III- Year I-Semester	Name of the Course	L	T	P	C
PC3103	Automata Theory and Compiler Design	**3**	**0**	**0**	**3**
Prerequisites:					

Course Objectives:

- To learn fundamentals of Regular and Context Free Grammars and Languages
- To understand the relation between Regular Language and Finite Automata and machines
- To understand the relation between Contexts free Languages, Push Down Automata and Turing Machine
- To study various phases in the design of compiler and understanding the machine independent phases of compiler
- To understand machine dependent phases of compiler

UNIT-I: Finite Automata 12 hrs

Automata: Need for Automata Theory, Chomsky hierarchy, Acceptance of a string, Design of NFA with Є, NFA without Є, DFA, Equivalence of NFA, DFA

Finite Automata Conversions: Conversion from NFA Є to NFA, NFA to DFA, Minimization of DFA, Moore and Mealy Machines.

UNIT-II: Regular Expressions and Grammars 14 hrs

Regular Expressions: Regular Sets, Identity Rules, Equivalence of two RE, Manipulations of REs, Finite Automata and Regular Expressions, Inter Conversion, Equivalence between FA and RE, Pumping Lemma of Regular Sets, Closure Properties of Regular Sets

Grammars: Grammars, Classification of Grammars, Right and Left Linear Regular Grammars, Equivalence between RG and FA, Inter Conversion.Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars, Normal Forms- Chomsky Normal Form, Griebach Normal Form.

Unit-III: Push Down Automata and Turing Machines 12 hrs

Push Down Automata (PDA): Design of PDA, Deterministic PDA, Non-deterministic PDA, Equivalence of PDA and Context Free Grammars.

Turing Machine (TM): Design of Turing Machine, Deterministic TM, Non-deterministic TM.

UNIT-IV: Machine Independent Phases 14 hrs

Lexical Analysis: Logical phases of compiler**,** Lexical Analysis, Lexemes Tokens and patterns, Lexical Errors, Regular Expressions, Regular definitions for the language constructs, Strings, Sequences, Comments, Transition diagram for recognition of tokens, Reserved words and identifiers.

Syntax Analysis: Parsing definition, types of parsing, left recursion, left factoring, Top-down parsing, First and Follow, LL(1) Grammars, Non- Recursive predictive parsing, Bottom-up Parsers, Shift Reduce Parsing, LR parsers.

Semantic Analysis: Syntax Directed Translation, L-attributed and S-attributed definitions
Symbol tables: use and need of symbol tables.

UNIT-V: Machine Dependent Phases **12 hrs**

Intermediate Code Generation: Intermediate code, three address code, quadruples, triples, directed acyclic graph.

Code Optimization: Common sub expression elimination, copy propagation, dead code elimination, constant folding, strength reduction, loop optimization.

Code Generation: Basic blocks & flow graphs, Peephole optimization, Register allocation and assignment.

Text Books:

1. Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3rd Edition, Pearson, 2008
2. Theory of Computer Science-Automata, Languages and Computation, K. L. P. Mishra and N. Chandrasekharan, 3rd Edition, PHI, 2007
3. Compilers, Principles Techniques and Tools- Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D. Ullman,2nd ed, Pearson,2007.

Reference Books

1. Elements of Theory of Computation, Lewis H.P. &Papadimition C.H., Pearson /PHI
2. Theory of Computation, V. Kulkarni, Oxford University Press, 2013
3. Principles of compiler design, V. Raghavan, 2nd ed, TMH, 2011.
4. Compiler construction, Principles and Practice, Kenneth C Louden, CENGAGE

e- Resources & other digital material

https://nptel.ac.in/courses/106/104/106104028/
https://nptel.ac.in/courses/106/105/106105190/

University Academy You tube Channel for Automata Theory and Compiler Design:

https://www.youtube.com/playlist?list=PL-JvKqQx2AtdhlS7j6jFoEnxmUEEsH9KH
https://www.youtube.com/playlist?list=PL-JvKqQx2Ate5DWhppx-MUOtGNA4S3spT

GATE Lectures:

https://www.youtube.com/playlist?list=PLEbnTDJUr_IdM___FmDFBJBz0zCsOFxfK
https://www.youtube.com/playlist?list=PLMzYNEvC0P7FwwnrXwAjPq8zLTC4MDQKQ

Course Outcomes:

By the end the of the course, the student will be able to

CO1:Classify machines by their power to recognize languages.

CO2:Summarize language classes and grammars relationship among them with the help of Chomsky hierarchy.

CO3: employ finite state machines in problem solving and also illustrate deterministic and non-deterministic machines.

CO4: design and implement scanners and parsers.

CO5: perform code optimization to improve performance and apply algorithms to generate code.

CO-PO-PSO Mapping Matrix:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSP O1	PSP O2
CO -1	2	2	1	-	-	-	-	-	-	-	-	-	1	1
CO -2	1	2	2	-	-	-	-	-	-	-	-	-	2	1
CO -3	1	-	2	2	-	-	-	-	-	-	-	-	2	1
CO -4	2	-	2	1	-	-	-	-	-	-	-	-	1	1
CO -5	-	2	1	2	-	-	-	-	-	-	-	-	1	1

MICRO SYLLABUS

UNIT-I: Finite Automata **12 hrs**

Automata: Need for Automata Theory, Chomsky hierarchy, Acceptance of a string, Design of NFA with Є, NFA without Є, DFA, Equivalence of NFA, DFA

Finite Automata Conversions: Conversion from NFA Є to NFA, NFA to DFA, Minimization of DFA, Moore and Mealy Machines.

Unit	Module	Micro content	No of hrs
Unit-I Finite Automata	**Automata**	Need for Automata Theory, Chomsky hierarchy, Acceptance of a string, Design of NFA with Є, NFA without Є, DFA, Equivalence of NFA, DFA	5
	Finite Automata Conversions	Conversion from NFA Є to NFA, NFA to DFA, Minimization of DFA, Moore and Mealy Machines,	7

UNIT-II: Regular Expressions, Grammar **14 hrs**

Regular Expressions: Regular Sets, Identity Rules, Equivalence of two RE, Manipulations of REs, Finite Automata and Regular Expressions, Inter Conversion, Equivalence between FA and RE, Pumping Lemma of Regular Sets, Closure Properties of Regular Sets

Grammars: Grammars, Classification of Grammars, Right and Left Linear Regular Grammars, Equivalence between RG and FA, Inter Conversion. Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars, Normal Forms- Chomsky Normal Form, Griebach Normal Form.

Unit	Module	Micro content	No of hrs
UNIT-II Regular Expressions, Regular Grammars	**Regular Expressions**	Regular Sets, Identity Rules, Equivalence of two RE, Manipulations of REs, Finite Automata and Regular Expressions, Inter Conversion	3
		Equivalence between FA and RE, Pumping Lemma of Regular Sets, Closure Properties of Regular Sets	2
	Regular Grammars	Grammars, Classification of Grammars, Right and Left Linear Regular Grammars, Equivalence between RG and FA, Inter Conversion.	4
		Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars	2
		Normal Forms- Chomsky Normal Form, Griebach Normal Form.	3

Unit – III: Push Down Automata and Turing Machines 12 hrs

Push Down Automata (PDA): Design of PDA, Deterministic PDA, Non-deterministic PDA, Equivalence of PDA and Context Free Grammars.

Turing Machine (TM): Design of Turing Machine, Deterministic TM, Non-deterministic TM.

Unit	Module	Micro content	No of hrs
Unit – III Push Down Automata and Turing Machines	**Push Down Automata (PDA)**	Design of PDA, Deterministic PDA, Non-deterministic PDA, Equivalence of PDA and Context Free Grammars	6
	Turing Machine (TM)	Design of Turing Machine, Deterministic TM, Non-deterministic TM	6

UNIT-IV: Machine Independent Phases 14 hrs

Lexical Analysis: Logical phases of compiler**,** Lexical Analysis, Lexemes, Tokens and patterns, Lexical Errors, Regular Expressions, Regular definitions for the language constructs,

Strings, Sequences, Comments, Transition diagram for recognition of tokens, Reserved words and identifiers.

Syntax Analysis: Parsing definition, types of parsing, left recursion, left factoring, Top-down parsing, First and Follow, LL(1) Grammars, Non- Recursive predictive parsing, Bottom-up Parsers, Shift Reduce Parsing, LR parsers.

Semantic Analysis: Syntax Directed Translation, L-attributed and S-attributed definitions

Symbol tables: use and need of symbol tables.

Unit	Module	Micro content	No of hrs
UNIT-IV Machine Independent Phases	**Lexical Analysis**	Logical phases of compiler, Lexical Analysis, Lexemes, Tokens and patterns, Lexical Errors, Regular Expressions, Regular definitions for the language constructs, Strings, Sequences, Comments, Transition diagram for recognition of tokens, Reserved words and identifiers.	4
	Syntax Analysis	Parsing definition, types of parsing, left recursion, left factoring, Top-down parsing, First and Follow, LL(1) Grammars, Non- Recursive predictive parsing, Bottom-up parsers, Shift Reduce Parsing, LR parsers.	7
	Semantic Analysis	Syntax Directed Translation, L-attributed and S-attributed definitions	2
	Symbol tables	use and need of symbol tables.	1

UNIT-V: Machine Dependent Phases **12 hrs**

Intermediate Code Generation: Intermediate code, three address code, quadruples, triples, directed acyclic graph.

Code Optimization: Common sub expression elimination, copy propagation, dead code elimination, constant folding, strength reduction, loop optimization.

Code Generation: Basic blocks & flow graphs, Peephole optimization, Register allocation and assignment.

Unit	Module	Micro content	No of hrs
Unit-V Machine Dependent Phases	**Intermediate Code Generation**	Intermediate code, three address code, quadruples, triples, directed acyclic graph.	4
	Code Optimization	common sub expression elimination, copy propagation, dead code elimination, constant folding, strength reduction, loop optimization.	4
	Code generation	Basic blocks & flow graphs, Peephole optimization, Register allocation and assignment.	4

****