|  |
| --- |
| **CS21003: Automata Theory and Formal Languages**  **School of Computer Engineering,**  **KIIT Deemed to be University, Bhubaneswar**  **Session: Aug-Dec, 2023**  **Credit: 4** |

**Course Objective:**

|  |
| --- |
| * To know about Chomsky hierarchy for organizing languages |
| * To introduce concepts in automata theory and theory of computation |
| * To identify different formal language classes and their relationships |
| * To design grammars and recognizers for different formal languages |
| * To understand undecidability and decide on languages that are undecidable |

**Course Outcome:** At the end of the course, the students will be able to:

|  |  |
| --- | --- |
| CO1: | Acquire a fundamental understanding of the core concepts in automata theory and formal languages |
| CO2: | Design finite automata or regular expression for any tokenization task |
| CO3: | Construct a context free grammar for parsing any language |
| CO4: | Design Turing machine for any language |
| CO5: | Conclude the decidable / undecidable nature of any language |
| CO6: | Apply mathematical and formal techniques for solving real-world problems |

### **Mapping of Course Outcomes with Programme Outcomes**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PO** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **PSO1** | **PSO2** | **PSO3** |
| CO1 | √ |  | √ |  |  |  |  |  |  |  |  |  | √ |  |  |
| CO2 | √ |  | √ |  |  |  |  |  |  |  | √ | √ | √ |  |  |
| CO3 |  |  | √ | √ |  |  |  |  |  |  |  |  |  |  | √ |
| CO4 |  | √ |  | √ |  |  |  |  |  |  | √ |  |  | √ | √ |
| CO5 |  |  | √ | √ |  |  |  |  |  |  |  | √ | √ |  | √ |
| CO6 |  | √ | √ | √ |  |  |  |  |  |  |  |  | √ |  | √ |

### **Course Contents**

**UNIT I**

**Finite Automata:**

Alphabets, Strings and Languages, Automata and Grammars, Deterministic Finite Automata (DFA), Formal Definition, Simplified notation: State transition graph, Transition table. Language of DFA, Non-deterministic Finite Automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata, Distinguishing one string from other, Myhill Nerode Theorem.

### **UNIT II**

### **Regular Expression (RE):**

Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleene’s Theorem, Regular expression to FA, DFA to Regular expression, Arden Theorem, Non Regular Languages, Pumping Lemma for regular Languages. Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, FA with output: Moore and Mealy machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.

### **UNIT III**

### **Context Free Grammar (CFG) and Context Free Languages:**

Definition, Examples, Derivation, Derivation trees, Ambiguity in Grammar, Inherent ambiguity, Ambiguous to Unambiguous CFG, Useless symbols, Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure properties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping lemma for CFLs.

### **UNIT IV**

### **Push Down Automata (PDA):**

Description and definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of acceptance by empty stack and final state, Conversion of CFG to PDA and PDA to CFG.

### **UNIT V**

### **Turing Machines (TM) and Undecidability:**

Basic model, definition and representation, Instantaneous Description, Language acceptance by TM, Variants of Turing Machine, TM as Computer of Integer functions, Universal TM, Church’s Thesis, Recursive and recursively enumerable languages, Halting problem, Introduction to Undecidability, Undecidable problems about TMs, Post correspondence problem (PCP), Modified PCP and undecidable nature of post correspondence problem, Introduction to recursive function theory.

**Text books:**

1. Introduction to automata theory, languages and computations, John E.Hopcroft, Jeffery D.Ullman, Pearson Education, 3rd Edition. ([PDF](https://drive.google.com/file/d/1r9uSSlZR6Vmy_7d5euMenGe7HmMagtmL/view?usp=sharing))

**Reference Books:**

1. An Introduction to Formal Language and Automata, Peter Linz, Jones & Bartlett Publishers, 6th Edition. ([PDF](https://drive.google.com/file/d/19wuWAkdxpbW9VlRhG21YS7vVuHQMuV8z/view?usp=sharing))
2. Elements of the theory of computation, Lewis, Harry R. and Christos H. Papadimitriou Prentice- Hall Englewood, 2nd Edition. ([PDF](https://drive.google.com/file/d/1ji2UXIKEXJy-ZbyLJAqroLamiNND2lY7/view?usp=sharing))
3. Introduction to the Theory of Computation, Michel Sipser, Thomson Brooks/Cole, 2nd Edition. ([PDF](https://drive.google.com/file/d/1n88zmFgNc3xI3-EZbjNOkSFNtHSwtjXV/view?usp=sharing))
4. Theory of computer science by KLP Mishra & N. Chandra Sekharan ,PHI, 3rd edition. ([PDF](https://drive.google.com/file/d/1CajsOUeTqZOWJNj1PC85ruBoDEen6tFL/view?usp=sharing))