Chapter 1 : Introduction to Formal Languages **1-1 to 1-5**

1.1 Introduction to Formal Language 1-1

1.2 Alphabets and Languages 1-1

1.2.1 Kleene Closure 1-2

1.2.2 Recursive Definition of a Language 1-3

1.3 Finite Representation of Language 1-3

1.4 Introduction to Language Translation Logic 1-4

1.5 Essential of Translation 1-4

1.6 University Questions and Answers 1-5

Chapter 2 : Finite Automata **2-1 to 2-77**

2.1 Introduction to Finite Automata 2-1

2.1.1 Working of a Finite Automata 2-1

2.1.2 Some Important Terms 2-3

2.1.2(A) Alphabet 2-3

2.1.2(B) Strings (words) 2-3

2.1.2(C) Languages 2-3

2.1.2(D) Symbol 2-4

2.1.3 Applications of Finite Automata 2-4

2.2 Deterministic Finite Automata (DFA) 2-4

2.2.1 Definition of a DFA 2-4

2.2.2 Representation of a DFA 2-4

2.2.3 Designing a DFA 2-5

2.2.4 Solved Examples on DFA 2-8

2.2.4(A) Examples on Counting of Symbols 2-8

2.2.4(B) Examples on Substring 2-13

2.2.4(C) Examples of Divisibility 2-20

2.2.5 Language of DFA 2-22

2.3 Equivalence of DFAs 2-23

2.4 Closure Property of Language Accepted by a DFA 2-25

2.4.1 Union, Intersection, Difference 2-25

2.4.2 Complementation 2-28

2.5 Minimization of DFA 2-28

2.5.1 Algorithm for Minimization DFA's 2-28

2.6 Non-deterministic Finite Automata 2-32

2.6.1 Definition of NFA 2-32

2.6.2 Processing of a String by NFA 2-32

2.6.3 NFA to DFA Conversion 2-36

2.6.4 NFA with ε-Transitions 2-50

2.6.4(A) Equivalence of ε-NFA and NFA 2-50

2.6.4(B) The Formal Notation for an ε-NFA 2-52

2.6.4(C) ε-Closures 2-52

2.6.4(D) ε-NFA to DFA 2-53

2.6.5 Difference between NFA and DFA 2-58

2.7 Finite Automata as Output Devices 2-59

2.7.1 A Sample Mealy Machine 2-59

2.7.2 Formal Definition of a Mealy Machine 2-60

2.7.3 A Sample Moore Machine 2-60

2.7.4 Formal Definition of a Moore Machine 2-60

2.7.5 Conversion of a Mealy Machine into a Moore Machine 2-65

2.8 Minimization of a Mealy Machine 2-70

2.8.1 Conversion of a Moore Machine into a Mealy Machine 2-72

2.8.2 Difference between Moore and Mealy Machine 2-74

2.9 FSM for Vending Machine 2-75

2.10 FSM for Spell Checker 2-75

2.11 University Questions and Answers 2-75

Chapter 3 : Regular Expressions (RE) 3-1 to 3-39

3.1 Introduction 3-1

3.1.1 Precedence of Operators 3-3

3.2 Finite Automata Representing a Regular Expression 3-3

3.2.1 Composite Finite State Automata 3-3

3.2.2 Algebraic Laws for Regular Expressions 3-6

3.3 Determination of Regular Expression 3-7

3.3.1 Language Generated by a Regular Expression 3-7

3.3.2 Basic Properties of Regular Expressions 3-7

3.4 DFA to Regular Expression 3-16

3.4.1 State/Loop Elimination Process 3-16

3.4.1(A) A Generic One State Machine 3-16

3.4.1(B) A Generic Two State Machine 3-16

3.4.2 Arden's Theorem 3-25

3.4.2(A) Application of Arden's Theorem 3-26

3.5 FA Limitations 3-29

3.6 Pumping Lemma for Regular Languages 3-29

3.6.1 Definition of Pumping Lemma 3-30

3.6.2 Interpretation of Pumping Lemma 3-30

3.6.3 Proof of Pumping Lemma 3-30

3.6.4 Applications of Pumping Lemma 3-31

3.7 Closure Properties of Regular Language 3-34

3.7.1 Regular Language is Closed under Union 3-34

3.7.2 Regular Language is Closed under Concatenation 3-34

3.7.3 Regular Language is Closed under Kleene Star 3-35

3.7.4 Regular Language is Closed under Complementation 3-35

3.7.5 Regular Language is Closed under Intersection 3-35

3.7.6 Regular Languages are Closed under Difference 3-35

3.7.7 Regular Languages are Closed under Reversal 3-36

3.8 Decision Properties of Regular Language 3-37

3.9 Applications of R.E. 3-37

3.9.1 R.E. in Unix 3-37

3.9.2 Lexical Analysis 3-38

3.10 Myhill - Nerode Theorem 3-38

3.11 University Questions and Answers 3-38

Chapter 4 : Context Free Grammars (CFG) and Languages 4-1 to 4-66

4.1 An Example to Explain Grammar 4-1

4.2 Context Free Grammar 4-3

4.2.1 Notations 4-3

4.2.2 The Language of a Grammar 4-3

4.2.2(A) Sentential Form 4-4

4.2.2(B) Parse Tree 4-4

4.2.3 Writing Grammar for a Language 4-6

4.2.3(A) Union Rule for Grammar 4-10

4.2.3(B) Concatenation Rule for Grammar 4-10

4.3 Ambiguous Grammar 4-19

4.4 Simplification of CFG 4-25

4.4.1 Elimination of Useless Symbols 4-26

4.4.1(A) Non-generating Symbols 4-26

4.4.1(B) Non-reachable Symbols 4-27

4.4.2 Elimination of ε-productions 4-29

4.4.3 Elimination of Unit Productions 4-31

4.5 Normal Forms for CFG 4-35

4.5.1 Chomsky Normal Form (CNF) 4-35

4.5.1(A) Algorithm for CFG to CNF Conversion 4-35

4.5.2 Greibach Normal Form (GNF) 4-40

4.5.2(A) Removing Left Recursion 4-40

4.5.2(B) Algorithm for Conversion from CFG to GNF 4-41

4.6 Chomsky Classification for Grammar 4-46

4.6.1 Type 3 or Regular Grammar 4-46

4.6.2 Type 2 or Context Free Grammar 4-46

4.6.3 Type 1 or Context Sensitive Grammar 4-46

4.6.4 Type 0 or Unrestricted Grammar 4-47

4.6.5 Derivation Graph 4-47

4.7 Regular Grammar 4-47

4.7.1 DFA to Right Linear Regular Grammar 4-47

4.7.2 Right Linear Grammar to DFA 4-48

4.7.3 DFA to Left Linear Grammar 4-49

4.7.4 Left Linear Grammar to DFA 4-51

4.7.5 Right Linear Grammar to Left Linear Grammar 4-52

4.7.6 Left Linear Grammar to Right Linear Grammar 4-53

4.8 Pumping Lemma for CFG 4-58

4.9 Properties of Context-free Languages 4-60

4.9.1 Closure Properties 4-60

4.9.1(A) CFL is Closed under Union 4-60

4.9.1(B) CFL is Closed under Concatenation 4-60

4.9.1(C) CFL is Closed under Kleene Star 4-61

4.9.1(D) CFL is not Closed under Intersection 4-61

4.9.1(E) CFL is not Closed under Complementation 4-61

4.9.1(F) Intersection of CFL and RL 4-62

4.9.1(G) CFL is Closed under Reversal 4-62

4.9.2 Algorithmic Properties (Decision Properties) 4-62

4.10 Applications of CFG 4-62

4.10.1 Parsers 4-63

4.10.2 Markup Languages 4-63

4.10.3 XML and Document-Type Definitions 4-63

4.11 Cock-Younger-Kasami Algorithm 4-64

4.12 University Questions and Answers 4-64

Chapter 5 : Pushdown Automata (PDA) 5-1 to 5-34

5.1 Introduction to Pushdown Automata (PDA) 5-1

5.2 The Formal Definition of PDA 5-2

5.3 Instantaneous Description of a PDA 5-4

5.4 The Language of a PDA 5-6

5.4.1 Acceptance by Final State 5-6

5.4.2 Acceptance by Empty Stack 5-6

5.5 Non-deterministic PDA (NPDA) 5-15

5.6 Pushdown Automata and Context Free Language 5-19

5.6.1 Construction of PDA from CFG 5-19

5.6.2 Construction of CFG from PDA 5-22

5.7 Deterministic Push Down Automata (DPDA) 5-29

5.7.1 Regular Language and DPDA 5-29

5.8 Application of PDA 5-30

5.9 Parsing 5-30

5.9.1 Top-Down Parsing 5-30

5.9.2 Bottom-Up Parsing 5-32

5.10 Differentiate between FA and PDA. 5-33

5.11 University Questions and Answers 5-33

Chapter 6 : Turing Machine (TM) 6-1 to 6-32

6.1 Introduction to Turing Machine 6-1

6.2 The Formal Definition of Turing Machine 6-3

6.2.1 A String Accepted by TM 6-3

6.2.2 Instantaneous Descriptions for Turing Machines 6-5

6.3 Turing Machines as Computer of Functions 6-17

6.4 Extension of Turing Machine 6-26

6.4.1 Two-way Infinite Turing Machine 6-26

6.4.2 A Turing Machine with Multiple Heads 6-26

6.4.3 Multi-Tape Turing Machine 6-26

6.4.3(A) Composite TM 6-28

6.4.4 Non-Deterministic Turing Machine 6-28

6.5 Universal Turing Machine 6-29

6.6 Linear Bounded Automata 6-30

6.7 Difference between FA and TM. 6-30

6.8 University Questions and Answers 6-31

Chapter 7 : Undecidability and Intractable Problems 7-1 to 7-25

7.1 Recursively Enumerable and Recursive Language 7-1

7.1.1 Turing Acceptable Language 7-1

7.1.2 An Un-decidable Problem that is RE (Recursively Enumerable) 7-5

7.2 Enumerating a Language 7-5

7.2.1 Finite and Infinite Sets 7-6

7.3 Chomsky Hierarchy 7-8

7.3.1 Type 3 or Regular Grammar 7-8

7.3.2 Type 2 or Context Free Grammar 7-8

7.3.3 Type 1 or Context Sensitive Grammar 7-8

7.3.4 Type 0 or Unrestricted Grammar 7-8

7.3.5 Compare Type 0, Type 1, Type 2 and Type 3 Grammars 7-9

7.4 Un-decidability 7-9

7.4.1 Halting Problem of a Turing Machine 7-10

7.4.2 Un-decidability of Post Correspondence Problem 7-11

7.4.3 Modified PCP Problem 7-13

7.5 Computational Complexity 7-14

7.5.1 P and NP-class Problem 7-14

7.5.2 Intractable Problems 7-15

7.6 The Classes P and NP 7-15

7.6.1 Problem Solvable in Polynomial Time 7-15

7.6.2 An Example : Kruskal's Algorithm 7-15

7.6.3 Minimal Spanning Tree 7-16

7.6.4 Kruskal's Algorithm 7-16

7.6.5 Kruskal's Algorithm using a Turing Machine 7-17

7.6.6 Polynomial-Time Reduction 7-17

7.6.7 NP-Complete Problems 7-18

7.7 An NP-Complete Problem 7-18

7.7.1 The Satisfiability Problem (SAT) 7-18

7.7.2 Traveling Salesman Problem (TSP) 7-18

7.7.3 Tractable and Intractable 7-18

7.7.4 3-SAT Problem 7-19

7.8 Representing SAT Instances 7-19

7.8.1 NP-Completeness of the SAT Problem 7-20

7.9 A Restricted Satisfiability Problem 7-20

7.9.1 Normal Forms for Boolean Expressions 7-20

7.9.2 Converting Expressions to CNF 7-20

7.9.3 Clique Problem is NP-Complete 7-21

7.9.4 The Problem of Independent Sets 7-22

7.9.5 The Node-Cover Problem 7-22

7.9.6 The Directed Hamilton-Circuit Problem 7-23

7.9.7 Undirected Hamiltonian Circuit 7-23

7.10 University Questions and Answers 7-24

❑❑❑