

Unit-V

1. Machine-independent optimization is performed on:

- A) Source code only
- B) Target machine code
- C) Intermediate representation of the program
- D) Assembly language only

Answer: C

2. The primary objective of machine-independent optimization is to:

- A) Increase hardware speed
- B) Improve program efficiency without considering target architecture
- C) Generate object code
- D) Eliminate syntax errors

Answer: B

3. Which of the following is a major source of optimization?

- A) Redundant computations
- B) Parsing conflicts
- C) Lexical errors
- D) Buffer overflow

Answer: A

4. Eliminating unnecessary calculations helps to:

- A) Increase code size
- B) Reduce execution time
- C) Increase memory usage
- D) Reduce readability only

Answer: B

5. Data-flow analysis is used to gather information about:

- A) Lexical tokens
- B) Flow of data through program variables
- C) Parse trees
- D) Grammar productions

Answer: B

6. Data-flow analysis is generally performed on:

- A) Syntax trees only
- B) Symbol tables
- C) Flow graphs
- D) Input buffers

Answer: C

7. A flow graph consists of:

- A) Tokens and lexemes
- B) Basic blocks and control-flow edges
- C) Variables and constants
- D) Terminals and non-terminals

Answer: B

8. In data-flow analysis, a basic block is:

- A) A sequence of instructions with one entry and one exit
- B) A collection of procedures
- C) A group of variables
- D) A memory allocation unit

Answer: A

9. The process of collecting information that can be used to optimize a program is called:

- A) Parsing
- B) Data-flow analysis

- C) Lexical analysis
- D) Garbage collection

Answer: B

10. Reaching definitions analysis determines:

- A) Variables that are no longer used
- B) Definitions that may reach a program point
- C) Available registers
- D) Parse tree nodes

Answer: B

11. Live variable analysis determines:

- A) Variables currently stored in memory
- B) Variables whose values may be used in the future
- C) Variables declared globally
- D) Variables with constant values

Answer: B

12. Available expressions analysis identifies:

- A) Expressions that have already been computed and not invalidated
- B) Expressions containing constants only
- C) Expressions with syntax errors
- D) Expressions used in parsing

Answer: A

13. Data-flow equations are solved using:

- A) Lexical rules
- B) Iterative algorithms
- C) Parsing tables
- D) Recursive descent

Answer: B

14. The initial framework for data-flow analysis includes:

- A) Domain, transfer functions, and meet operator
- B) Tokens and lexemes
- C) Stacks and heaps
- D) Registers and caches

Answer: A

15. Constant propagation replaces:

- A) Variables with machine instructions
- B) Known constant values at compile time
- C) Loops with functions
- D) Registers with memory locations

Answer: B

16. If the compiler determines that $x = 10$, then replacing subsequent uses of x with 10 is called:

- A) Dead code elimination
- B) Constant folding
- C) Constant propagation
- D) Copy propagation

Answer: C

17. Constant folding is the process of:

- A) Replacing variables with constants
- B) Evaluating constant expressions at compile time
- C) Eliminating loops
- D) Assigning constants to registers

Answer: B

18. Which optimization removes assignments like $x = y$?

- A) Loop unrolling
- B) Copy propagation
- C) Constant folding
- D) Register allocation

Answer: B

19. Dead code elimination removes:

- A) Unreachable or unused code
- B) Variable declarations
- C) Loop structures
- D) Syntax errors

Answer: A

20. Partial redundancy elimination aims to:

- A) Remove all loops
- B) Eliminate expressions that are redundant on some execution paths
- C) Reduce the number of variables
- D) Minimize memory allocation

Answer: B

21. An expression is partially redundant if:

- A) It is never executed
- B) It is redundant along some, but not all, paths
- C) It appears only once
- D) It contains constants only

Answer: B

22. Loop optimization is important because:

- A) Loops are rarely executed
- B) Most program execution time is spent in loops

C) Loops reduce code readability

D) Loops simplify parsing

Answer: B

23. A loop in a flow graph is characterized by:

A) Multiple entry points

B) A back edge

C) No exit point

D) No control flow

Answer: B

24. A natural loop consists of:

A) A header and all nodes that can reach the back edge source

B) Only one instruction

C) A sequence of tokens

D) A set of parse trees

Answer: A

25. Loop-invariant code motion moves computations:

A) Into the loop body

B) Outside the loop when their values do not change

C) Into the symbol table

D) Into activation records

Answer: B

26. Which optimization reduces loop control overhead by replicating loop bodies?

A) Loop unrolling

B) Constant propagation

C) Register allocation

D) Dead code elimination

Answer: A

27. Strength reduction replaces:

- A) Expensive operations with cheaper ones
- B) Variables with constants
- C) Registers with memory
- D) Loops with functions

Answer: A

28. Replacing multiplication by 2 with addition is an example of:

- A) Constant folding
- B) Copy propagation
- C) Strength reduction
- D) Partial redundancy elimination

Answer: C

29. Which optimization technique can improve both execution speed and code size?

- A) Dead code elimination
- B) Loop unrolling
- C) Register spilling
- D) Heap allocation

Answer: A

30. Machine-independent optimizations are performed before:

- A) Lexical analysis
- B) Parsing
- C) Target-specific code generation
- D) Source code editing

Answer: C