Course Code: CSE3702	***Course Name:*** *Theory of Computation*	
L-D-P: *3-0-0 (3 Credits) / 16 weeks*	***Contact hours per week:*** *3 hours*	
Course Faculties and Emails:	**Dr. Soharab Hossain Shaikh** soharab.hossain@bmu.edu.in **Dr. Meenakshi Malik** meenakshi.malik@bmu.edu.in	

Aim of the course: The aim of the Theory of Computation and Automata course is to provide students with a deep understanding of foundational concepts in computation theory, including automata, formal languages, and models of computation. By exploring these theoretical frameworks, the course aims to equip students with the analytical and problem-solving skills necessary for designing efficient algorithms and comprehending the limits and possibilities within the realm of computation.

Course Overview and Context: The course introduces some of the fundamental concepts in automata theory and formal languages including grammar, finite automaton, regular expression, formal language, pushdown automaton, and Turing machine. These theoretical foundations not only underpin the comprehension of computational models but also serve as the cornerstone for various branches of computer science, such as compilers, software engineering, and concurrent systems.

Pre-requisite: Mathematics- Sets, Relations, and Functions; Propositions and Predicates

Course Outcomes: At the end of the course, students should be able to

CO1: Apply computational models, namely finite automata, pushdown automata, and Turing machines, to solve problems, showcasing a practical understanding of core concepts in computation theory encompassing automata, formal languages, and models of computation.

CO2: Analyze the limits of computational power by investigating decidability, undecidability, and computational complexity, gaining insights into theoretical constraints and challenges within the realm of computation.

CO3: Design Turing machine-based solutions, employing theoretical foundations in computation theory, to effectively address and solve intricate computational problems.

CO-PO Mapping

Course Outcome	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	POS 4
CO1	3	2	1								1		2			
CO2	3	3	2		3						1		2			
CO3	3	2	3								2		2			

Three Levels of Mapping: 1: means weak, 2: means medium, and 3: means strong.

Note: For details about PO, PSO, and CO-PO mapping, refer to OBE document for B. Tech. – CSE program.

Topics to be covered:

Fundamentals: Strings, Alphabet, Language, Operations, relations to computing

Finite State Machines (FSM): Definition of an automaton - representation of a Finite Automaton - Transition Systems, Properties of Transition Functions, Language - acceptability of a String by a Finite Automaton; Deterministic and Nondeterministic Finite State Machines (DFA and NDFA), Equivalence of DFA and NDFA

Finite Automata with Outputs - Mealy and Moore Models, Transforming a Mealy Machine into a Moore Machine and Moore Machine into a Mealy Machine

Minimization of Finite Automata: Minimizing states of an FSM

Formal Languages: Definition of a Grammar, Derivations, and the Language Generated by a Grammar, Chomsky Classification of Languages - Regular, Context-free, Context-sensitive grammar/languages, Recursive and Recursively Enumerable Sets, Operations on Languages, Languages and Automata

Regular Sets and Regular Grammar: Regular Expressions - Identities for Regular Expressions, Finite Automata and Regular Expressions, Transition System Containing ^-moves NDFAs with ^-moves and Regular Expressions, Conversion of Nondeterministic Systems to Deterministic Systems, Algebraic Method Using Arden's Theorem, Construction of Finite Automata Equivalent to a Regular Expression, Equivalence of Two Finite Automata, Equivalence of Two Regular Expressions, Pumping Lemma for Regular Sets, Application of Pumping Lemma, Closure Properties of Regular Sets, Regular Sets and Regular Grammars, Construction of a Regular Grammar Generating for a Given DFA, Construction of a Transition System M Accepting L(G) for a Given Regular Grammar G

Context-free Languages: Context-free Languages and Derivation Trees, Ambiguity in Context-free Grammars, Simplification of Context-free Grammars, Construction of Reduced Grammars, Elimination of Null productions, Elimination of Unit productions, Normal Forms for Context-free Grammars - Chomsky Normal Form (CNF) and Greibach Normal Form (GNF), Pumping Lemma for Context-free Languages, Decision Algorithms for Context-free Languages

Pushdown Automata (PDA): Definition, Deterministic and Non-deterministic PDA, PDA Corresponding to given CFG and vice-versa.

Turing Machine and Linear Bounded Automata: Turing Machine Model - Representation of Turing Machines -- Instantaneous Descriptions, Transition Table, Transition Diagram; Language Acceptability by Turing Machines, Design of Turing Machines, Description of Turing Machines, Different types of Turing machine; The Model of Linear Bounded Automaton - Relation Between LBA and Context-sensitive Languages

Computability, Decidability and Recursively Enumerable Languages: The Definition of an Algorithm, Decidability, Decidable Languages, Undecidable Languages, Halting Problem of Turing Machine, Theory of complexity – P and NP problems, NP-Completeness

Course Competencies: (Course Outcomes further elaborated) and Instruction Schedule:

Competency	CO	No. of Sessions
Computation, Alphabet, Languages, Computing	CO1	1
Finite State Machine, Deterministic finite automata (DFA), Non-Deterministic Finite Automata (NDFA), NDFA with ε moves, Converting NDFA to DFA	CO1	4
Eliminating ε transitions, Minimization of Finite Automata.	CO1	2
Mealy machine, Moore Machine, Conversion of Mealy machine to Moore machine, and vice-versa.	CO1	2
Regular Expression, Conversion of Regular Expression to Finite Automata, and vice-versa.	CO1	3
Pumping Lemma for regular sets, Application of pumping Lemma, Closure properties of regular sets.	CO1, CO2	2
Introduction to Context Free Grammar (CFG), Derivation Trees, Ambiguity in CFG.	CO1, CO2	3
Simplification of CFG, Normal forms for CFG – CNF, GNF and related topics	CO1, CO2	4
Pumping Lemma for CFG, Closure properties of CFG	CO1, CO2	2
Definition of Pushdown Automata (PDA), Deterministic PDA, Non-deterministic PDA	CO1, CO2	2
PDA Corresponding to given CFG and vice-versa	CO1, CO2	4
Turing Machine model, Representation of Turing machines	CO1, CO2, CO3	3
Design of Turing Machines, Variants of Turing Machine, Turing machine as enumerator	CO2, CO3	3
Properties of recursive and recursively enumerable languages, Decidable and Undecidable problems, Halting	CO1, CO2,	3

problem	CO3

Total Hours for Course: 38

Learning Resources:

Textbooks:

1. K.LP Mishra and N.Chandrasekaran; “Theory of Computer Science”; PHI Learning, 3rd Ed., 2006.
2. Kamala Krithivasan, Rama R.; "Introduction to Formal Languages, Automata Theory and Computation"; Pearson, 1st Ed.; 2009.

References

1. John C. Martin; "Introduction to Languages and The Theory of Computation"; Tata McGraw–Hill Publishing Company Limited, 3rd Ed., 2013.
2. C. K. Nagpal; "Formal Languages and Automata Theory"; Oxford University Press, 2012.
3. Peter Linz; "An Introduction to Formal Languages and Automata", Jones & Bartlett, 6th Ed., 2016.
4. Michael Sipser; “Introduction to the Theory of Computation”; Carnegie Learning, 3rd Ed., 2014.

E-Learning (and other) Resources: Other relevant study material, MOOCs (and other e-learning material) will be suggested and discussed during the lecture session(s) as and when deemed suitable.

Assessment Pattern:

The final grade will be based on the marks/ grades obtained in the End-semester exam along with other assessments defined in the assessment table given below. A relative grading method defined in the academic regulations of the university will be followed to grade the students.

Component	Duration (Min)	Weightage (%)	Evaluation Time	Remarks
Quiz 1 & 2	30 minutes each	2 x 15% = 30%	2nd week of March & 3rd week of April	Closed Book, Online MCQ-based test
Assignment	Students will get approx. a month to work on the solution.	30%	Last 2 weeks of the semester (May 1st & 2nd week)	Take-home assignment based on problems involving the design of a machine and implementation of the same. Open Book, Submission, and Viva-

				based
End Semester Examination	120 minutes	40%	As per the University calendar	Closed Book written examination

Experiential Learning:

- One take-home assignment will be given to the students. The solution to the given problem will require the students to design automata and write suitable grammatical rules to realize the language generated by the automata. It also involves the implementation of the automata by writing a program. It consists of 30% weightage of the whole course.

Student Responsibilities:

- Regular and attentive during lectures/tutorial classes
- Write class notes and submit assignments on time.
- Check announcements at Maitri / Google Classroom / E-mail regularly.
- Regularly check marks and attendance on the Maitri/Google Classroom and inform if any concerns.

Support for Slow Learners:

- Extra classes as per the requirements.
- Additional assignments.
- Group them with advanced learners for project development.

Challenges for Advanced Learners:

- Ask them to solve complex automata design problems.

Attendance Policy: As per the University policy.

Quiz / Assignments/Exams: Two quizzes; one take-home assignment; and one End-semester examination.

Evaluation procedures for tests, and assignments: As per the weightage mentioned above in the assessment pattern.

Late assignment submission policy: Student loses one mark for each late day except for an exceptional situation like a medical problem.

Make-up examination policy: Students who will miss the End-semester examination for unavoidable reasons should inform the course faculty. Recourse examination will be provided only to these students in the End-semester examination component. There will be **no makeup** examination held for any other components.

Behavior expectations: Attend the classes regularly, be punctual, and use of mobile during the lectures and labs is strictly prohibited. Ensure academic integrity – no copying, no cheating, focus on learning and gaining knowledge.

Academic dishonesty/cheating/plagiarism: If any student is found to be violating

expected behavior, then he will be punished according to the University's rules and regulations.