राष्ट्रीय प्रौद्योगिकी संस्थान पटना / NATIONAL INSTITUE OF TECHNOLOGY PATNA
संगणक विज्ञान एंव अभियांत्रिकी विभाग / DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
अशोक राजपथ, पटना-८००००५, विहार / ASHOK RAJPATH, PATNA-800005, BIHAR
Phone No.: 0612-2372715, 2370419, 2370843, 2371929 Ext- 200, 202 Fax-0612-2670631 Website: www.nitp.ac.in

No:- **Date:**

CS44114 Automata Theory and compiler design

L-T-P-Cr: 3-0-2-4

Pre-requisites: Discrete mathematics

Objectives/Overview:

- To provide theoretical concepts of computation.
- To provide knowledge on applications of automata theory in computer science
- To explore the principles, algorithms, and data structures involved in the design and construction of compilers

Course Outcomes:

At the end of the course, a student will understand:

Sl. No	Course Outcome (CO)	Mapping to POs
1.	Basic concepts of several formal languages, their machine portions and their relationship	PO1
.2.	Basic concepts and need of regular expression, design of Non-deterministic and deterministic finite automata and application of finite automata in compiler design	PO1, PO2
3.	Concepts of different grammars, parsers, and the role of the grammars in syntax analysis phase of compiler design.	PO1, PO2,PO3
4.	Design of push down automata and the relation between context free grammar and push down automata	PO1, PO2,PO3
5	Design of various types of Turing machines and the relation between recursively enumerable language and Turing machines.	PO1, PO2,PO3
	Various syntax directed translation methods used in compiler design.	PO1, PO2, PO3
7	Intermediate code generation, code optimization, and target code generation techniques in compiler design.	PO1, PO2, PO3

UNIT I: Finite automata and lexical analysis **Lectures: 08**

Study and Central concepts of automata theory, introduction to compiler, types of languages and Chomsky hierarchy, regular expression, regular expression and finite automata, application of finite automata in compiler design, applications of regular expressions, algebraic laws of regular expressions, design of a lexical analyzed generator, pumping lemma for regular languages.

UNIT II: Grammars and syntax analysis **Lectures: 10**

Grammars---context free, context sensitive, regular; parse trees, ambiguous grammars, role of grammars in syntax analysis, concept and design of Push down automata, parsers: top-down (recursive descent, predictive parser) and bottom up (shift-reduce, SLR parser, CLR parser).

UNIT III: Turing Machine **Lectures: 06**

Introduction to Turing machine (TM), design of TM, introduction and design of multitape multitrack TM, recursively enumerable language and TM, Undecidable Problem about Turing Machine.

UNIT IV: Syntax-directed translation **Lectures: 06**

Syntax directed definition, construction of syntax trees, bottom-up evaluation of S-attributed definition, L-attributed definitions, top-down translation.

UNIT V: Intermediate code generation and code optimization **Lectures: 07**

Intermediate languages, declarations, assignment statements, CASE statements; principal sources of optimization, optimization of basic blocks, loops in flow graphs, global data flow analysis.

UNIT VI: Code generation **Lectures: 05**

Issues in designing a code generator, basic blocks and flow graphs, a simple code generator, register allocation and assignment, DAG representation of basic blocks.

Text Book:

1) *Introduction to Automata Theory, Languages, and Computation, 2e*, by John E. Hopcroft, Rajeev Motwani, and Jeffery D. Ullman, Pearson Education.
2) *Compilers: Principles, Techniques and Tools* by A. V. Aho, R. Sethi, J. D. Ullman.