|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subject Name:** | | | | | | | Automata Theory and Compiler Construction | | | | | | | | | | | | | | | | | | | **Subject Code:** | | | | | | | | TMC 303 | | |
|  | | | | | | |  | | | | | | | | | | | | | | | | | | |  | | | | | | | |  | | |
| **Course Name:** | | | | | | | Master of Computer Applications (MCA) | | | | | | | | | | | | | | | | | | |  | | | | | | | |  | | |
|  | | | | | | |  | | | | | | | | | | | | | | | | |  | | | | | | | | |  | | | |
| **1** | **Contact Hours:** | | | | | | | | | 45 | | |  | | | | | | | | | | | | | | **L** | | 3 | | | **T** | | 0 | **P** | 0 |
|  |  | | | | | | | | |  | | |  | | | | | | | | | | | | | |  | |  | | |  | |  |  |  |
| **2** | **Examination Duration (Hrs):** | | | | | | | | | | | | | | |  | **Theory** | | | 0 | 3 |  | **Practical** | | | | | 0 | | 0 | |  | | | | |
|  |  | | | | | | | | | | | | | | |  |  | | |  |  |  |  | | | | |  | |  | |  | | | | |
| **3** | **Relative Weightage:** | | | | | | | | | |  | | | | **CWE:** | | | | 25 | **MTE:** | | | 25 | | **ETE:** | | | | 50 | | | |  | | | |
|  |  | | | | | | | | | |  | | | |  | | |  | |  | | |  | |  | | | |  | | | |  | | | |
| **4** | **Credits:** | | | | | 0 | | 3 | |  | | | | | | | |  | |  | | |  | |  | | | |  | | | |  | | | |
|  |  | | | | |  | |  | |  | | | | | | | |  | |  | | |  | |  | | | |  | | | |  | | | |
| 5 | **Pre-Requisite:** | | | | | | | | | Elementary knowledge of Discrete Mathematics. | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | **Subject Area:** | | | | | | | | | Computer Science. | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | **Objective:** | | | | | | | | To teach how the various machines work. | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | **Course Outcome:** | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | |
|  | A student who successfully fulfills the course requirements will be able to: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | **CO 1** | | | | Classify machines by their power to recognize languages, | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | **CO 2** | | | | Employ finite state machines to solve problems in computing, | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | **CO 3** | | | | Analyze and implement different parsing algorithms, | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | **CO 4** | | | | Analyze and implement different intermediate code generation techniques | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | **CO 5** | | | | Use different tools to analyze and implement lexical and syntax analyzer | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | **CO 6** | | | | Analyze and implement code optimization techniques. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | **Details of the Course:** | | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | |
| **Unit No.** | | | | **CONTENT** | | | | | | | | | | | | | | | | | | | | | | | | | | | **CONTACT HOURS** | | | | | |
| **1** | | | | **Finite Automata (FA)**: Nondeterministic finite Automata (NFA), Deterministic finite Automata (DFA), Construction of DFA from NFA, Minimization of DFA, Myhill-Nerode theorem, Moore machine, Mealy machine, Applications and Limitation of FA. | | | | | | | | | | | | | | | | | | | | | | | | | | | 10 | | | | | |
| **2** | | | | **Introduction to languages:** Chomsky hierarchy, Regular Grammar, Regular expressions, Arden Theorem, Pumping Lemma for regular language. **Context free grammar:** Ambiguity, Simplification of CFGs, Ambiguous to Unambiguous CFG. Introduction to Push Down Automata (PDA), Acceptance of a string by PDA. | | | | | | | | | | | | | | | | | | | | | | | | | | | 10 | | | | | |
| **3** | | | | Structure of a compiler, Lexical Analysis, Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, Symbol Table, Lex. Role of Parser, Top Down Parsing, Recursive Descent Parser, Predictive Parser: LL(1) Parser, Shift Reduce Parser, LR Parser, LR (0)Item, Construction of SLR Parsing Table, Introduction to LALR Parser, Error Handling and Recovery in Syntax Analyzer, YACC. | | | | | | | | | | | | | | | | | | | | | | | | | | | 13 | | | | | |
| **4** | | | | Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking. | | | | | | | | | | | | | | | | | | | | | | | | | | | 6 | | | | | |
| **5** | | | | **Code Generation and Optimization:**  Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management, Issues in Code Generation, Design of a simple Code Generator, Introduction to Code Optimization. | | | | | | | | | | | | | | | | | | | | | | | | | | | 6 | | | | | |
|  | | | | **TOTAL** | | | | | | | | | | | | | | | | | | | | | | | | | | | **45** | | | | | |
|  | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | |
|  | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | |
| **10** | | **Suggested Books:** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | |
| **Sl. NO.** | | | **NAME OF AUTHERS/BOOKS/PUBLISHERS** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | **YEAR OF PUBLICATION** | | | | |
| **1** | | | J. E. Hopcroft, J. D. Ullman and R. Motwani: Introduction to Automata Theory, Languages and Computation, Addison-Wesley, California, 2001. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 2001 | | | | |
| **2** | | | H. R. Lewis and C. H. Papadimitriou: Elements of The Theory of Computation, Prentice Hall, Englewood. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 1981 | | | | |
| **3** | | | Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, Pearson | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 1986 | | | | |
| **4** | | | Allen I. Holub, Compiler Design in C, Prentice Hall Software Series | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 1990 | | | | |
| **5** | | | Keith D. Cooper, Linda Torczon, Engineering a Compiler, Morgan Kaufmann | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 2003 | | | | |