

OBJECTIVES:

- To understand different forms of intermediate languages and analyzing programs.
- To understand optimizations techniques for single program blocks.
- To apply optimizations on procedures and low level code.
- To explore and enhance inter procedural optimizations.
- To enhance resource utilization.

UNIT I INTERMEDIATE REPRESENTATION OF PROGRAMS AND ANALYSIS 9+6

Structure of an Optimizing Compiler – Compiler Construction tools – LIR, MIR, HIR, DAG, SyntaxTree and Postfix. Analysis: Control Flow Analysis, Iterative Data Flow Analysis, Static Single Assignment – A Linear Time Algorithm for Placing ϕ -Nodes, Basic Block Dependence, Alias Analysis. Introduction to LLVM – Compiling a language.

UNIT II LOCAL AND LOOP OPTIMIZATIONS 9+6

Early Optimizations: Constant-Expression Evaluation – Scalar Replacement of Aggregates – Algebraic Simplifications and Re-association – Value Numbering – Copy Propagation – Sparse Conditional Constant Propagation. Redundancy Elimination: Common – Subexpression Elimination – Loop-Invariant Code Motion – Partial-Redundancy Elimination – Redundancy Elimination and Reassociation – Code Hoisting. Loop Optimizations: Induction Variable Optimizations – Unnecessary Bounds Checking Elimination. LLVM pass –LLVM Test Infrastructure.

UNIT III PROCEDURE OPTIMIZATION AND SCHEDULING 9+6

Procedure Optimizations: Tail-Call Optimization and Tail-Recursion Elimination – Procedure Integration – In-Line Expansion – Leaf-Routine Optimization and Shrink Wrapping. Code Scheduling: Instruction Scheduling – Speculative Loads and Boosting – Speculative Scheduling – Software Pipelining – Trace Scheduling – Percolation Scheduling. Control-Flow and Low-Level Optimizations: Unreachable-Code Elimination – Straightening – If Simplifications – Loop Simplifications –Loop Inversion Un-switching – Branch Optimizations – Tail Merging or Cross Jumping – Conditional Moves – Dead-Code Elimination – Branch Prediction – Machine Idioms and Instruction Combining. LLVM API procedure optimization.

UNIT IV INTER PROCEDURAL OPTIMIZATION 9+6

Symbol table Runtime Support – Interprocedural Analysis and Optimization: Interprocedural Control- Flow Analysis – The Call Graph – Interprocedural Data-Flow Analysis – Interprocedural Constant Propagation – Interprocedural Alias Analysis – Interprocedural Optimizations – Interprocedural Register Allocation – Aggregation of Global References. LLVM – Interprocedural Analyses.

UNIT V OPTIMIZING FOR MEMORY 9+6

Register Allocation: Register Allocation and Assignment – Local Methods – Graph Coloring Priority Based Graph Coloring. Computations on Iteration Spaces- Optimization for the Memory Hierarchy: Impact of Data and Instruction Caches – Instruction-Cache Optimization – Scalar Replacement of Array Elements – Data-Cache Optimization – Scalar vs. Memory-Oriented Optimizations. Software Prefetching – Parallelization – Instruction Level Parallelism – Automatic Parallelization.

TOTAL: 45 +30 = 75 PERIODS

OUTCOMES:

Upon completion of the course, the student will be able to

- Identify the different optimization techniques that are possible for a sequence of code.
- Design performance enhancing optimization techniques.
- Manage procedures with optimal overheads.
- Understand modern programming language features and constructs.
- Learn to work on a larger software project.

REFERENCES:

1. Steven.S. Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufman Publishers, 1997.
2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, "Compilers: Principles, Techniques, and Tools", Addison Wesley, Second Edition, 2007.
3. Y.N.Srikant, Priti Shankar, "The Compiler Design Handbook – Optimizations and Machine Code Generation", CRC Press, Second Edition, 2008.
4. Andrew W. Appel, Jens Palsberg, "Modern Compiler Implementation in Java", Cambridge University Press, Second Edition, 2002.
5. Keith Cooper, Linda Torczon, "Engineering a Compiler", Morgan Kaufmann, Second Edition, 2011.
6. Randy Allen and Ken Kennedy, Optimizing Compilers for Modern Architectures: A Dependence based Approach, Morgan Kaufman, 2001.

CO	PO						PSO		
	1	2	3	4	5	6	1	2	3
1.	√		√			√	√	√	√
2.	√	√	√	√			√	√	√
3.	√		√	√			√	√	√
4.	√		√	√				√	
5.	√		√	√		√	√	√	√

CP5253**MACHINE LEARNING****L T P C
3 0 0 3****OBJECTIVES:**

- To understand the concepts of Machine Learning.
- To appreciate supervised learning and their applications.
- To appreciate the concepts and algorithms of unsupervised learning.
- To understand the theoretical and practical aspects of Probabilistic Graphical Models.
- To appreciate the concepts and algorithms of advanced learning.

UNIT I INTRODUCTION**8**

Machine Learning–Types of Machine Learning –Machine Learning process- preliminaries, testing Machine Learning algorithms, turning data into Probabilities, and Statistics for Machine Learning- Probability theory – Probability Distributions – Decision Theory.

UNIT II SUPERVISED LEARNING**10**

Linear Models for Regression – Linear Models for Classification- Discriminant Functions, Probabilistic Generative Models, Probabilistic Discriminative Models – Decision Tree Learning – Bayesian Learning, Naïve Bayes – Ensemble Methods, Bagging, Boosting, Neural Networks, Multi-layer Perceptron, Feed- forward Network, Error Back propagation - Support Vector Machines.