FORMAL LANGUAGES AND AUTOMATA THEORY
(Common to CSE & IT)

Course Code :13CT1115

L	T	P	C
4	1	0	3

Course Educational Objectives:

The course aims to develop an appreciation of the theoretical foundations of computer science through study of mathematical and abstract models of computers and the theory of formal languages.

- ✤ Theory of formal languages and use of various abstract machines as 'recognizers' and parsing will be studied for identifying the synthetic characteristics of programming languages.
- ✤ To understand the fundamental models of computation that underlies modern computer hardware, software, and programming languages.
- ✤ Explain computational thinking
- ✤ Learn the foundations of automata theory, computability theory.
- ✤ Discuss the applications of theory to other areas of computer science such as algorithms, programming languages, compilers, natural language translation, operating systems, and software verification.

Course Outcomes:

At the end of the course the student will be able to

- ✤ Design deterministic and non-deterministic machines.
- ✤ Design the pushdown automata.
- ✤ Comprehend the hierarchy of problems arising in the computer sciences.
- ✤ The Student will get an idea for designing Compiler Design.
- ✤ The students will get knowledge about regular expressions and computability theory .

UNIT-I (14 Lectures)

FUNDAMENTALS & FINITE AUTOMATA:

Basic concepts, Formal languages, Strings, Alphabets, Languages, Finite state machine, definitions, Finite automaton model, Acceptance of strings and languages, Deterministic finite automaton (DFA) and Non-deterministic finite automaton (NFA), Transition diagrams and Language recognizers.Acceptance of languages, Equivalence of NFA and DFA, NFA to DFA conversion(Proof needed), NFA with $\in$ - transitions, Significance, Conversion of NFA with - transitions to NFA without - transitions, Minimization of finite automata, Equivalence between two DFA's, Finite automata with output - Moore and Mealy machines, Equivalence between Moore and Mealy machines, conversion of Moore to Mealy and Mealy to Moore.

UNIT-II (8 Lectures)

REGULAR LANGUAGES:

Regular sets, Regular expressions, Operations and applications of regular expressions, Identity rules, Conversion of a given regular expression into a finite automaton, Conversion of finite automata into a regular expression (Arden's theorem Proof) , Pumping lemma for regular sets (Proof needed), Closure properties of regular sets (proofs not required).

UNIT-III (14 Lectures)

GRAMMAR FORMALISM:

Definition of a grammar, Language of a grammar, Types of grammars, Chomsky classification of languages, Regular grammars, Right linear and left linear grammars, Conversion from left linear to right linear grammars, Equivalence of regular grammar and finite automata, Inter conversion, Context sensitive grammars and languages, Linear bounded automata, Context free grammars and languages, Derivation trees, Leftmost and rightmost derivation of strings and Sentential forms.

CONTEXT FREE GRAMMARS:

Ambiguity, left recursion and left factoring in context free grammars, Minimization of context free grammars, Normal forms for context free grammars, Chomsky normal form, Greibach normal form, Pumping lemma for context free languages(Proof), Closure and decision properties of context free languages(Proofs needed), Applications of context free languages. (Proofs omitted).

UNIT-IV (12 Lectures)

PUSHDOWN AUTOMATA:

Pushdown automata, definition, model, Graphical notation, Instantaneous descriptions, Acceptance of context free languages, Acceptance by final state and acceptance by empty state and its equivalence, Equivalence of context free grammars and pushdown automata, Inter-conversion(Proofs not required), Introduction to deterministic pushdown automata.

TURING MACHINE:

Turing Machine, definition, model, Instantaneous descriptions, Representation of Turing machines, Design of Turing machines, Types of Turing machines, Computable functions, Unrestricted grammar, Recursive and recursively enumerable languages and Church's hypothesis. (Proofs required)

UNIT-V (8 Lectures)

COMPUTABILITY THEORY:

LR(0) grammar, Decidable and un-decidable problems, Universal Turing machine, Halting problem of a Turing machine, Un-decidability of post's correspondence problem(Proof needed) and modified post's correspondence problem, Turing reducibility, Definition of classes P and NP problems, NP complete and NP hard problems.

TEXT BOOKS:

1. Hopcroft H.E. and Ullman J. D, *"Introduction to Automata Theory Languages and Computation"*, 3rd Edition, Pearson Education, 2011.

REFERENCES:

1. Daniel I.A. Cohen, *"Introduction to Computer Theory"*, 2nd Edition, John Wiley Publication, 2007
2. Mishra and Chandrashekaran, *"Theory of Computer Science –Automata Languages and Computation"*, 3rd Edition, PHI, 2009.

3. John C Martin, *"Introduction to languages and the Theory of Computation"*, 3rd Edition, TMH, 2010.
4. Michel Sipser, *"Introduction to Theory of Computation"*, 2nd Edition, Thomson, 2012.
5. J.E.Hopcraft and JefferyD.Ulman,S.N.Maheswari, *"Introduction to Automata Theory, Languages & Computation"*, 2nd Edition, Narosa publishing company, 2011.
6. K.V.N.Sunitha , N.Kalyani, *"Formal Languages and Automata Theory"*, 1st Edition, TMH, 2010.
7. Rajendra Kumar, *"Theory of Automata, Languages & Computations"*, 1st Edition, TMH, 2010.

WEB REFERENECES:

http://nptel.iitm.ac.in/courses/Webcourse-contents/IIT-%20Guwahati/afl/index.htm