AUTOMATA THEORY AND COMPUTABILITY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS54	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Introduce core concepts in Automata and Theory of Computation
- Identify different Formal language Classes and their Relationships
- Design Grammars and Recognizers for different formal languages
- Prove or disprove theorems in automata theory using their properties
- Determine the decidability and intractability of Computational problems

Module – 1	**Teaching Hours**
Why study the Theory of Computation, Languages and Strings: Strings, Languages. A Language Hierarchy, Computation, **Finite State Machines (FSM)**: Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, From FSMs to Operational Systems, Simulators for FSMs, Minimizing FSMs, Canonical form of Regular languages, Finite State Transducers, Bidirectional Transducers. **Textbook 1: Ch 1,2, 3,4, 5.1 to 5.10**	**10 Hours**
Module – 2	
Regular Expressions (RE): what is a RE?, Kleene's theorem, Applications of REs, Manipulating and Simplifying REs. Regular Grammars: Definition, Regular Grammars and Regular languages. Regular Languages (RL) and Non-regular Languages: How many RLs, To show that a language is regular, Closure properties of RLs, to show some languages are not RLs. **Textbook 1: Ch 6, 7, 8: 6.1 to 6.4, 7.1, 7.2, 8.1 to 8.4**	**10 Hours**
Module – 3	
Context-Free Grammars(CFG): Introduction to Rewrite Systems and Grammars, CFGs and languages, designing CFGs, simplifying CFGs, proving that a Grammar is correct, Derivation and Parse trees, Ambiguity, Normal Forms. Pushdown Automata (PDA): Definition of non-deterministic PDA, Deterministic and Non-deterministic PDAs, Non-determinism and Halting, alternative equivalent definitions of a PDA, alternatives that are not equivalent to PDA. **Textbook 1: Ch 11, 12: 11.1 to 11.8, 12.1, 12.2, 12,4, 12.5, 12.6**	**10 Hours**
Module – 4	
Context-Free and Non-Context-Free Languages: Where do the Context-Free Languages(CFL) fit, Showing a language is context-free, Pumping theorem for CFL, Important closure properties of CFLs, Deterministic CFLs. Algorithms and Decision Procedures for CFLs: Decidable questions, Un-decidable questions. Turing Machine: Turing machine model, Representation, Language acceptability by TM, design of TM, Techniques for TM construction. **Textbook 1: Ch 13: 13.1 to 13.5, Ch 14: 14.1, 14.2, Textbook 2: Ch 9.1 to 9.6**	**10 Hours**
Module – 5	
Variants of Turing Machines (TM), The model of Linear Bounded automata: Decidability: Definition of an algorithm, decidability, decidable languages,	**10 Hours**

Undecidable languages, halting problem of TM, Post correspondence problem. Complexity: Growth rate of functions, the classes of P and NP, Quantum Computation: quantum computers, Church-Turing thesis. **Textbook 2: Ch 9.7 to 9.8, 10.1 to 10.7, 12.1, 12.2, 12.8, 12.8.1, 12.8.2**	

Course outcomes: The students should be able to:

- Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation
- Learn how to translate between different models of Computation (e.g., Deterministic and Non-deterministic and Software models).
- Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers.
- Develop skills in formal reasoning and reduction of a problem to a formal model, with an emphasis on semantic precision and conciseness.
- Classify a problem with respect to different models of Computation.

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Elaine Rich, Automata, Computability and Complexity, 1st Edition, Pearson Education,2012/2013
2. K L P Mishra, N Chandrasekaran , 3rd Edition, Theory of Computer Science, PhI, 2012.

Reference Books:

1. John E Hopcroft, Rajeev Motwani, Jeffery D Ullman, Introduction to AutomataTheory, Languages, and Computation, 3rd Edition, Pearson Education, 2013
2. Michael Sipser : Introduction to the Theory of Computation, 3rd edition, Cengage learning,2013
3. John C Martin, Introduction to Languages and The Theory of Computation, 3rd Edition, Tata McGraw –Hill Publishing Company Limited, 2013
4. Peter Linz, “An Introduction to Formal Languages an d Automata”, 3rd Edition, Narosa Publishers, 1998
5. Basavaraj S. Anami, Karibasappa K G, Formal Languages and Automata theory, Wiley India, 2012
6. C K Nagpal, Formal Languages and Automata Theory, Oxford University press, 2012.