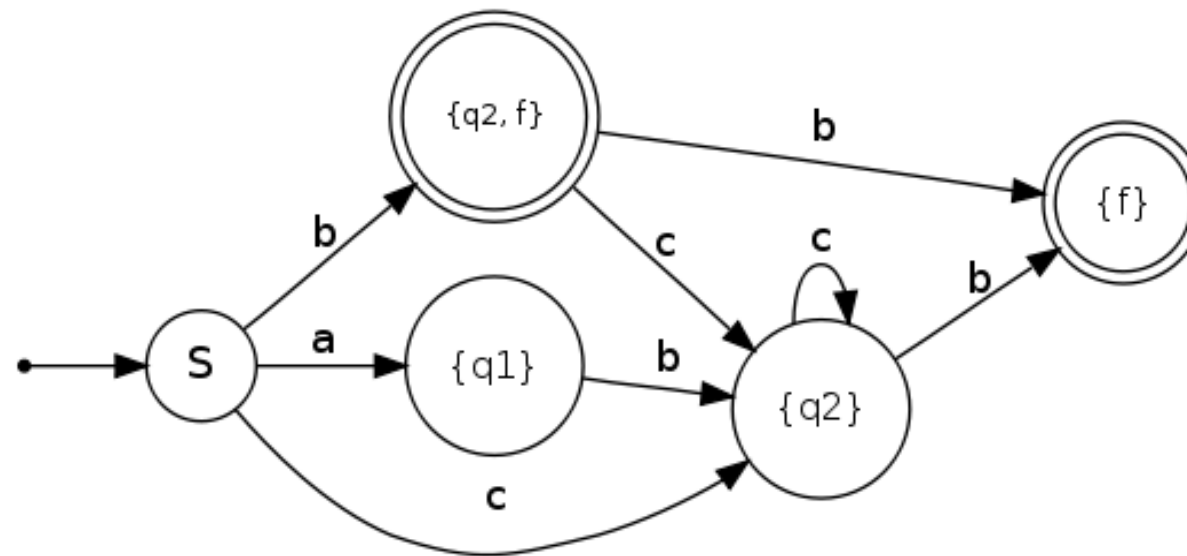


# 15CSE303-Theory of Computation

## Course Overview



**MS.PRATHILOTHAMAI**

**ASSISTANT PROFESSOR (SR.GR)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,  
AMRITA SCHOOL OF ENGINEERING, AMRITA VISHWA VIDYAPEETHAM, COIMBATORE.**

# Course Objectives & Outcomes

## Objectives

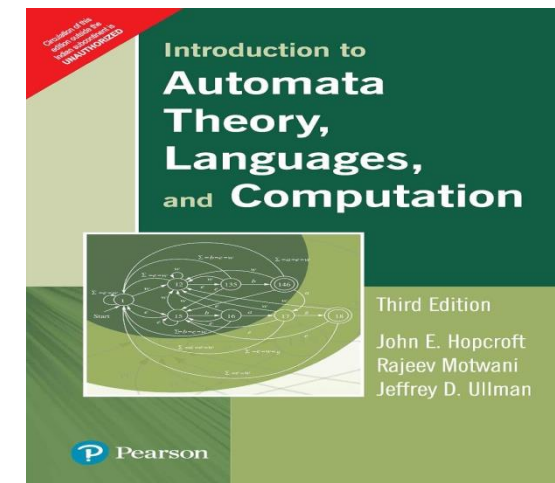
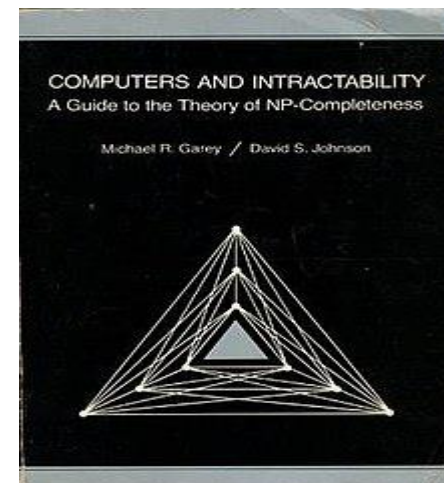
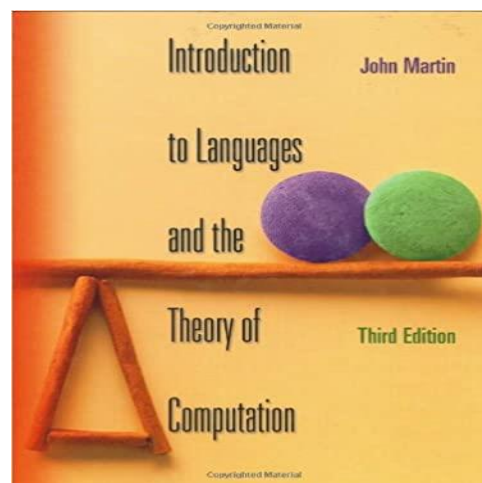
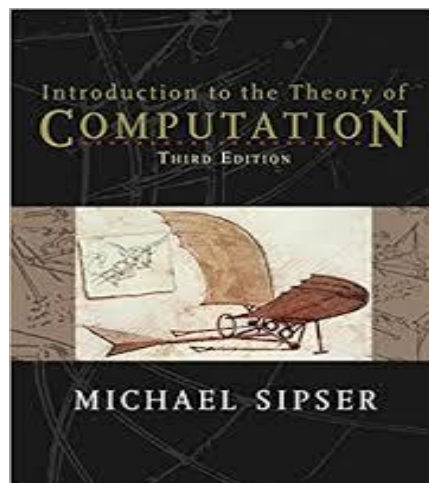
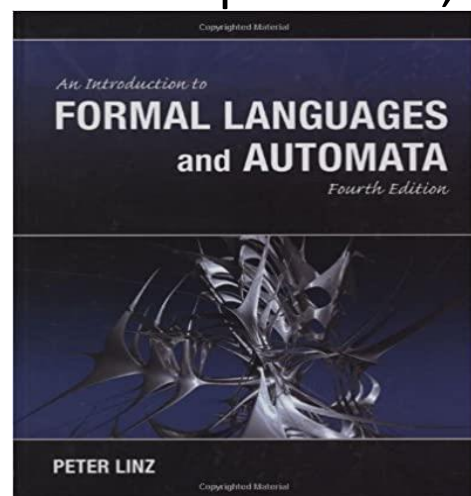
- Introduce concepts in automata theory and theory of computation
- Identify different formal language classes and their relationships
- Design grammars and recognizers for different formal languages
- Design automata and recognizers for different formal languages

## Outcomes

COs	Course Outcome	Bloom's Taxonomy Level	Course Plan
15CSE303.CO 1	Understand and apply the properties of formal languages	L3	<a href="#">Course Plan</a>
15CSE303.CO 2	Illustrate grammar and grammar transformations for formal languages	L3	
15CSE303.CO 3	Construct finite state machines	L3	
15CSE303.CO 4	Apply stack data structures for automata	L3	
15CSE303.CO 5	Design and develop computing devices such as Turing machines	L4	

# Text / Reference Books

- Peter Linz, “An Introduction to Formal Languages and Automata”, Fourth Edition, Narosa Publishing House, 2009 (**Text Book**)
- Michael Sipser, “Introduction to the Theory of Computation”, Third Edition, Cengage Learning, 2012.
- John C Martin, “Introduction to Languages and the Theory of Computation”, McGraw Hill, Third Edition, 2002.
- Michael R Garey and Johnson D S, “Computers and Intractability: A Guide to the Theory of NP-Completeness”, First Edition, W.H. Freeman and Company, 1979.
- J E Hopcroft, R Motwani and J D. Ullman, “Introduction to Automata Theory, Languages, and Computation”, Third Edition, Pearson Education India, 2007.



# Course Pre-Requisites & Organization

## Pre-Requisites

- Data Structure and Algorithms
- Discrete Mathematics

## Course Organization

Generally, the course will contain three parts:

**Part I)** Regular languages (21 Hours)

**Part II)** Context-free languages (15 Hours)

**Part III)** Context-sensitive languages & Turing machines & decidability. (9 Hours)

# Course Assessment Details

S.No	Name of the Assessment	Nos.	Max. Marks	Weightage %	Mode
1	Quizzes	15	20	30	AUMS/AMPLE
2	Tutorial/Assignment	5	20	20	AUMS/AMPLE
3	Case Study(in Groups)	1	20	20	MS Teams
4	EndSem-Online Quiz	1	50	15	AUMS
5	EndSem-Viva	1	15	15	MS Teams
Total				100	

# Faculty Team

S.No	Name of the Faculty	Section	Mail Id
1	Ms.P.Malathi	A	p_malathy@cb.amrita.edu
2	Ms.R.Sujee	B	r_sujee@cb.amrita.edu
3	Ms.T.Bagyammal	C	t_bagyammal@cb.amrita.edu
4	Dr. M. Senthil Kumar	D	m_Senthil@cb.amrita.edu
5	Ms. M.Prathilothamai	E	m_prathilothamai@cb.amrita.edu

**Mentor:** Ms. M. Prathilothamai

# Why Theory of Computation (TOC) is Important?

- To understand the nature of efficient **computation**.
- In theoretical **Computer Science** and **Mathematics**, the **TOC** is the branch that deals with how efficiently problems can be solved on a model of **computation**, using an algorithm.
- To understand and design a Compiler in better manner.
- Student will be able to answer all questions of important exams like GATE, ISRO, DRDO etc.

# TOC: Practical Applications

- **Circuit Design and Verification**

- Finite Automata

- **Natural Language Processing**

- Linguistics (Modelling by grammar)

- **Game Development**

- Finite Automata as strategy models in decision making



# TOC: Practical Applications

## Finite Automata (FA)

- For the designing of lexical analysis of a compiler.
- For recognizing the pattern using regular expressions.
- For the designing of the combination and sequential circuits using Mealy and Moore Machines.
- Used in text editors.
- For the implementation of spell checkers.

## Push Down Automata (PDA)

- For designing the parsing phase of a compiler (Syntax Analysis).
- For implementation of stack applications.
  - Evaluating the arithmetic expressions.
  - Solving the Tower of Hanoi Problem.

# TOC: Practical Applications

## Linear Bounded Automata (LBA)

- For implementation of genetic programming.
- For constructing syntactic parse trees for semantic analysis of the compiler.

## Turing Machine (TM)

- For solving any recursively enumerable problem.
- For understanding complexity theory.

# Interesting Applications & Tools

## Interesting Applications

- Finite State Machines – Traffic Light Simulation  
<https://youtu.be/jLLAtxPNaC0>
- Finite State Machines in Embedded Programming  
<https://youtu.be/0Fqa5RFZ29E>
- Finite State Machines in games: AI  
<https://youtu.be/ayssrKvqJ4g>
- The AI of Half-Life: Finite State Machines  
<https://youtu.be/JyF0oyarz4U>