**UNIT-IV**

1. One of the purposes of using intermediate code in compilers is to

(A) make parsing and semantic analysis simpler.

(B) improve error recovery and error reporting.

(C) increase the chances of reusing the machine-independent code optimizer in other compilers.

(D)improve the register allocation.

Answer:(C)

2. Consider the following Syntax Directed Translation Scheme (SDTS), with non-terminals {S, A} and terminals {a,b}.

S→aA{print1} S→a{print2} S→Sb{print3}

Using the above SDTS, the output printed by a bottom-up parser, for the input aab is:

(A) 1 3 2

(B) 2 2 3

(C) 2 3 1

(D) syntax error

Answer: (C)

3. Which languages necessarily need heap allocation in the runtime environment?

(A) Those that support recursion

(B) Those that using dynamic scoping

(C) Those that allow dynamic data structure

(D) Those that use global variables

Answer: (C)

4. In the compiler, the function of using intermediate code is:

(A) to improve the register allocation

(B) to increase the error reporting & recovery.

(C) to make semantic analysis easier.

(D) to increase the chances of re-using the machine-independent code optimizer in other compilers.

Answer: (D)

5. In how many types of optimization can be divided?

(A) two types

(B) three types

(C) four types

(D) five types

Answer: (A)

6. Which algorithm invokes a function GETREG()?

(A) Code motion algorithm

(B) Code optimization algorithm

(C) Intermediate Code

(D) Code generation algorithm

Answer: (D)

7. Which statement is an abstract form of intermediate code?

(A) 3- address

(B) 2-address

(C) address

(D) Intermediate code

Answer: (A)

8. Which mapping is described by the implementation of the syntax-directed translator?

(A) Parse table

(B) Input

(C) Output

(D) Input-Output

Answer: (D)

9. How many descriptors are used for track both the registers (for availability) and addresses (location of values) while generating the code?

(A) 2

(B) 3

(C) 4

(D) 5

Answer: (A)

10. x \* 2 can be replaced by x << 1 is an example of?

(A) Algebraic expression simplification

(B) Accessing machine instructions

(C) Strength reduction

(D) Code Generator

Answer: (C)

11. The following code is an example of?

voidadd\_ten(int x)

{

return x + 10;

printf(""value of x is %d"", x);

}

(A) Redundant instruction elimination

(B) Unreachable code

(C) Flow of control optimization

(D) Reachable code

Answer: (B)

12. Code generator uses \_\_\_\_\_\_ function to determine the status of available registers and the location of name values.

(A) setReg

(B) cinReg

(C) pfReg

(D) getReg

Answer:(D)

13. Which of the following is not a form of Intermediate representation?

(A) Abstract Syntax Tree

(B) 3-address code

(C) Directed cyclic Graph

(D) Reverse Polish Notation

Answer: (C)

14. Which of the following is not a form of Intermediate representation?

(A) Abstract Syntax Tree

(B) 3-address code

(C) Directed cyclic Graph

(D) Reverse Polish Notation

Answer: (C)

15. Some code optimizations are carried out on the intermediate code because \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(A) They enhance the portability of the complier to other target processors

(B) Program analysis is name accurate on intermediate code than on machine code

(C) The information from data flow analysis cannot otherwise be used for optimization

(D) The information from the front end cannot otherwise be used for optimization

Answer: (B)

16. Consider the following intermediate program in three address code

p = a - b

q = p \* c

p = u \* v

q = p + q

Which one of the following corresponds to a static single assignment form of the above code?

(A) p1 = a - b

q1 = p1 \* c

p1 = u \* c

q1 = p1 + q1

(B) p3 = a - b

q4 = p3 \* c

p4 = u \* c

q5 = p4 + q4

(C) p1 = a - b

q1 = p2 \* c

p3 = u \* c

q2 = p4 + q3

(D) p1 = a - b

q1 = p \* c

p2 = u \* c

q2 = p + q

Answer:(B)

17. Which one of the following is FALSE?

(A) A basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end.

(B) Available expression analysis can be used for common subexpression elimination.

(C) Live variable analysis can be used for dead code elimination.

(D) x = 4 ∗ 5 ⇒ x = 20 is an example of common subexpression elimination.

Answer:(D)

**PART-B**

1. Open MPI is a Message Passing Interface library project combining technologies and resources from several other projects that is developed and maintained by a consortium of academic, research, and industry partners. In particular case, Intel C Compiler icc is used as the backend compiler but we also have variants that use GCC. The following are the packages that are involved in the versioning of OpenMPI model.

Fft= Numpy + Scipy + plot

Gsl = Scipy + plot

Python = Scikit + Matlib + Seaborn + plot + numpy

Gcc = Boost + atlas + hdf5 + plot

Torque = Fft + Gsl

OpenMPI = Torque + Python + Gcc

1. For the above hierarchy of the OpenMPI versioning, interpret the instruction and generate Three Address Code for the same.
2. Create Directed Acyclic Graph for the Three Address Code generated from the above Q.No (1a)

2. . For the code given below, determine the following : Constant propagation, live-variable analysis.

(a) N = 10;

(b) k = 1;

(c) prod = 1;

(d) MAX = 99;

(e) while (k > N) {

(f) read(num);

(g) if (MAX/num < prod) {

(h) print("cannot compute prod");

(i) break;

(j) }

(k) prod = prod \* num;

(l) k++;

(m) }

print(prod);

3. Construct Three address Code for the following Code and Explain different kinds of Three address code representation.

begin

i := 1;

do begin

prod := prod + a[i] \* b[i];

i = i+ 1;

end

while i <= 20

end

Note: Consider the type of array is integer.

3. Translate the conditional statement if a<b then 1 else 0 into three address code

4. Explain the following with example: i) Quadruples ii) Triples iii) Indirect triple

5. What are different intermediate code forms? Discuss different Three Address code types and implementations of Three Address statements.

6.Explain the translation scheme to produce three address code for assignment statements.

7.Explain the Backpatching process for Boolean expression.

8.Explain the following terms: i) Register Descriptor ii) Address Descriptor iii) Instruction Cost.

9.Explain various issues in the design of the code generation.

**UNIT-V**

1. Two standard storage-allocation strategies are

A. Stack Allocation and Queue Allocation

B. Stack Allocation and Static Allocation

C. Static Allocation and Queue Allocation

D. Simple Allocation and Static Allocation

ANSWER: B

1. Which field is not present in activation record

A. saved machine status

B. register allocation

C. optional control link

D. temporaries

ANSWER: B

1. which statement is correct about passing by value parameter?

A. it cannot change the actual parameter value

B. it can change the actual parameter value

C. parameter is always in read only mode

D. parameter is always in write only mode

ANSWER: A

1. Identify the synthesized and inherited attribute from the attributes: gen, kill, in, out

A. synthesized, inherited, inherited, synthesized

B. inherited, synthesized, synthesized, synthesized

C. synthesized, synthesized, inherited, inherited

D. synthesized, synthesized, inherited, synthesized

ANSWER: D

1. Which one of the following is FALSE?

A. A basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end.

B. Available expression analysis can be used for common subexpression elimination.

C. Live variable analysis can be used for dead code elimination.

D. x = 4 ∗ 5 => x = 20 is an example of common subexpression elimination.

ANSWER: D

1. The effect of the instruction LD R2, a(R1) is

A. R2= contents(a) + contents(R1)

B. R2= contents(a+ contents(R1))

C. R2= a + contents(R1)

D. R2= contents (contents(a+contents(R2)))

ANSWER: B

1. In the context of compiler design, reduction in strength refers to

A. code optimization obtained by the use of cheaper machine instructions.

B. reduction in accuracy of the output.

C. reduction in the range of values of input variables.

D. reduction in efficiency of the program.

ANSWER: A

1. In static allocation, names are bound to storage at \_\_\_\_\_ time

A. Compile

B. Runtime

C. Debugging

D. both (a) and (b)

ANSWER: A

9, Issues in code generation (i) Addressing Modes (ii) Choice of Evaluation Order (iii) Microprocessor Mnemonics (iv) Linking and Loading

A. Only (i), (ii)

B. Only (i), (ii), (iv)

C. Only (ii), (iii), (iv)

D. All of them

ANSWER: D

10. Activation Record stores: (i)Parameters (ii) Local Variables (iii) Parameters and Local Variables (iv) Parameters, Local Variables and code for procedures

A. (i)

B. (ii)

C. (iii)

D. (iv)

ANSWER: C

11. Function call actions are divided into sequences

A. Calling and composition

B. Return and composition

C. Calling and return

D. None of the other options

ANSWER: C

12. Evaluation of actual parameters is done by

A. Callee

B. Caller

C. Both Caller and Callee

D. None of the other options

ANSWER: B

13. If an activation of procedure ‘A’ calls procedure ‘B’ then which one is TRUE?

A. Activation of B must end before the activation of A can end.

B. Activation of A must end before the activation of B can end.

C. Activation of A must end before the activation of B can start.

D. Activation of B must start after the activation of A can end.

ANSWER: A

14. Which of the following field of an activation record will point to the activation record of the caller?

A. Retrned Values

B. Access Link

C. Temporaries

D. Control Link

ANSWER: D

15. Consider the following three address code. Identify the CORRECT collection of different optimization can be performed? m = 3

j = n

v = 2 \* n

limit = integer n / 2

L1: j = j – 1

t4 = 4 \* j

t5 = a[t4]

if t5 > limit – v goto L1

A. Code Motion, Constant Folding, Induction Variable Elimination, Reduction in Strength

B. Copy Propagation , Code Motion, Deadcode Elimination, Reduction in Strength

C. Constant Folding, Copy Propagation, Deadcode Elimination, Reduction in Strength

D. Code Motion, Constant Folding, Copy Propagation, Induction Variable Elimination

ANSWER: A

16. Which of the following tasks is managed by a runtime stack?

A. Static data and functions

B. Garbage collection

C. Procedure calls and returns

D. All of the above

ANSWER: C

17. Is code-optimization could be performed over the basic-block? x = t3

a[t2] = t5

a[t4] = x

goto B1

A. Not Possible

B. Yes Possible

C. Not enough information

D. Don’t know

ANSWER: B

18. The sequence of procedure calls of a program corresponds to which traversal of the activation tree?

A. In order traversal

B. Pre order traversal

C. Post order traversal

D. Level-order travel

ANSWER: B

19. The \_\_\_\_\_\_ code must not, in any way, change the meaning of the program.

A. input

B. output

C. print

D. execute

ANSWER: A

20.Optimization should increase the \_\_\_\_\_ of the program and if possible, the program should demand less number of resources.

A. length

B. count

C. speed

D. cost

ANSWER: C

PART-B

1. Write the code sequence for the d:=(a-b)+(a-c)+(a-c).

Apply code generation algorithm to generate a code sequence for the three address statement.

1. Consider the following block and construct a DAG for it-

(1) a = b x c

(2) d = b

(3) e = d x c

(4) b = e

(5) f = b + c

(6) g = f + d

1. Explain different storage allocation strategies.
2. Discuss briefly about DAG representation of basic blocks.
3. Explain in detail about primary structure-preserving transformations on basic blocks.
4. Discuss in detail about global data analysis.