RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Information Technology, V-Semester

Departmental Elective IT- 503 (A) Theory of Computation

Course Objectives

- Student learns some fundamental concepts in automata theory and designing of Finite Automata, conversion NFA to DFA. Application of Finite Automata in computer science and real world.
- Obtain minimized DFA and Application of regular expression and conversion from RE to Finite Automata and Finite Automata to Regular Expression and Proving language are not regular.
- Designing of CFG's , Construction of parse trees, finding and removing ambiguity in grammars, simplification of CFG, Conversion of grammar to Chomsky Normal Form ,Greibach normal form.
- Designing problems on Pushdown Automata and conversion of grammar to PDA, PDA to Grammar.
- Designing Turing machines, understanding the working of various types of Turing machines and study P and NP type problem.

UNIT I

Introduction of the theory of computation, Finite state automata – description of finite automata, properties of transition functions, Transition graph, designing finite automata, FSM, DFA, NFA, 2-way finite automata, equivalence of NFA and DFA, Mealy and Moore machines.

UNIT II

Regular grammars, regular expressions, regular sets, closure properties of regular grammars, Arden's theorem, Myhill-Nerode theorem, pumping lemma for regular languages, Application of pumping lemma, applications of finite automata, minimization of FSA.

UNIT III

Introduction of Context-Free Grammar - derivation trees, ambiguity, simplification of CFGs, normal forms of CFGs- Chomsky Normal Form and Greibach Normal forms, pumping lemma for CFLs, decision algorithms for CFGs, designing CFGs, Closure properties of CFL's.

UNIT IV

Introduction of PDA, formal definition, closure property of PDA, examples of PDA, Deterministic Pushdown Automata, NPDA, conversion PDA to CFG, conversion CFG to PDA.

UNIT V

Turing machines - basics and formal definition, language acceptability by TM, examples of TM, variants of TMs – multitape TM, NDTM, Universal Turing Machine, offline TMs, equivalence of single tape and multitape TMs. Recursive and recursively enumerable languages, decidable and undecidable problems – examples, halting problem, reducibility. Introduction of P, NP, NP complete, NP hard problems and Examples of these problems.

Reference Books:

1. Daniel I.A. Cohen,"Introduction to Computer Theory",Wiley India.
2. John E Hopcroft, Jeffrey D. Ullman and Rajeev Motwani, "Introduction to Automata Theory, Languages and Computation", Pearson Education.
3. K.L.P Mishra & N.Chandrasekaran,"Theory of Computer Science", PHI Learning.
4. Peter Linz, "Introduction to Automata Theory and Formal Languages", Narosa Publishing.
5. John C Martin, "Introduction to languages and the theory of computation", TATA McGraw Hill.

Course Outcomes

At the completion of the course, students will be able to...

- Convertbetween finite automata, regular grammars, and regular expression representations of regular languages
- Apply the pumping lemma for regular languages to determine if a language is regular
- Convertbetweengrammars and push-down automata for context-free languages
- Determine if a language is regular or context-free
- Demonstrate that a grammar is ambiguous
- Translate a context-free grammar from one form to another
- Produce simple programs for a Turing Machine
- Explain the concept of undecidability
- List examples of undecidable problems

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Information Technology, V-Semester

Departmental Elective IT- 503 (B) Microprocessor and Interfacing

Course Objectives:

- To introduce basic concepts of microprocessor
- Tointroduceserialandparallelbusstandards.
- To introduce programming in assembly language.
- To introduce basic concepts of interfacing memory and peripheral devices to a microprocessor.

UNIT –I:
Evolution of microprocessor, single chip micro computers, Micro processor Application, Microprocessor and its architecture, addressing modes, instruction, Instruction sets, Arithmetic and Logic Instruction, Program control instruction, Introduction –8086 family, procedure and macros, connection , Timing and Troubleshooting interrupt, 80286, 80836 and 80486 micro processor system concept.

UNIT –II:
Microprocessor Cycle, AIU, Timing and control Unit, Register data, Address bus, Pin Configuration, Intel 8086 instruction, Opcode and operands, limitation word size. Programming the microprocessor Assembly language, The Pentium and Pentium Pro Micro Processorwith features, Pentium II, Pentium III and Pentium –IV Microprocessor with software changes. Instruction set for Intel 8086, Introduction Intimation and data formats, Addressing modes, Status flags, Symbols and abbreviations, programming of microprocessors, Assembly language, high level language, areas of application of various languages, Stacks, Sub routines system, software, commands in assembly language, software Development, Debugging program, Modular programming, Structured programming, Top-down, Bottom-up design , MACRO microprogramming.

UNIT-III:
Assembly language programming with Examples like Addition of 8/16-bit Binary number, subtraction of 8/16 bit binary number, Address partitioning, addressing mode, type of addressing mode, memory and I/o interfacing, Data transfer schemes, Interfacing device and I/o devices I/o ports, Basic I/o Interfacing MDS, Micro controllers, I/o processor and co-processors ,Microcomputer Development system, Single chip micro computers, intel 8748 intel 8051, inter 8096, intel 8049intel 2920/2921, I/o processor UPI-425,UPI-41,42, Co-processor, math processor math co-processor –8087, 80287, 80387DX 803875x

UNIT –IV:
Bus Interface I/o port Addressing, decoding 8279, Programmable key board/display interface, 8254 Internal Timer, 16550 programmable communication interface A/D, 8259A Programmable Interrupt Controller, 8237 DMA Controller, Shared bus operation, disk Memory system Video display. ISA Bus, Extended ISA (EISA) and VESA Local Buses, Peripheral Component Inter Connect (Pc I) Bus, Parallel Printer interface (LPT) Universal serial Bus (USB) Accelerated graphics port (AGP),Programmable Communication interfere 8251 VSART CRT Controller 8275, 6854, Floppy disk Controller 8272, I/o processor 8089.

UNIT –V:
Memory Unit, RAM,SRAM, DRAM,ROM, PROM EPROM, EEPROM Nonvolatile RAM semiconductor Technology for memory, Shift register, Magnetic Memory, Tap, disc, main memory and secondary memory

cache memory, program memory and Data Memory, Real and virtual memory Buses, memory Addressing capacity of CPU, processing speed of computer

Reference Books:

1. Douglas V Hall, “Microprocessors and interfacing –Programming & Hardware” TMH
2. Barry B. Brey, “The intel Microprocessor –8086”, Pearson Education
3. Kenneth J.Ayala,”The 8086 Microproccessor: Programming & Interfacing The PC”,CengageLearning
4. Krishna Kant,”Microprocessors and Microcontrollers”, PHI Learning
5. A.K.Ray KM Bhurchandi, “Advanced Microprocessor and peripherals” McGraw Hill
6. R.S. Gaonkar ,”Microprocessors and interfacing”, TMH

Course Outcomes:

At the completion of the course, students will be able to...

- Explainthemicroprocessor’sandMicrocontroller’sinternalarchitecture
- Applyknowledgeanddemonstrateprogrammingproficiencyusingthevariousaddressingmodesanddatatransferinstructionsofthetargetmicroprocessorandmicrocontroller.
- CompareacceptedstandardsandguidelinestoselectappropriateMicroprocessor(8085&8086)andMicrocontrollertomeetspecifiedperformancerequirements.
- Analyzeassemblylanguageprograms
- DesignelectricalcircuitrytotheMicroprocessorI/Oportsinordertointerfacetheprocessortoexternaldevices.
- Evaluateassemblylanguageprograms

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Information Technology, V-Semester

Departmental Elective IT- 503 (C) Object Oriented Analysis and Design

Course Objectives:
The prime objective of this course is to teach the students to analyze, design and implement object-oriented software systems

UNIT I Introduction: Overview of object oriented concepts, Object Orientation, OO Software Development life cycle, Object oriented methodology, OO Themes, Modeling Concepts, Role of Analysis and Design in software development, Overview of various OOAD methodologies, OO approach vs conventional approach, Unified process of Software development, UML, Goals of UML, Overview of different models.

UNIT II Static Modeling using Class Diagrams: Object and Class concepts, Link and association, Multiplicity, Ternary Association, Recursive association, Association class, Generalization and Inheritance, Multiple inheritance, Aggregation and composition, Abstract Class, Packages.

UNIT III Dynamic Modeling using State Diagrams: Events, States, Transitions and conditions, Types of state diagrams, Continuous life cycle state diagrams, one-shot life cycle state diagrams, Sub states, Nested state diagrams, Signal generalization, Concurrency, Junction state, Synch state, Relation of class and state models.

UNIT IV Interaction Modeling: Use case Models, Actors and use cases, Use Case relationships, Use of Use cases for validation and verification, Sequence diagrams, Procedural sequence models, activity models, swim lanes, Dynamic concurrency, decomposing an activity, Communication Diagrams, Architectural Modeling: Component and Deployment Diagrams.

UNIT V System design and class design, Implementation modeling, Implementing structure and implementing functionality, Frameworks, Design Patterns, Object-Oriented Languages and their comparison, Object-Oriented Databases, ObjectOriented Programming Style, CORBA, COM, DCOM.

Reference Books:
1. Michael Blaha, Object-Oriented modeling and Design with UML, PHI
2. Mahesh P. Matha, Object-Oriented Analysis and Design Using UML, PHI
3. D Jeya Mala and S. Geetha, Object-Oriented Analysis and Design Using UML, McGraw Hill
4. Andrew Haigh, Object-Orientd Analysis and Design, TMH
5. O' Docherty, Object-Oriented Analysis and Design Understanding, System Development with UML 2.0, Wiley India

Course Outcomes:
At the end of the course student will be able to:

1. Explain OOAD concepts
2. Perform object oriented analysis and develop static model of system after identifying classes and their relationships
3. Develop dynamic model of system by identifying states and events
4. Develop interaction model of system by drawing use case, sequence and activity diagrams
5. Select an appropriate design patternand effectively construct object-oriented programs