Course:CS402- Formal Language and Automata Theory

PROGRAMME: **COMPUTERSCIENCE&ENGINEERING**	DEGREE:**B. TECH**
COURSE: Formal Language andAutomata Theory	SEMESTER: **6** CREDITS: **4**
COURSECODE: **CS402**	COURSE TYPE: **Theory**
COURSE AREA/DOMAIN: Theory of Computation	CONTACTHOURS: **4(weekly)**
CORRESPONDINGLABCOURSE CODE (IFANY):---	LABCOURSE NAME:---

Course pre-requisites

CODE	COURSE NAME	DESCRIPTION	SEM
CS201	Basic Computation & Principles of Computer Programming	Programming basics	II
CS302	Data structures andalgorithms	Concept of algorithms	III
CS503	Discrete Mathematics	Elementary discrete mathematics including the notion of set,function,relation,product,partial order,equivalence relation,graph& tree. They should have a thorough understanding of the principle of mathematical induction	V

Course Objectives

1. To develop an understanding of computability and complexity
2. To develop an ability to design machine models for various computation problems

Course Outcomes

1. Students would be able to determine computability of problems
2. Students would be able to design algorithms using machine models
3. Students would be able to apply the knowledge in compiler design, text and image processing.
4. The student will be able to define a system and recognize the behavior of a system. They will be able to minimize a system and compare different systems.

ProgrammeOutcomes addressed in this course

a. An ability to apply knowledge of mathematics, science, and engineering
b. An ability to identify, formulate and solve engineering problems (e)
c. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (k)

Syllabus

Module 1

UNIT	DETAILS	HOURS
I	Fundamentals: Basic definition of sequential circuit, block diagram, mathematical representation, concept of transition table and transition diagram (Relating of Automata concept to sequential circuit concept) Design of sequence detector, Introduction to finite state model	2
II	Finite state machine: Definitions, capability & state equivalent, kth- equivalent concept	1
III	Merger graph, Merger table, Compatibility graph	1
IV	Finite memory definiteness, testing table & testing graph	1
V	Deterministic finite automaton and non deterministic finite automaton	1
VI	Transition diagrams and Language recognizers.	1
VII	Finite Automata: NFA with Î transitions - Significance, acceptance of languages.	1
VIII	Conversions and Equivalence: Equivalence between NFA with and without Î transitions. NFA to DFA conversion	2
IX	Minimization of FSM, Equivalence between two FSM's , Limitations of FSM	1
X	Application of finite automata, Finite Automata with output- Moore & Mealy machine.	2

Module 2

UNIT	DETAILS	HOURS
I	Regular Languages: Regular sets.	1
II	Regular expressions, identity rules. Arden's theorem state and prove	1
III	Constructing finite Automata for a given regular expressions, Regular string accepted by NFA/DFA	1
IV	Pumping lemma of regular sets. Closure properties of regular sets (proofs not required).	1
V	Grammar Formalism: Regular grammars-right linear and left linear grammars.	1
VI	Equivalence between regular linear grammar and FA	1
VII	Inter conversion, Context free grammar	1
VIII	Derivation trees, sentential forms. Right most and leftmost derivation of strings. (Concept only)	1

Module 3

UNIT	DETAILS	HOURS
I	Context Free Grammars, Ambiguity in context free grammars	1
II	Minimization of Context Free Grammars.	1
III	Chomsky normal form and Greibach normal form	1
IV	Pumping Lemma for Context Free Languages	1
V	Enumeration of properties of CFL (proofs omitted). Closure property of CFL, Ogden's lemma & its applications	1
VI	Push Down Automata: Push down automata, definition.	1
VII	Acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence	1
VIII	Equivalence of CFL and PDA, interconversion. (Proofs not required).	1
IX	Introduction to DCFL and DPDA.	1

Module 4

UNIT	DETAILS	HOURS
I	Turing Machine : Turing Machine, definition, model	1
II	Design of TM, Computable functions	1
III	Church's hypothesis, counter machine	1
IV	Types of Turing machines (proofs not required)	1
V	Universal Turing Machine, Halting problem	2

Gaps in the syllabus - to meet industry/profession requirements

S.NO.	DESCRIPTION	PROPOSED ACTIONS	PO MAPPING
1	How to apply machine models in designing programming logic	Lab	b
2	Interconnection between Compiler and Automata Theory	Extra Class	b

Topics beyond syllabus/advanced topics

S.NO.	DESCRIPTION	HOURS
1	Multi Tape Turing Machine and its Applications	1
2	Decidability and Un decidability	1

Web Source References

S.NO.	URL
1	https://class.coursera.org/automata/lecture/preview

Delivery/Instructional Methodologies

S.NO.	DESCRIPTION
1	Chalk and Talk
2	Study Material
3	Power Point Presentation

Assessment Methodologies

S.NO.	DESCRIPTION	TYPE
1	Student Assignment	Direct
2	Tests	Direct
3	University Examination	Direct
4	Student Feedback	Indirect

Course Plan

S. NO.	Day	Module	Topic
1.	Day 1	I	Need for this subject, Decision making as computation
2.	Day 2		Concept of sequential circuit
3.	Day 3		Flip Flop and concept of memory
4.	Day 4	II	Definition of Finite State Automaton
5.	Day 5		Transition table and diagram, Mathematical representation
6.	Day 6		Worked out examples
7.	Day 7	III	Merger graph, Merger table, Compatibility graph
8.	Day 8		Finite memory definiteness, testing table & testing graph
9.	Day 9	IV	Definition of language and grammar
10.	Day 10		Rules – single symbol, concatenation, union, Kleene Closure
11.	Day 11		Closure properties
12.	Day 12		Worked out examples
13.	Day 13	V	Definition, Conversion from NFA to DFA
14.	Day 14		Worked out examples
15.	Day 15	VI	Introduction, Equivalent States, Distinguished State
16.	Day 16		Myhill-Nerode Theorem
17.	Day 17		Table filling method with worked out example
18.	Day 18		Partitioning method with worked out example
19.	Day 19	VII	Arden's Theorem
20.	Day 20		Worked out example
21.	Day 21	VIII	Limitations of Finite State Machines
22.	Day 22		Pumping Lemma for Regular Language
23.	Day 23		Worked out examples

S. NO.	Day	Module	Topic
24	Day 24	IX	Revisiting Formal Grammar, Generalization of grammar with increased power
25	Day 25		Introduction to Context Free Grammar and Context Free Language
26	Day 26		Mathematical Definition of Context Free Language
27	Day 27		Closure properties of Context Free Language
28	Day 28		Normal forms – Chomsky and Greibach Normal Form
29	Day 29		Worked out examples
30	Day 30	X	Limitations of Context Free Language
32	Day 31		Pumping Lemma for Context Free Language
32	Day 32		Worked out examples
33	Day 33	XI	Push Down Automata: Push down automata, definition.
34	Day 34		Acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence
35	Day 35		Equivalence of CFL and PDA, interconversion. (Proofs not required).
36	Day 36		Introduction to DCFL and DPDA.
37	Day 37	XII	Turing Machine : Turing Machine, definition, model
38	Day 38		Design of TM, Computable functions
39	Day 39		Church's hypothesis, counter machine
40	Day 40		Types of Turing machines (proofs not required)
41	Day 41		Universal Turing Machine
42	Day 42		Halting problem

Assignment Set

<table>
<tr><th>S. NO.</th><th>Question</th><th>Assesses CO</th></tr>
<tr><td>1</td><td>Let L be the language with the set of strings over alphabet {0, 1, 2} that do not have two consecutive identical symbols. That is, strings of L are any string in {0,1,2}* such that there is no occurrence of 00, no occurrence of 11, and no occurrence of 22. In the space below, design a DFA (transition table or transition diagram -- your choice) that accepts L.

For each of your states, give a brief description of what strings get you to that state.</td><td>CO 1</td></tr>
<tr><td>2</td><td>Below is the transition table of a DFA.

<table>
<tr><th></th><th>0</th><th>1</th></tr>
<tr><td>->A</td><td>E</td><td>B</td></tr>
<tr><td>* B</td><td>D</td><td>A</td></tr>
<tr><td>C</td><td>G</td><td>A</td></tr>
<tr><td>* D</td><td>G</td><td>E</td></tr>
<tr><td>E</td><td>A</td><td>D</td></tr>
<tr><td>F</td><td>B</td><td>E</td></tr>
<tr><td>* G</td><td>B</td><td>A</td></tr>
</table>

Find the distinguishable states by filling out the table. However, place X's only for those pairs that are distinguishable by the basis step. For those discovered to be distinguishable during the induction, place numbers 1, 2, ... indicating the order in which you discovered these pairs to be distinguishable. Note that many different orders are correct. Also note: cells with # are not to be filled in.</td><td>CO 4</td></tr>
<tr><td>3</td><td>The grammar :

S -> SAS | 0

A -> ASA | 1

(S is the start symbol) is ambiguous. Give an example of a string that has two or more parse trees.</td><td>CO 2</td></tr>
</table>

	Show two of its parse trees.	
4	Let L be the language {ai | i is a perfect square}. That is, L contains the strings a, aaaa, aaaaaaaaa, aaaaaaaaaaaaaaaa, and so on. We want to prove that L is not context free. (Hint: Notice that as strings in L get larger, the difference in length between one string and the next longer string grows without limit.) Suppose n is the pumping-lemma constant for L. What string z do you choose to apply the pumping lemma?	CO 2
5	Write a program to simulate a finite state machine that accepts all floating point numbers. Specify the floating point number format. It should contain Exponents.	CO 3
6	Design a Turing Machine that can multiply two numbers	CO 4
7	Design CFGs for ‘C’ Language	CO3 CO 4
8	Design a PDA for palindrome	CO 4