Gyanmanjari Innovative University

Course Syllabus
Gyanmanjari Institute of Technology
Semester-6 (B.Tech.)

Subject: Theory of Computation - BETCE16325

Type of course: Professional Core

Prerequisite: Calculus, Data Structures and Algorithms, Set Theory

Rationale:

Theory of computation teaches how efficiently problems can be solved on a model of computation. The main thrust is to identify the limitations of the computers through formalizing computation (by introducing several models including Turing Machines) and applying mathematical techniques to the formal models obtained. It is also necessary to learn the ways in which computers can be made to think. Finite state machines can help in natural language processing which is an emerging area.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
CI	T	P	C	Theory Marks		Practical Marks		CA	
				ESE	MSE	V	P	ALA	
4	1	0	5	60	30	10	-	50	150

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; ESE - End Semester Examination; MSE- Mid Semester Examination; V – Viva; CA - Continuous Assessment; ALA- Active Learning Activities.

Course Content:

Sr. No	Course Content	Hrs.	% Weightage
1	**Fundamentals of Formal Languages** Input Alphabet, Strings: Definition, Length, Concatenation, Reversal, Powers, Kleene closure, Substrings, Languages: Definition, Representation, Operations on Languages.	10	15%
2	**Finite Automata & Regular Languages** Regular Expressions, Regular Languages, Application of Finite Automata, Automata with output - Moore machine & Mealy machine, Finite Automata, Non-Deterministic Finite Automata, Conversion from NFA to FA, ε-Non-Deterministic Finite Automata, Conversion of NFA-ε to NFA, Minimization of Finite automata, Regular Expressions Conversion of RE to FA, Arden's theorem, Properties of Regular Languages: Closure properties, Pumping Lemma, Pumping Length, Membership Problem.	15	25%
3	**Context Free Grammar (CFG) & Pushdown Automata (PDA)** Definitions and Examples, BacosNaur Form (BNF), Chomsky Normal Form (CNF). **Pushdown Automata:** Pushdown Automata, CFL And NCFL: Definitions, Deterministic PDA, Equivalence of CFG and PDA & Conversion, Pumping Lemma for CFL.	15	25%
4	**Turing Machines & Recursive Languages** Turing Machine (Definition, Model, Diagram, Transition Table), Model of Computation, Turing Machine as Language Acceptor, TM that Compute Partial Function, Church Turning Thesis, Combining TM, Variations of TM, Non-Deterministic TM, Universal TM, Linear Bounded Automata (LBA), Membership Algorithm, Recursively and Enumerable Languages, Context sensitive languages and Chomsky hierarchy.	15	25%
5	**Undecidability** Definition, Decision Problems: membership, emptiness, finiteness, equivalence, ambiguity, regularity, completeness, disjointness, Undecidable Problems: Halting Problems, Blank Tape Halting Problem, State Entering Problem, Post 's Correspondence Problem (PCP), Modified Post 's Correspondence Problem. Other Theorems: Rice's Theorem.	05	10%

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	**Finite Automata Design and Simulation** Students are required to individually design and simulate DFA and NFA for a given language using a software tool like JFLAP. They will need to illustrate the transition states, final states, and the input alphabet used in their automaton and PDF report that documents the automaton design, explains the transitions, and provides results. This report must be uploaded to the GMIU web portal.	10
2	**Regular Expression to Finite Automaton Conversion** Students will individually convert Regular Expressions (RE) into Finite Automata (FA) by breaking down the RE and applying rules for union, Kleene star, and concatenation. Tools like JFLAP and Automata Tutor will assist in visualizing the conversion. The final FA design, along with a detailed PDF report explaining the process, should be submitted to the GMIU web portal.	10
3	**Designing Pushdown Automata (PDA) from Context-Free Grammar (CFG)** Students will individually design a Pushdown Automaton (PDA) for a given Context-Free Grammar (CFG), demonstrating its ability to process strings from the CFG. They must submit a PDF report detailing the PDA design, states, transitions, and how it processes input. Using a PDA simulation tool like JFLAP, they will verify the PDA's functionality and include screenshots in the report. The final report should be uploaded to the GMIU web portal.	10
4	**Turing Machine Design and Computation** Students will individually design a Turing Machine (TM) for a specific language or problem, including a state diagram, transition table, and step-by-step operation explanation. They will use simulation tools like JFLAP, Turing Machine Simulator (TMS), or Visual Turing Machine Simulator to test and visualize the TM's behavior. The final deliverable is a PDF report detailing the TM's design, computation steps, and the results of the simulation, which should be uploaded to the GMIU web portal.	10
5	**Grammar Simplification and CNF Conversion** Students will individually perform grammar simplification and convert a given Context-Free Grammar (CFG) into Chomsky Normal Form (CNF) or Greibach Normal Form (GNF). The activity involves documenting the execution steps, intermediate results, and the final CNF/GNF grammar. Students will submit a PDF report with detailed explanations of the simplification and conversion steps, which should be uploaded to the GMIU web portal.	10
	Total	50

Suggested Specification table with Marks (Theory): 60

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	25%	20%	20%	15%	10%	10%

Course Outcome:

After learning the course, the students should be able to:	
CO1	Explain the basic concepts of formal languages, alphabets, strings, and language operations.
CO2	Apply finite automata and regular expressions to represent and analyze regular languages.
CO3	Design Context-Free Grammars and equivalent Pushdown Automata for Context-Free Languages.
CO4	Analyze Turing machine models and classify languages using the Chomsky hierarchy.
CO5	Evaluate decidable and undecidable problems to understand the limits of computation.

Instructional Method:

The course delivery method will depend upon the requirement of content and the needs of students. The teacher, in addition to conventional teaching methods by black board, may also use any tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.

Practical examination will be conducted at the end of semester for evaluation of performance of students in the laboratory.

Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory

Reference Books:

[1] Introduction to Languages and the Theory of Computation (4th Edition) by John Martin, Tata McGraw Hill

[2] An Introduction to Automata Theory and Formal Languages by Adesh K. Pandey, Publisher: S.K. Kataria & Sons

[3] Introduction to Computer Theory by Daniel I. Cohen, John Wiley & Sons, Inc

[4] Computation: Finite and Infinite by Marvin L. Minsky, Prentice-Hall

[5] Compiler Design by Alfred V. Aho, Addison-Wesley

[6] Introduction to the Theory of Computation by Michael Sipser

[7] Automata Theory, Languages, and Computation by John Hopcroft, Rajeev Motwani, and Jeffrey Ullman