CSEN2041	FORMAL LANGUAGES AND AUTOMATA THEORY	L	T	P	S	J	C
		3	0	0	0	0	3
Pre-requisite	None						
Co-requisite	None						
Preferable exposure	None						

Course Description:

Automata Theory comprised of theoretical computer science and discrete mathematics, is the study of abstract machines for solving computation problems. This course is intended to help the students to gain knowledge in fundamentals of theory of computation that can recognize formal languages typically illustrated by the Chomsky hierarchy. This knowledge can further be applied widely in compiler construction, artificial intelligence.

Course Educational Objectives:

- Impart the mathematical concepts of theoretical computer science from the perspective of formal languages in the design of solving computational machines.
- Familiarize various formal languages, grammar and their relationships.
- Demonstrate various finite state machines and recognize formal languages**.**

UNIT 1 Finite Automata 9 hours

Central concepts of strings, languages and automata theory, Deterministic Finite Automata, Non- Deterministic Finite Automata, Finite Automata with Epsilon Transitions, Finite Automata with Output, Finite Automata Conversions (With Outputs and Without Outputs)

UNIT 2 Regular Expressions and Languages 9 hours

Regular Expressions, Finite Automata and Regular Expressions, Applications of Regular Expressions, Closure Properties of Regular Sets, Pumping Lemma for Regular Languages, Equivalence and Minimization of Finite Automata by partitioning.

UNIT 3 Grammars 9 hours

Context–free grammars; Parse trees; Applications; Ambiguity in grammars and Languages, Simplification of Context Free Grammars, Normal Forms, Pumping Lemma for Context Free Languages, Closure Properties of Context Free Languages.

UNIT 4 Pushdown Automata 9 hours

Definition of the Pushdown Automaton, The Language of Push Down Automaton, Equivalence between Acceptance by Empty Stack and Acceptance by Final State, Equivalence of CFG and PDA, Deterministic Pushdown Automaton, Membership Algorithm (CYK).

UNIT 5 Turing Machines 9 hours

Turing Machine as Acceptor, Turing Machine as a Computing Device, Programming Techniques for Turing Machine, Extensions to the Basic Turing Machine, Linear-Bounded Automata (Introduction), Universal Turing Machines (UTM)

TextBooks:

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory,Languages and Computation, 3/e, Pearson, 2008.
2. Kamala Krithivasan and Rama R, Introduction to Formal Languages, Automata Theory and Computation, Pearson Education, 2009.

References:

1. Peter Linz, An Introduction to Formal Language and Automata, NarosaPub. House, Reprint 2000.
2. John C.Martin, Introduction to Languages and the Theory of Computation,3/e, Tata McGrawHill,2003.
3. 3. Michael Sipser, Introduction to Theory of Computation, 3/e, Wadsworth Publishing Co Inc, 2012

Course Outcomes:

After successful completion of the course the student will be able to:

1. Identify different finite state machines for modelling and solving computational problems
2. Illustrate the concepts in the design of Finite State Machines to recognize Regular Languages
3. Analyse the relation between grammar and language, and design Context Free Grammars for formal languages
4. construct Pushdown Automata for the Context Free Languages and analyse the equivalence between them
5. Design and analyse Turing Machine for Unrestricted Grammar

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	1					1				2	1	1
CO2	3	3	2	2					2				2	2	2
CO3	1	1							1				1	2	2
CO4	1	2	2						2				2	1	
CO5	1	1	2						2				2	2	2

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

APPROVED IN:
BOS : 06-09-2021 **ACADEMIC COUNCIL: 01-04-2022**

SDG No. & Statement: 16 SDG 16: Provides safety and security to the citizens of the county in cyberspace, which creates peaceful and inclusive societies

SDG Justification: