SAVITRIBAI PHULE PUNE UNIVERSITY
T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS
TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16
TITLE OF PAPER : Theoretical Computer Science
Code No. : CS-332

Semester III **Total Lectures : 48**

Aim:
To have a introductory knowledge of automata, formal language theory and computability.

Objectives:

- To have an understanding of finite state and pushdown automata.
- To have a knowledge of regular languages and context free languages.
- To know the relation between regular language, context free language and corresponding recognizers.
- To study the Turing machine and classes of problems.

Prerequisite:

- Sets, Operations on sets, Finite & infinite sets Formal Language
- Relation, Equivalence Relation,(reflexive, transitive and symmetric closures)

1. Introduction **[3]**

1.1 Symbol, Alphabet, String, Prefix& & Suffix of Strings, Formal Language, Operations on Languages.
1.2 Regular Expressions (RE) : Definition & Example
1.3 Regular Expressions Identities.

2. Finite Automata **[12]**

2.1 Deterministic finite Automaton – Definition, DFA as language recognizer, DFA as a pattern recognizer.
2.2 Nondeterministic finite automaton – Definition and Examples.
2.3 NFA TO DFA : Method (From Book 4)
2.4 NFA with ε- transitions Definition and Examples.
2.5 NFA with ε-Transitions to DFA & Examples
2.6 Finite automaton with output – Mealy and Moore machine, Definition and Examples.
2.7 Minimization of DFA, Algorithm & Problem using Table Method.

3. Regular Languages **[5]**

3.1 Regular language-Definition and Examples.
3.2 Conversion of RE To FA-Examples.
3.3 Pumping lemma for regular languages and applications.
3.4 Closure properties of regular Languages
(Union, Concatenation, Complement, Intersection and Kleene closure)

4. Context Free Grammar and Languages **[12]**

4.1 Grammar - Definition and Examples.
4.2 Derivation-Reduction - Definition and Examples.
4.3 Chomsky Hierarchy.
4.4 CFG : Definition & Examples. LMD, RMD, ,Parse Tree
4.5 Ambiguous Grammar : Concept & Examples.
4.6 Simplification of CFG :
  4.6.1 Removing Useless Symbols,
  4.6.2 Removing unit productions
  4.6.3 Removing ϵ productions & Nullable symbols
4.7 Normal Forms :
  4.7.1 Chomsky Normal Form (CNF) Method & Problem

4.7.2 Greibach Normal form (GNF) Method & Problem

4.8 Regular Grammar : Definition.

4.8.1 Left linear and Right Linear Grammar-Definition and Example.

4.8.2 Equivalence of FA & Regular Grammar

4.8.2.1 Construction of regular grammar equivalent to a given DFA

4.8.2.2 Construction of a FA from the given right linear grammar

4.9 Closure Properties of CFL's(Union, concatenation and Kleen closure) Method and examples

5. Push Down Automaton [6]

5.1 Definition of PDA and examples

5.2 Construction of PDA using empty stack and final State method : Examples using stack method

5.3 Definition DPDA & NPDA, their correlation and Examples of NPDA

5.4 CFG (in GNF) to PDA : Method and examples

6. Turing Machine [10]

6.1 The Turing Machine Model and Definition of TM

6.2 Design of Turing Machines

6.3 Problems on language recognizers.

6.4 Language accepted by TM

6.5 Types of Turing Machines(Multitrack TM,Two way TM, Multitape TM,Non-deterministic TM)

6.6 Introduction to LBA (Basic Model) &CSG.(Without Problems)

6.7 Computing TM, Enumerating TM, Universal TM

6.8 Recursive Languages

6.5.1. Recursive and Recursively enumerable Languages.

6.5.2. Difference between recursive and recursively enumerable language.

6.9 Turing Machine Limitations

6.10 Decision Problem, Undecidable Problem, Halting Problem of TM

References :-

1 Introduction to Automata theory, Languages and computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House.

2. Introduction to Automata theory, Languages and computation By John Hopcroft, Rajeev Motwani and Jeffrey Ullman –Third edition Pearson Education

3. Introduction to Computer Theory Daniel I. A. Cohen – 2nd edition – John Wiley & Sons

4. Theory of Computer Science (Automata, Language & Computation) K. L. P. Mishra & N. Chandrasekaran, PHI Second Edition

5. Introduction to Languages and The Theory of Computation John C. Martin TMH, Second Edition

SAVITRIBAI PHULE PUNE UNIVERSITY
T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS
TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16
TITLE OF PAPER : Compiler Construction
Code No. : CS-342

Semester IV **Total Lectures : 48**

Aim :
To understand the various phases of a compiler and to develop skills in designing a compiler

Objective :

- To understand design issues of a lexical analyzer and use of Lex tool
- To understand design issues of a parser and use of Yacc tool
- To understand issues related to memory allocation
- To understand and design code generation schemes

1. Introduction **[5]**

1.1 Definition of Compiler, Aspects of compilation.
1.2 The structure of Compiler.
1.3 Phases of Compiler – Lexical Analysis,Syntax Analysis, Semantic Analysis, Intermediate Code generation, code optimization, code generation.
1.4 Error Handling
1.5 Introduction to one pass & Multipass compilers, cross compiler, Bootstrapping.

2. Lexical Analysis(Scanner) **[5]**

2.1 Review of Finite automata as a lexical analyzer,
2.2 Applications of Regular Expressions and Finite Automata (lexical analyzer, searching using RE), Input buffering, Recognition of tokens
2.3 LEX: A Lexical analyzer generator (Simple Lex Program)

3. Syntax Analysis(Parser) **[20]**

3.1 Definition , Types of Parsers
3.2 Top-Down Parser –
- 3.2.1Top-Down Parsing with Backtracking: Method & Problems
- 3.2.2 Drawbacks of Top-Down parsing with backtracking,
- 3.2.3Elimination of Left Recursion(direct & indirect)
- 3.2.4Need for Left Factoring & examples

3.3 Recursive Descent Parsing : Definition
- 3.3.1Implementation of Recursive Descent Parser Using Recursive Procedures

3.4 Predictive [LL(1)]Parser(Definition, Model)
- 3.4.1Implementation of Predictive Parser[LL(1)]
- 3.4.2 FIRST & FOLLOW
- 3.4.3 Construction of LL(1) Parsing Table
- 3.4.4Parsing of a String using LL(1) Table

3.5 Bottom-Up Parsers
3.6 Operator Precedence Parser -Basic Concepts
- 3.6.1Operator Precedence Relations form Associativity & Precedence
- 3.6.2 Operator Precedence Grammar
- 3.6.3 Algorithm for LEADING & TRAILING(with ex.)
- 3.6.4Algorithm for Operator Precedence Parsing (with ex.)
- 3.6.5Precedence Functions

3.7 Shift Reduce Parser
- 3.7.1Reduction, Handle, Handle Pruning
- 3.7.2Stack Implementation of Shift Reduce Parser (with examples)

3.8 LR Parser
- 3.8.1Model
- 3.8.2Types [SLR(1), Canonical LR, LALR] Method & examples.

3.9 YACC (from Book 3) –program sections, simple YACC program for expression evaluation

4. Syntax Directed Definition [8]

4.1Syntax Directed Definitions(SDD)
- 4.1.1 Inherited & Synthesized Attributes
- 4.1.2 Evaluating an SDD at the nodes of a Parse Tree, Example

4.2 Evaluation Orders for SDD's
- 4.2.1 Dependency Graph
- 4.2.2 Ordering the Evaluation of Attributes
- 4.2.3 S-Attributed Definition
- 4.2.4 L-Attributed Definition

4.3 Application of SDT
- 4.3.1 Construction of syntax trees,
- 4.3.2 The Structure of a Type

4. 4 Translation Schemes
- 4.4.1 Definition, Postfix Translation Scheme

5. Memory Allocation [2]

5.1 Memory allocation – static and dynamic memory allocation,

5.2 Memory allocation in block structure languages, Array allocation and access.

6. Code Generation and Optimization [8]

6.1 Compilation of expression –
- 6.1.1 Concepts of operand descriptors and register descriptors with example.
- 6.1.2 Intermediate code for expressions – postfix notations,
- 6.1.3 triples and quadruples, expression trees.

6.2 Code Optimization – Optimizing transformations – compile time evaluation, elimination of common sub expressions, dead code elimination, frequency reduction, strength reduction

6.3 Three address code
- 6.3.1. DAG for Three address code
- 6.3.2 The Value-number method for constructing DAG's.

6.4 Definition of basic block, Basic blocks And flow graphs

6.5 Directed acyclic graph (DAG) representation of basic block

6.6 Issues in design of code generator

References :-

1. Compilers: Principles, Techniques, and Tools ,Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman
2. Principles of Compiler Design By : Alfred V. Aho, Jeffrey D. Ullman (Narosa Publication House)
3. LEX & YACC (O'reilly Publication)