THEORY OF COMPUTATION
ENCT 203

Lecture : 3
Tutorial : 1
Practical : 0

Year : II
Part : I

Course Objectives:
The objective of this course is to introduce students to the foundational concepts of theory of automata, formal languages, computational models and computational complexity.

1 Introduction to Formal Language, Logic and Proof (7 hours)

1.1 Brief review of set theory, function and relation
1.2 Propositional logic, expressing statements in propositional logic, rules of inference and proofs in propositional logic, introduction to predicate logic
1.3 Proofs, principle of mathematical induction, diagonalization principle, pigeonhole principle
1.4 Alphabet and language
1.5 Operations on languages: Union, concatenation, Kleene star

2 Finite Automata and Regular Language (10 hours)

2.1 Introduction to finite automata, finite state machine
2.2 Deterministic finite automata (DFA), representation of DFA, language of DFA, design of DFA
2.3 Non deterministic finite automata (NFA), equivalence of DFA and NFA
2.4 Finite automata with epsilon transition (ε - NFA), equivalence of NFA and ε –NFA, equivalence of DFA and ε – NFA
2.5 Regular expressions and regular languages
2.6 Equivalence of regular expression and finite automata
2.7 Closure properties of regular languages
2.8 Pumping lemma for regular languages
2.9 Decision algorithm for regular language

3 Context Free Grammar and Pushdown Automata (10 hours)

3.1 Introduction to context free grammar (CFG), component of CFG, context free language (CFL)
3.2 Types of derivations, parse tree and its construction, ambiguity
3.3 Simplification of CFG, normal forms, Chomsky normal form (CNF), Greibach normal form (GNF), Backus-Naur form (BNF)
3.4 Closure properties of context free languages

3.5 Pumping Lemma for context free languages
3.6 Decision algorithm for context free language
3.7 Introduction to push down automata (PDA), representation of PDA, operations of PDA, move of a PDA, instantaneous description for PDA
3.8 Language of PDA, equivalence of CFL and PDA, conversion of CFG to PDA
3.9 Context sensitive grammar

4 Turing Machine (10 hours)

4.1 Introduction to turing machine (TM), representation of TM, move of a TM, instantaneous description for TM
4.2 Computing with turing machine
4.3 Variants of turing machine
4.4 Unrestricted grammar, Chomsky hierarchy of grammar
4.5 Recursive function theory

5 Decidability and Computational Complexity (5 hours)

5.1 Church turing thesis
5.2 Universal turing machine, encoding of turing machine
5.3 Undecidable problem about turing machines, halting problems and its implications
5.4 Computational complexity, time and space complexity of a turing machine
5.5 Complexity classes class P, class NP, NP-complete problems

6 Automata Theory and Compiler (3 hours)

6.1 Basic concept of compiler, role of lexical analyzer, lexical analysis with deterministic finite automata
6.2 Parser and context free grammar, top down parsing, bottom up parsing, IR parsing

Tutorial (15 hours)

1. Set operations and proof using mathematical induction
2. Proof using rules of inference in propositional logic
3. Design of DFA, conversion of NFA to DFA, proof using pumping lemma for regular language
4. Writing grammar for context free language, design of PDA, proof using pumping Lemma for context free language
5. Design of turing machine for a language
6. Problem related to compiler design

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Marks Distribution*
1	7	9
2	10	13
3	10	13
4	10	14
5	5	6
6	3	5
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Lewis, H. R., Papadimitriou, C. H. (1981). Elements of the Theory of Computation. United Kingdom: Prentice-Hall.
2. Sipser, M. (2006). Introduction to the Theory of Computation. United Kingdom: Thomson Course Technology.
3. Rosen, K. (2006). Discrete Mathematics and Its Applications. United Kingdom: McGraw-Hill Education.
4. Aho, A. V. (2003). Compilers: Principles, Techniques and Tools (for VTU). India: Pearson.