FORMAL LANGUAGES AND AUTOMATA THEORY

Program: Second Year B.Tech. in Computer Engineering

Semester : IV

Course Code: DJ19CEC402

FLOW OF CONTENTS

- Introduction
- Course Outcomes
- Syllabus
- Books Recommended
- List of Tutorials / Experiments

ABOUT ME...

Sridhar C Iyer

Ph.D (Pursuing), M.E, B.E, Certified Ethical Hacker (CEH)

Assistant Professor
Dept. of Computer Engineering
Dwarkadas J Sanghvi College of Engineering

Experience: ~~9~~ 10 Years

Authored ~~2~~ 7 Books on Cyber Security and Digital Forensics.

https://www.youtube.com/c/csridhar89

https://www.linkedin.com/in/sridhar-chandramohan

https://sridhariyer.learnityourway.com/

https://www.facebook.com/sridhar.chandramohan

COURSE OUTCOMES

On completion of the course, the learner will be able to:

- Understand basic concepts in automata theory and theory of computation.
- Identify different formal language classes and their relationships.
- Design grammars and recognizers for different formal languages.
- Prove or disprove theorems in automata theory using its properties.
- Determine the decidability and intractability of computational problems.

SYLLABUS

Module 1: Fundamentals (5 Hours)

Strings, Alphabet, Language, Operations, Finite state machine, definitions, finite automaton model, acceptance of strings, and languages, deterministic finite automaton and non deterministic finite automaton, transition diagrams and Language recognizers

SYLLABUS

Module 2: Finite Automata (6 Hours)

NFA with ε transitions - Significance, acceptance of languages. Conversions and Equivalence: Equivalence between NFA with and without ε transitions, NFA to DFA conversion, minimisation of FSM, equivalence between two FSM's, Finite Automata with output- Moore and Mealy machines.

Applications: –

For the designing of lexical analysis phase of a compiler.

SYLLABUS

Module 3: Regular Languages (3 Hours)

Regular sets, regular expressions, identity rules, Constructing finite Automata for a given regular expressions, Conversion of Finite Automata to Regular expressions. Pumping lemma of regular sets, closure properties of regular sets (proofs not required).

SYLLABUS

Module 4: Grammars (12 Hours)

Regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and FA, inter conversion, Context free grammar, derivation trees, and sentential forms. Right most and leftmost derivation of strings.

Context Free Grammars:

Ambiguity in context free grammars. Minimisation of Context Free Grammars. Chomsky normal form, Greibach normal form, Pumping Lemma for Context Free Languages. Enumeration of properties of CFL (proofs omitted), Chomsky hierarchy of languages, linear bounded automata and context sensitive language, LR(0) grammar.

SYLLABUS

Module 5: Push Down Automata (8 Hours)

Push down automata, definition, model, acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. Equivalence of CFL and PDA, interconversion. (Proofs not required). Introduction to DCFL and DPDA.

Applications:

- For designing the parsing phase of a compiler (Syntax Analysis).
- For evaluating the arithmetic expressions

SYLLABUS

Module 6: Turing Machines (8 Hours)

Turing Machine, definition, model, design of TM, Computable functions, recursively enumerable languages. Church's hypothesis, counter machine, types of Turing machines, Universal Turing Machine, Halting Problem.

BOOKS RECOMMENDED

Text books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, " Introduction to Automata Theory, Languages and Computation", Pearson Education.

2. J.C.Martin, "Introduction to languages and the Theory of Computation", TMH.

3. Michael Sipser, "Theory of Computation",Cengage Learning.

Reference Books:

1. O.G.Kakde, "Theory of Computation", LP.

2. Krishnamurthy E.V., "Introductory Theory of Computer Science", East-West press.

LIST OF TUTORIALS/EXPERIMENTS:

- 1. Finite state machine and NFA with and without epsilon.
- 2. NFA to DFA, DFA minimization (Myhill-Nerode theorem), Moore and Mealy machines
- 3. Regular expressions, Arden’s theorem
- 4. Derivation, Parse tree, ambiguity, Right and left linear grammar
- 5. CNF and GNF
- 6. Push down automata
- 7. Pumping lemma: RL and CFL,CFG to PDA
- 8. Turing Machine
- 9. Implement any 1 application of finite automata
- 10. Implement any 1 application of push down automata