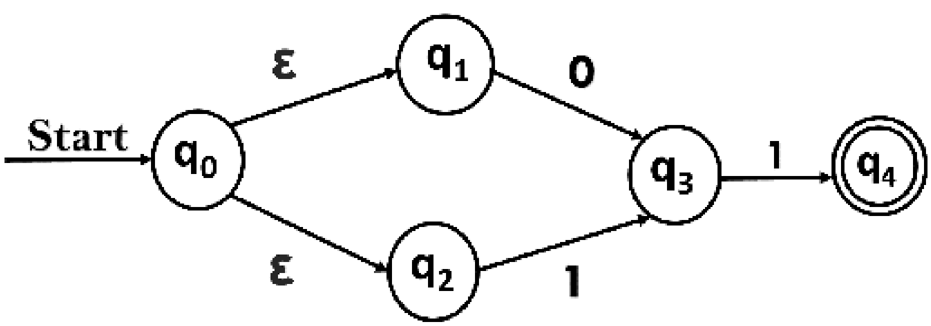
https://cs-people.bu.edu/mbun/courses/332_F21/

SWAYAM/NPTEL/MOOCs Certification (One from Each Platform, Max 3 Platforms)

https://onlinecourses.nptel.ac.in/noc21_cs83/preview

PROBLEM-BASED LEARNING

Exercises in Problem-based Learning (Assignments)

<table>
<tr><th>SNo</th><th>Problem</th></tr>
<tr><td>1</td><td>Prove that the sum of first n natural numbers is equal to n*(n+1)/2</td></tr>
<tr><td>2</td><td>L1={aa, ab, aab, aba} L2={bb, aa, ba, ab, bab} find L1 - L2, L2 – L1.</td></tr>
<tr><td>3</td><td>Find SUFIX and PREFIX of the string “GALGOTIASUNIVERSITY”.</td></tr>
<tr><td>4</td><td>Explain proof by construction, proof by contradiction and proof by induction through examples.</td></tr>
<tr><td>5</td><td>Design DFA for the following language over input alphabet (a,b):
L = String doesn’t start with aab. Using JFlap.</td></tr>
<tr><td>6</td><td>Design DFA for the following language over input alphabet (a,b):
L = Starting with a and end with b. Using JFlap.</td></tr>
<tr><td>7</td><td>Design DFA for the following Language over input alphabet(0,1):
L = Even no. of 0’s or Even no. of 1’s. Using JFlap.</td></tr>
<tr><td>8</td><td>Design Mealy Machine to convert 2’s Complement of the binary input.</td></tr>
<tr><td>9</td><td>How do you remove epsilon transitions from an NFA?</td></tr>
<tr><td>10</td><td>Prove that regular languages is closed under union, intersection.</td></tr>
<tr><td>11</td><td>Design DFA for the following Language over input alphabet(0,1):
L = Starting with 01 and end with 10. Using JFlap.</td></tr>
<tr><td>12</td><td>Design DFA where String does not end with 001. Using JFlap.</td></tr>
<tr><td>12</td><td>Give formal description of Pumping Lemma for Regular Languages.</td></tr>
<tr><td>13</td><td>Convert ϵ-NFA to DFA of the given State Diagram. And Show the Step by Step Process.

</td></tr>
<tr><td>14</td><td>Define – Moore machine. Convert the following Moore machine into its equivalent Mealy machine:
<table>
<tr><td>Present State</td><td>Input 0</td><td>Input 1</td><td>Output</td></tr>
<tr><td>A</td><td>A</td><td>B</td><td>0</td></tr>
</table></td></tr>
</table>

	B	C	B	1
	C	B	C	0
	D	C	C	1

15	Given a CFL. How would you construct a PDA for it?
16	Explain derivation of a CFL sentence from a CFG through example.
17	What are Type 0, Type 1 and Type 2 languages?
18	S –> aSS / aSaS/ aSab | b find left factoring for the given grammar.
19	Apply Pumping Lemma for CFL to show a certain language is not CFL
20	Check whether string W ϵ L(G) or not using membership algorithm. W = baab S –> AB A –> BB | a B –> AB | b
21	What are ambiguous grammar? Explain through examples
22	Convert Context Free Grammar to GNF (Greibach normal form) . S –> CB / AB A –> a / AA B –> b C –> d
23	Write Context Free Grammar for the following languages : i) L={an bn | n>=1} ii) L={am bn | m=2n, n>=0}
24	What are recursive enumerable and recursive language?
25	Prove equivalence between PDA with two stacks and TMs
26	Explain Church Turing Thesis.
27	What are decidable and undecidable languages? Give Examples
28	Design a PDA, a to accept L = { a2n bn | n ≥ 1 }
29	Write Context Free Grammar for the language L={$a_m b_n$ | m=2n, n>=0}.
30	Construct the DPDA Machine for language L={$a_m b_n c_m$ | m, n>=0}.
31	Prove that the following language is ambiguous and convert into unambiguous S → S + S | S * S | a | b Where W = a + a * b.
32	Design a Turing Machine to convert the Binary value to 2's Complement.
33	Recursive Enumerable Languages are Decidable in case of Emptiness, Finiteness and Equivalence. TRUE / FALSE. Justify your answer.

34	Construct the Turing Machine for language L = {W C W | W ϵ (0, 1) *}.
35	Design the Turing Machine for the Language L={$a^n b^n c^n$ | n>=0}
36	Construct the Turing Machine to implement adder for unary value.
37	What is PCP problem? Is it decidable or undecidable?
38	Prove that PCP is undecidable?
39	Differentiate between Decidable and Undecidable problem.
40	Identify the language L= {a ^ x where x is a prime number}.