7. (a) What are the limitations of access links? How displays solve those issues? Explain an example. (7)

(b) General Code for the following three-address statements assuming a and b are arrays whose elements are 4-byte values: (7)

x = a [i]

y = b Explain the activities of caller and callee in stack [j]

a[i] = y

b[j] = x

8. (a) Discuss about copy propagation and dead code elimination. (7)

(b) With suitable examples, explain about live-variable analysis. (7)

– DEC 2024

[07 BENG – 3145]

III/IV B.Tech. DEGREE EXAMINATION.

First Semester

Computer Science and Engineering

COMPILER DESIGN

(Effective from the admitted batch of 2022–2023)

Time : Three hours Maximum : 70 marks

Question No.1 is compulsory.

Answer any FOUR from the remaining questions.

All questions carry equal marks.

PART – A (14 marks)

1. (a) Copy propagation leads to dead-code elimination, justify this with example.

(b) Give three-address code for the statement: do i = i + l; while (a[i]< v);

(c) What is syntax-directed definition?

(d) What is the purpose of Loader/Linker in language processing?

(e) What are the basic functions of the memory manager?

(f) What is a preprocessor? Mention its objectives.

(g) Write about the sub-division of run-time memory.

PART B — (56 marks)

2. (a) Define the formal definition and notational conventions of CFG. (7)

(b) Explain the procedure for eliminating ambiguity from a grammar. Give an example. (7)

3. (a) Design LALR(l) parser for the following grammar: (7)

S ->aAd |bBd| aBc |bAc

A->e

B -> e where a, b, c, d, e are terminals.

(b) Give the SDT scheme for desk calculator. (7)

4. (a) Discuss about the principal sources of optimization with examples. (7)

(b) Explain in brief about the DAG based local optimization. (7)

5. (a) Generate the flow-graphs for the following expressions: (7)

S -> id: = E | S;S | if E then S else S | do S while E.

E -> id + id | id

(b) Define garbage collection is important for code optimization? Explain garbage collection by using reference counting. (7)

6. (a) What is an LL(1) grammar? Can you convert every context free grammar into LL(1). (7)

(b) Consider the following grammar (7)

E->T+E | T

T->V*T/V

V->id

Write down the procedures for the non-terminals of the grammar to make a recursive descent parser.

III/IV B.Tech. DEGREE EXAMI...

First Semester

Computer Science and Engineering

COMPILER DESIGN

(Effective from the admitted batch of 2021-2022)

Time : Three hours Maximum : 70 marks

First question is compulsory.

Answer any FOUR questions from the following.

All the questions carry equal marks.

1. (a) Define Cross Compiler.

(b) What is input buffering?

(c) Explain the Concept of predictive parsing.

(d) What is type propagation?

(e) Discuss branch optimization.

(f) Define object code.

(g) What is heap management?

2. (a) Explain about boot strapping in Compilers.

(b) What is a symbol table and why it is important in a compiler explain.

3. (a) Describe the concept of a deterministic finite automation [DFA] in detail.

(b) What is the primary role of lexical analysis in the compilation process? Explain.

4. (a) Explain shift reduce parsing with an example.

(b) Discuss about Recursive Descent parsing with an example.

5. (a) Provide examples of intermediate code representation used in Compiler Construction.

(b) Discuss about type checking and type conversion with examples.

6. (a) Explain the process of dead code elimination in a program.

(b) Explain the importance of loop optimization in improving program performance.

7. (a) What role does the instruction cache play in code scheduling and performance Optimizations explain in-details with a neat diagram?

(b) Consider the following grammar :

$E \rightarrow T + E \mid T$

$T \rightarrow V^*T \mid V$

$V \rightarrow id$

Write down the procedures for the non-terminals of the grammar to make a recursive descent parser.

8. (a) Describe various approaches to organize a symbol table.

(b) Explain about inherited and synthesized attributes.

DEC 2024

[07 BENG – 3122]

III/IV B.Tech. DEGREE EXAMINATION.

First Semester

Computer Science and Engineering

COMPILER DESIGN

(Effective from the admitted batch of 2020-2021)

Time : Three hours Maximum : 70 marks

First question is compulsory.

Answer any FOUR questions from the following.

All questions carry equal marks.

1. (a) Define boot strapping in the context of Compiler design.

   (b) Write the types of language processing system.

   (c) Write short notes on recursive descent parsing.

   (d) Write a briefly about an intermediate code.

   (e) Write short notes on peephole optimization.

   (f) Describe the concept of program and instruction costs in code generation.

   (g) Write about Error handling routines.

2. (a) Explain about boot strapping in Compilers.

2. (a) Define compiler. Describe the phases of a compiler with a neat sketch.

(b) Explain how input buffering helps lexical analyser in compilation process.

3. (a) State and explain the rules used to construct the LR(1) items.

(b) What is an LL (1) grammar? Can you convert every context free grammar into LL (1).

4. (a) How does bottom-up parsing differ from top-down parsing?

(b) Find whether the following grammar is LL (I) or not

$S \rightarrow absa \mid aaAb,\ A \rightarrow baAb \mid b$

5. (a) Explain the Concept of three-address code and its role in intermediate code generation.

(b) Write procedure to construct CLR parsing table.

6. (a) How do different optimization techniques affect the trade-off between code size and Execution speed.

(b) What is machine independent optimization? What are the different techniques used for it?

7. (a) What are the trade-offs between generating compact code and generating code for speed?

(b) Write the algorithm to generate basic blocks and flow graph for quick sort algorithm.

8. Explain the following two classes of local machine independent transformations

(a) Structure preserving transformations

(b) Algebraic transformations

DEC 2023

[07 BENG - 3122]

III/IV B.Tech. DEGREE EXAMINATION

Computer Science and Engineering

First Semester

COMPILER DESIGN

(Effective from the admitted batch of 2020-2021)

Time : 3 hours Max. Marks : 70

Question No. 1 is compulsory.
Answer any FOUR from the remaining.
All questions carry equal marks.

1. (a) Define regular expression with an example.
   (b) Define the terms token and lexeme.
   (c) Explain the role of Parser in compiler model.
   (d) Define type checking and write the rules for type checking.
   (e) Write about dead code elimination.
   (f) What is a simple target machine model?
   (g) What are the key components of an error handling system in a compiler?

2. (a) Describe the lexical analysis phase in the context of compiler of construction.
   (b) Write a regular expression for identifiers and reserved votes. Design the transition diagrams for them.

(P.T.O.)

3. (a) Write the limitations of recursive parser with an example of grammar.

   (b) Write an algorithm to find LR (0) items and give an example.

4. (a) Discuss the advantages and disadvantages of LL(1) and LR(1) parsing.

   (b) Eliminate the left recursion from the following grammar.

   $S \rightarrow Bb \mid a$ $\qquad$ $B \rightarrow Bc \mid Sd \mid c$

5. (a) What is the role of Semantic analysis in the compilation process, and how does it differ from syntax analysis.

   (b) Discuss the concept of back patching with an example.

6. (a) What is loop inversion, and when is it applied for optimization?

   (b) Draw the DAG for the following expression.
   a +(b*d) + c* (b*d) + e + a/(b*d).

7. What are main issues in code generation within a compiler and define a simple target machine model used in code generation.

8. (a) Explain about inheritance and synthesized attributes.

   (b) What is data flow analysis? Explain its role in code of imagination.

* * *

[07 BENG - 3212]

[07 BENG - 3212]

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Effective from the admitted batch of 2019–2020)

Time : Three hours Maximum : 70 marks

First Question No.1 is Compulsory.

Answer any FOUR from remaining questions.

All questions carry equal marks.

Answer ALL parts of any question at one place

1. Answer the following in brief :

   (a) What are the uses of cross compilers?

   (b) Determine whether the following regular expression define the same language.
   (ab)* and a*b*

   (c) Is macro processing a phase in compilation? Justify your answer.

   (d) What is left-factoring? Explain.

   (e) Define e-closure.

   (f) What language does the grammar $S \rightarrow aSa \mid bSb \mid c$ generate?

   (g) What is ICAN? Explain.

2. *Define compiler and explain various phases of compiler in detail. Also write down the output for the following expression after each phase.* a : = b*c–d.

3. (a) Explain the role of Lexical analysis and the issues with lexical analyser.

   (b) Explain the general format of LEX program with example.

4. (a) Construct the predictive parser for the following grammar :

   $S \rightarrow (L) \mid a$

   $L \rightarrow L, s \mid s.$

   Construct the behaviour of the parser on the sentence (a, a) using the grammar specified above.

   (b) Analyse whether the following grammar is LR(1) or not. Explain your answer with reasons.

   $S \rightarrow L, R$

   $S \rightarrow R$

   $L \rightarrow *R$

   $L \rightarrow id$

   $R \rightarrow L$

5. Consider the grammar given below :

   $E \rightarrow E + T$

   $E \rightarrow T$

   $T \rightarrow T * F$

   $T \rightarrow F$

   $F \rightarrow (E)$

   $F \rightarrow id$

   Prepare LR Parsing table for the above grammar. Give the moves of LR Parser on id * id + id.

   Also explain error recovery in LR parsing.

6. (a) Explain different schemes *f* storing name attribute in symbol table.

   (b) Describe the method of generating syntax directed definition of control statements.

7. (a) Explain the principle sources of code optimization in detail.

   (b) Explain the DAG representation of the basic block with example.

8. (a) Efficient code generation requires the remember *f* internal architectural of the target machine. Justify your answer with an example.

   (b) How the instruction forms effect the computation time? Explain.

5. Consider the ...

2. Define compiler and explain various phases of compiler in detail. Also write down ...

[07 BENG - 3209] (C-19)

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science Engineering

COMPILER DESIGN

(Common with IT)

(Effective from the admitted batch of 2015–2016)

For the academic year 2020-2021 batch only

Time : Three hours Maximum : 70 marks

First Question is compulsory

Answer any FOUR questions from the remaining

Write all parts of any question at one place.

1. (a) Write regular expression over alphabet {a, b, c} containing at least one 'a' and at least one 'b'.

(b) What is meant by pass and phase?

(c) What is left factoring?

(d) Define handle pruning.

8. (a) Optimize the following loop and also construct flow graph:

```
Begin
    Prod = 0
    i = 1
    do
    Begin
        Prod = Prod + a[i] * b[i]
        i = i + 1
    End
    While (i  20);
End
```

(e) What is abstract syntax tree? Give an example.

(f) Define constant folding.

(g) What are the rules to identity leader basic blocks?

2. (a) Show the output produced by different stages in compiler for the expression a := b*c/36; where a, b and c are real numbers.

(b) Explain bootstrapping a compiler with suitable diagrams.

3. (a) Construct DFA from the following NFA: M = ({p, q, r, s}, {0, 1}, δ, p, {s}).

Present State	Next State	
	0	1
p	{p, q}	{p}
q	{r}	{r}
r	{s}	-
s	{s}	{s}

(b) What is the structure of LEX program? Write a LEX program that accepts the keywords: "begin, if, else" and identifier: "abc".

4. (a) Illustrate brute force parsing technique with suitable example.

(b) Computer FIRST and FOLLOW of the flowing grammar:

S → aBDh, B → cC, C → bc | ε, D → EF, E→g | ε, F→f | ε.

5. Design CLR Parser for the following grammar.

S → L = R | R
L → *R | a
R → L

6. (a) Write about syntax directed definition and syntax directed translation.

(b) Describe about type expressions.

7. (a) Generate three address code for the following switch block:

```
Switch(ch)
{
Case 1: c = a + b;
    break;
Case 2: c = a – b;
    break;
Case 3: C = a * b;
    break;
}
```

(b) Translate the expression (a + b) / (c + d) * (a + b/ c) – d into quadruples, triples and indirect triples.

(e) What is abstract syntax tree? Give an example.

[107 BENG – 3209]

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Common with Information Technology)

(Effective from the admitted batch of 2015–2016)

Time : Three hours Maximum : 70 marks

First Question is compulsory.

Answer any FOUR from the remaining questions.

All questions carry equal marks.

Answer all parts of any question at one place.

1. (a) Define a compiler and list the phases of a compiler.

   (b) What is meant by a transition diagram? Give an example.

   (c) What are the problems with Top-down parsing?

   (d) Construct a syntax tree for the arithmetic expression a * – (b+c).

(e) *Define DAG and give an example.*

(f) Which object code form makes the code generation process easier and why?

(g) List out the contents of a symbol table.

2. (a) Explain about Cross compilers.

(b) Describe compiler construction tools.

3. (a) Construct DFA for the regular expression (11+0)*(00+1)*

(b) Explain about Lexical analysis.

4. (a) Explain Shift-reduce parsing with an example.

(b) Discuss about Recursive Descent Parsing.

5. (a) Translate the arithmetic expression a:=b*–c+b*–c into

(i) Syntax tree ii) Postfix notation (iii) Three-address code.

(b) Discuss about Type Checking and Type conversions.

6. (a) Discuss basic blocks and flow graphs with an example.

(b) Explain machine dependent code optimization techniques.

7. (a) Discuss various strategies used in register allocation and assignment.

(b) Explain about code generators.

8. (a) Describe various approaches to organize a symbol table.

(b) Explain about Error Handling Routines.

[07 BENG – 3209]

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Common with Information Technology)

(Effective from the admitted batch of 2015-2016)

Time : Three hours Maximum : 70 marks

Question No. 1 is compulsory.

Answer any FOUR from the remaining questions.

ALL questions carry equal marks.

1. (a) What is meant by boot strapping?

(b) Mention the disadvantages of lexical analysis.

(c) What is the classification of top-down parsing?

(d) What is meant by syntax tree?

(e) Explain flow graph.

(f) What is meant by register allocation?

(g) Describe about lexical phase errors.

2. (a) Discuss about the compiler construction tools.

(b) Explain the various phases of a compiler with a neat sketch.

3. (a) Explain how transition diagram is helpful in lexical analyzer.

(b) What is compiler? Explain the role of lexical analyzer?

4. (a) Construct predictive parser for the following grammar and verify whether the string 'ab' is accepted by it or not.

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \varepsilon$

$B \rightarrow \varepsilon$

(b) Explain error recovery in predictive parsing.

5. (a) Discuss about syntax directed definition and syntax directed translation.

(b) Explain type checking and type conversions.

6. (a) Explain DAG and its use. Write the procedure to construct the DAG for a statement.

(b) Discuss about data flow analysis of structured programs.

7. (a) Describe about code generation algorithm.

(b) Explain issues in code generation.

8. (a) Write briefly about static allocation and stack allocation.

(b) Discuss about Heap management.

[07 – 3216]

III/IV B.Tech. DEGREE EXAMINATION

Second Semester

Computer Science Engineering

COMPILER DESIGN

(Common with B.Tech. Information Technology)

(Effective from the Admitted Batch of 2006–2007)

Time : Three hours Maximum : 70 marks

Q.No. 1 which is compulsory.

Answer any FOUR questions from the remaining.

Answer ALL parts of any question at one place.

1. (a) Define regular expression. Find a regular expression for the language $L = \{a^n b^m \mid n >= 0, m >= 1\}$. $(7 \times 2 = 14)$

   (b) List the phases of a compiler.

   (c) Define lexical error with an example.

   (d) State and define the conflicts that occur in shift-reduce parsing.

2. (a) Discuss about the compiler construction tools.

7. (a) Describe about code generation algorithm.

7. (a) Discuss in detail about loop optimization techniques. (7)

   (b) Briefly explain about semantics preserving transformations with an example. (7)

8. (a) Write a short note on Register allocation and Register assignment. (6)

   (b) Discuss about code generation algorithm with an example. (8)

———————

(e) *What is S-attributed definition?*

(f) What is strength reduction?

(g) List the issues in the design of a code generator.

$(4 \times 14 = 56)$

2. (a) Obtain a regular expression for the following finite automata. (8)

		0	1
→	q_1	q_2	q_3
*	q_2	q_2	q_3
*	q_3	q_2	q_1

(b) Design an NFA and DFA accepting all strings ending with 01 over an alphabet {0, 1}. (6)

3. (a) Explain different phases of a compiler with a neat diagram, showing the output of each phase, using the statement a = a * 5 + b + 2. (Assume that a and b are floating point numbers) (10)

(b) Illustrate the differences between Compiler and Interpreter. (4)

4. (a) Discuss about tokens, patterns and lexemes with suitable examples. (8)

(b) Explain in detail about the role of lexical analyzer. (6)

5. (a) Construct SLR parsing table for the following grammar. (10)

S→Aa | bAc | dc | bda

A→d

(b) Compute the FIRST and FOLLOW sets for the following grammar : (4)

S→iEtSS' | a

S'→eS | E

E→b

6. (a) Define Syntax-Directed Definition. Give SDD of a simple desk calculator and construct annotated parse trees for the following expressions.
(3 + 4)*(5 + 6) n (7)

(b) Write down the implementation of three address code and translate the following expressions into quadruple, triple and indirect triple.
–(a+b) *(c+d) *(a+c) (7)

6. (a) Write down the implementation of three address code and translate the following expressions into quadruple, triple and indirect triple. (7)

$$-(a+b)*(c+d)*(a+c)$$

(b) Discuss about loop optimization techniques. (7)

7. (a) What is a basic block and flow graph? Write the algorithm to partition the TAC into basic blocks. Explain the construction of a flow graph with an example. (7)

(b) Write the algorithm for determining the liveness and next-use information for each statement in a basic block. Explain the algorithm with an example. (7)

8. (a) Discuss about machine dependent (peephole) optimization. (7)

(b) Discuss in detail about Garbage Collection. (7)

(e) What is S-attributed definition?

(f) What is

[07 – 3216]

III/IV B.Tech. DEGREE EXAMINATION

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Common with B.Tech. Information Technology)

(Effective from the admitted batch of 2006–2007)

Time : Three hours Maximum : 70 marks

Q.No. 1 which is compulsory.

Answer any four questions from the remaining.

Answer all parts of any question at one place.

$(7 \times 2 = 14)$

1. (a) Design a DFA that accepts set of all strings containing even number of 0's and odd number of 1's over an alphabet {0, 1}.

(b) What is the purpose of symbol table in a compiler?

(c) Differentiate between a pass and a phase.

(d) What is Reduce-Reduce conflict? Give an example.

(e) *List the different types of intermediate representations.*

(f) Differentiate between DAG and syntax tree.

(g) Determine the costs of the following instruction sequences :

LD R0, x

LD R1, y

SUB R0, R0, R1

BLTZ *R3, R0

$(4 \times 14 = 56)$

2. (a) $M = (\{q_1, q_2, q_3\}, \{0,1\}, \delta, q_1, \{q_3\})$ is a Non-deterministic finite automation, where δ is given by

$\delta(q_1, 0) = \{q_2, q_3\}$

$\delta(q_1, 1) = \{q_1\}$

$\delta(q_2, 0) = \{q_1, q_2\}$

$\delta(q_2, 1) = \phi$

$\delta(q_3, 0) = \{q_2\}$

$\delta(q_3, 1) = \{q_1, q_2\}$

Construct an equivalent DFA. (7)

(b) Explain Kleene's theorem. Construct ϵ-NFA for the regular expression (0 + 1) 1 (0 + 1)* using Kleene's theorem. (7)

3. (a) Explain about lexical analyzer generator. (7)

(b) Explain the different phases of a compiler with a neat diagram, showing the output of each phase, using the following statements.(7)

$$i = i * 70 + j + 2$$

4. (a) Explain about various data structures used in a compiler. (7)

(b) Construct the CLR parsing table for the following grammar. Justify your design with an example. (7)

S → CC

C → cC

C → d

5. (a) Construct SLR parsing table for the grammar and parse the string ( )( )$. (7)

S → S(S)

S → ε

(b) Define Syntax-Directed Definition. Give SDD of a simple desk calculator and construct annotated parse trees for the following expressions. (7)

$(3 + 4)*(5 + 6)n$

III/IV B.Tech. DEGREE EXAMINATION

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Effective from the admitted batch of 2019–2020)

Time : Three hours Maximum : 70 marks

First question is compulsory.

Answer any FOUR from the remaining questions.

All questions carry equal marks.

Answer ALL parts of any question at one place.

1. Answer the following in brief :
   (a) List and explain few compiler construction tools.
   (b) Define the term 'Tokens' in lexical analysis phase.
   (c) What are the grass of error handless in a parser?
   (d) Write the advantages and disadvantages of operator precedence parsing.
   (e) How can you generate 3-Address code?

(c) What are the patterns used for code optimization?

2. (a) Explain Single pass and multipass compiler with examples.

   (b) Describe how various phases could be combined as a pass in a compiler.

3. (a) Explain Shift-reduce parsing. What are the conflicts that may occur during shift-reduce parsing?

   (b) For the grammar given below, calculate the operator precedence relation and precedence functions.

$$E \rightarrow E+E \mid E-E \mid E*E \mid E/E \mid E\,E \mid (E) \mid -E \mid id$$

4. Explain 3-addres codes and mention its types. How would you implement the three-address statement? Explain with suitable examples.

5. (a) Discuss about allocation strategies of Heap allocation and stack allocation.

   (b) What is Activation record? Explain the purpose of different fields in an activation record.

6. (a) How to trace the data flow analysis of structured program? Explain.

   (b) Explain about Peephole optimization technique.

7. (a) Explain the concept of Syntax directed translation with an example.

   (b) Explain the syntax directed translation of switch statement.

8. Write short notes on the following :

   (a) Code generation algorithm

   (b) Error handling routines in compliers.

   (c) Machine dependent optimization.

11.07.2022

[07 BENG - 3209] (C-19) 05 JUL 2022

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science Engineering

COMPILER DESIGN

(Common with IT)

(Effective from the admitted batch of 2015–2016)

For the academic year 2020-2021 batch only

ime : Three hours Maximum : 70 marks

First Question is compulsory

Answer any FOUR questions from the remaining.

Write all parts of any question at one place.

1. (a) Write regular expression over alphabet {a, b, c} containing at least one 'a' and at least one 'b'.

(b) What is meant by pass and phase?

(c) What is left factoring?

(d) Define handle pruning.

(c) Define context-free grammar.

(d) List the various forms of three instructions.

6. (a) How to trace the

optimization? What are the patterns used for code

(e) What is abstract syntax tree? Give example.

(f) Define constant folding.

(g) What are the rules to identity leader blocks?

2. (a) Show the output produced by different st in compiler for the expression a := b*c where a, b and c are real numbers.

(b) Explain bootstrapping a compiler w suitable diagrams.

3. (a) Construct DFA from the following NF M = ({p, q, r, s}, {0, 1}, δ, p, {s}).

Present State	Next State	
	0	1
p	{p, q}	{p}
q	{r}	{r}
r	{s}	-
s	{s}	{s}

(b) What is the structure of LEX program? Writ a LEX program that accepts the keyword "begin, if, else" and identifier: "abc".

(a) Illustrate brute force parsing technique with suitable example.

(b) Computer FIRST and FOLLOW of the flowing grammar:

S → aBDh, B → cC, C → bc | ε, D → EF, E → g | ε, F → f | ε.

Design CLR Parser for the following grammar.

S → L = R | R
L → *R | a
R → L

(a) Write about syntax directed definition and syntax directed translation.

(b) Describe about type expressions.

(a) Generate three address code for the following switch block:

```
Switch(ch)
{
Case 1: c = a + b;
      break;
Case 2: c = a – b;
     break;
Case 3: C = a * b;
      break;
}
```

(b) Translate the expression (a + b) / (c + d) * (a + b/ c) – d into quadruples, triples and indirect triples.

8. (a) Optimize the following loop and also co
flow graph:

```
Begin
    Prod = 0
    i = 1
    do
    Begin
        Prod = Prod + a{i] * b[i]
        i = i + 1
    End
    While (i  20);
End
```

[07 BENG - 3212]

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Effective from the admitted batch of 2019–2020)

Time : Three hours Maximum : 70 marks

First Question No.1 is Compulsory.

Answer any FOUR from remaining questions.

All questions carry equal marks.

Answer ALL parts of any question at one place

Answer the following in brief :

(a) What are the uses of cross compilers?

(b) Determine whether the following regular expression define the same language.
(ab)* and a*b*

(c) Is macro processing a phase in compilation? Justify your answer.

(d) What is left-factoring? Explain.

(e) Define e-closure.

(f) What language does the grammar $S \rightarrow aSa \mid bSb \mid c$ generate?

(g) What is ICAN? Explain.

2. Define compiler and explain various phase compiler in detail. Also write down the outpu the following expression after each ph a : = b*c–d.

3. (a) Explain the role of Lexical analysis and issues with lexical analyser.

(b) Explain the general format of LEX prog with example.

4. (a) Construct the predictive parser for following grammar :

$S \rightarrow (L) \mid a$

$L \rightarrow L, s \mid s.$

Construct the behaviour of the parser on sentence (a, a) using the grammar specif above.

(b) Analyse whether the following grammar LR(1) or not. Explain your answer w reasons.

$S \rightarrow L, R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow id$

$R \rightarrow L$

Consider the grammar given below :

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow id$

Prepare LR Parsing table for the above grammar. Give the moves of LR Parser on id * id + id.

Also explain error recovery in LR parsing.

(a) Explain different schemes *f* storing name attribute in symbol table.

(b) Describe the method of generating syntax directed definition of control statements.

(a) Explain the principle sources of code optimization in detail.

(b) Explain the DAG representation of the basic block with example.

(a) Efficient code generation requires the remember *f* internal architectural of the target machine. Justify your answer with an example.

(b) How the instruction forms effect the computation time? Explain.

(a) Discuss about allocation strategies of Heap allocation and stack allocation.

(b) What is Activation record? E

(c) Define context-free grammar.

11.07.2022

[07 BENG – 3209] JUL 2022

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Common with Information Technology)

(Effective from the admitted batch of 2015–2016)

Time : Three hours Maximum : 70 marks

First Question is compulsory.

Answer any FOUR from the remaining questions.

All questions carry equal marks.

Answer all parts of any question at one place.

1. (a) Define a compiler and list the phases of a compiler.

(b) What is meant by a transition diagram? Give an example.

(c) What are the problems with Top-down parsing?

(d) Construct a syntax tree for the arithmetic expression a * – (b+c).

(c) Define context-free grammar.

(d) List the various forms of th instructions.

(e) Define DAG and give an example.

(f) Which object code form makes the generation process easier and why?

(g) List out the contents of a symbol table.

2. (a) Explain about Cross compilers.

(b) Describe compiler construction tools.

3. (a) Construct DFA for the regular expres (11+0)*(00+1)*

(b) Explain about Lexical analysis.

4. (a) Explain Shift-reduce parsing with example.

(b) Discuss about Recursive Descent Parsing.

5. (a) Translate the arithmetic expressi a:=b*–c+b*–c into

(i) Syntax tree ii) Postfix notati (iii) Three-address code.

(b) Discuss about Type Checking and Ty conversions.

6. (a) Discuss basic blocks and flow graphs with example.

(b) Explain machine dependent cod optimization techniques.

(a) Discuss various strategies used in register allocation and assignment.

(b) Explain about code generators.

(a) Describe various approaches to organize a symbol table.

(b) Explain about Error Handling Routines.

28/8/2021

[07 BENG – 3209] AUG 2021

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science Engineering

COMPLIER DESIGN

(Common with Information Technology)

(Effective from the admitted batch of 2015–2016)

Time : Three hours Maximum : 70 marks

First question is compulsory.

Answer any FOUR from the remaining questions.

All questions carries equal marks.

Answer all parts of any question at one place.

1. Answer the following in brief :

(a) Finite automata.

(b) LR (K) parsing.

(c) Type conversion and type checking.

(d) Local optimization.

(e) Loop inversion.

(f) Lexical phase errors.

(g) Code generator.

(c) Define context-free grammar.

(d) List the various forms of three instructions.

2. (a) Explain different phases of co showing the output of each phase, usi example of the following statement.

Position : = initial + rate * 60

(b) What is ambiguious grammar? Elimina ambiguities for the grammar

$E \rightarrow E+E \mid E*E \mid (E) \mid id$

3. (a) Construct SLR parsing table for following grammar

$S \rightarrow AS \mid b$

$A \rightarrow SA \mid a$

(b) What is recursive descent parser? Consi recursive descent parser for the follow grammar.

$E \rightarrow E+T \mid T$

$T \rightarrow TF \mid F$

$F \rightarrow F^* \mid a \mid b$

4. (a) Write the quadruple, triple, indirect tri for the statement $a := b^* - c + b^* - c$.

(b) Explain the role of intermediate co generator in compilation process.

5. (a) Describe various register allocati techniques.

(b) What is code optimization? What are i advantages? Discuss the problems i optimizing the code.

statement? Explain with suitable examples.

5. (a) Discuss about allocation strategies of Heap allocation and stack allocation.

(b) What is Activation record? Explain the purpose of different fields in an activation record.

MAR 2021

TION.

gy)

: 70 m

ining

lace. ×2 =

unsig

(a) What are the various attributes of symbol table? Discuss about ordered and unordered symbol tables.

(b) What is the function of symbol table in compilation process? Explain.

(a) Explain about dataflow analysis of structured programs.

(b) Describe about Peep-Hole optimization and machine dependent optimization techniques.

Write short notes on the following :

(a) Brute force parsing

(b) 3-Address code

(c) Code generation from DAG.

(d) List the various forms of three addi instructions. lock? Give an example.

[07 BENG – 3209]

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science Engineering

COMPILER DESIGN

(Common with Information Technology)

(w.e.f. 2015–16 A.B.)

Time : Three hours Maximum : 70 marks

Q. No. 1 which is compulsory.

Answer any FOUR questions from the remaining.

Answer ALL parts of any question at one place.

$(7 \times 2 = 14)$

1. (a) Define bootstrapping.
   (b) Design finite automata for an unsigned number.
   (c) Define context-free grammar.
   (d) List the various forms of three address instructions.
   (e) What is a basic block? Give an example.
   (f) Differentiate register allocation and register assignment.
   (g) List the common semantic errors.

$(4 \times 14 = 56)$

2. (a) Explain about the structure of a compiler with a neat diagram. (8)

   (b) Discuss in detail about compiler construction tools. (6)

3. (a) What are the approaches to design lexical analyzers? Explain. (8)

   (b) What is regular definition? Give a regular definition for Pascal identifier and also design finite automata for the same. (6)

4. (a) What is brute force parsing? Explain with an example. (6)

   (b) Construct SLR parsing table for the following grammar. (8)

   $S \rightarrow Aa \mid bAc \mid dc \mid bda$

   $A \rightarrow d$

5. (a) Write a syntax directed translation scheme that converts infix expression to the corresponding postfix expression. Explain with an example. (7)

   (b) Write an attributed grammar (SDD) for constructing syntax trees for simple expressions and also construct the syntax tree for the following expressions using the SDD $a + (b * c)$. (7)

6. (a) What are the applications of DAG? Enumerate the process of local optimization. (7)

   (b) Discuss in detail about Dead Code Elimination. (7)

7. (a) Briefly discuss about Simple Target Machine Model. (7)

   (b) What are the issues in the design of code generator? Explain. (7)

8. (a) Discuss in detail about the implementation of a simple stack allocation. (7)

   (b) Explain about Block Structured Languages. (7)

[07 BENG – 3209]

III/IV. B. Tech. DEGREE EXAMINATION.

Second Semester

Computer Science Engineering

COMPILER DESIGN

(Common with Information Technology)

(Effective from the admitted batch of 2015–2016)

Time : Three hours Maximum : 70 marks

First question is compulsory.

Answer any FOUR from the remaining questions.

All questions carry equal marks.

Answer all part of any question at one place.

1. (a) Differentiate pattern with lexeme.

(b) What is cross compiler?

(c) Write an algorithm for eliminating left recursion from a left recursive grammar with example.

(d) Write SDD for a simple desk calculator.

(e) What is DAG? Draw DAG for the expression $a+a*(b-c)+(b-c)*d$?

(f) What is type equivalence?

(g) What is Dead code elimination?

2. (a) Give the algorithm for simulating DFA. Draw NFA for the regular expression $aa*|bb*$.

(b) (i) Write the algorithm for subset construction of a DFA from an NFA.

(ii) Write the algorithm for computing E – closure (T) for any set of NFA states. T.

3. Write an algorithm for LR–parsing? Construct LR parsing table for the grammar.

(a) $E \rightarrow E+T$

(b) $E \rightarrow T$

(c) $T \rightarrow T*F$

(d) $T \rightarrow F$

(e) $F \rightarrow (E)$

(f) $F \rightarrow id$.

4. (a) What are FIRST and FOLLOW functions? State the rules to construct FIRST and FOLLOW sets.

(b) Construct predictive parsing table for the following grammar :

$S \rightarrow iEtss'|a$

$S' \rightarrow es|\varepsilon$

$E \rightarrow b.$

5. (a) Explain about S–attributed definitions and L–attributed definitions.

(b) What is Heap management? What is the role of Heap management in implementing a compiler?

6. What is type conversion? Explain the methods used for type conversion with pseudo code for each method.

7. (a) How can you partition the Three–Address instructions into basic blocks? Construct intermediate code to set a 10×10 matrix to an identify matrix.

(b) Explain the method of register allocation by using Graph coloring.

8. Discuss about Simple Code Generator.

[07 BENG – 3209]

III/IV B.Tech. DEGREE EXAMINATION

Second Semester

Computer Science Engineering

COMPILER DESIGN

(Common with Information Technology)

(Effective from the admitted batch of 2015–2016)

Time : Three hours Maximum : 70 marks

First question is compulsory.

Answer any FOUR from the remaining questions.

All questions carry equal marks.

Answer ONE question from each Unit.

Answer all parts of any questions at one place.

1. (a) Give the regular definition for unsigned numbers.

   (b) List compiler construction tools.

   (c) Write an algorithm for left factoring a grammar with example.

   (d) What is Annotated parse tree? Draw Annotated parse tree for the expression (2 + 3)*5 with suitable grammar.

(e) What is Three – Address Code? Draw DAG and its corresponding Three-Address code for the expression $a+a*(b-c)+(b-c)*d$...

(f) What is type checking? Discuss their types in brief.

(g) What is Copy Propagation?

2. (a) What is DFA? Construct DFA for the given regular expression $(a|b)^* abb$.

(b) Write notes on the Lexical – Analyzer Generator Lex.

3. (a) Construct predictive parsing table for the following grammar :

$S \rightarrow E$
$E \rightarrow TE'$
$E' \rightarrow +E' | -E' | \in$
$T \rightarrow FT'$
$T' \rightarrow *T' | T' | \in$
$F \rightarrow num | id.$

(b) Construct canonical LR(O) item set for the following grammar.

$S \rightarrow L = R | R$
$L \rightarrow *R | id$
$R \rightarrow L.$

4. Give the algorithm for LALR parsing table. Describe state diagram and construct parsing table for the grammar :

$S \rightarrow CC$
$C \rightarrow cC$
$C \rightarrow d.$

5. (a) What is the role of Dependency graphs in evaluating SDDs?

(b) Draw and explain the annotated parse tree for the expression 3* 5, and its associated Dependency grapy for the grammar given below :

$T \rightarrow FT'$
$T' \rightarrow *FT'$
$T' \rightarrow \varepsilon$
$F \rightarrow digit.$

6. Explain about various methods of implementing Three – Address instructions.

7. (a) What is Peep-Hole optimization? Explain.

(b) How Branch Optimization is achieved in the final machine code of program fragment?

4. (a) What are FIRST and FOLLOW functions? State the rules to construct FIRST and FOLLOW sets.

8. (a) What are semantics – preserving Transformations in optimizing a code? Discuss with quick sort method.

(b) How global common sub expressions are eliminated in optimizing a code? Discuss with quick sort method.

3rd year 2nd sem

[07 BENG – 3209]

III/IV. B. Tech. DEGREE EXAMINATION.

Second Semester

Computer Science Engineering

COMPILER DESIGN

(Common with Information Technology)

(Effective from the admitted batch of 2015–2016)

Time : Three hours Maximum : 70 marks

First question is compulsory.

Answer any FOUR from the remaining questions.

All questions carry equal marks.

Answer all part of any question at one place.

1. (a) Differentiate pattern with lexeme.

   (b) What is cross compiler?

   (c) Write an algorithm for eliminating left recursion from a left recursive grammar with example.

   (d) Write SDD for a simple desk calculator.

(e) What is DAG? Draw DAG for the expression $a+a*(b-c)+(b-c)*d$?

(f) What is type equivalence?

(g) What is Dead code elimination?

2. (a) Give the algorithm for simulating DFA. Draw NFA for the regular expression $aa*|bb*$.

(b) (i) Write the algorithm for subset construction of a DFA from an NFA.

(ii) Write the algorithm for computing E – closure (T) for any set of NFA states. T.

3. Write an algorithm for LR–parsing? Construct LR parsing table for the grammar.

(a) $E \rightarrow E+T$

(b) $E \rightarrow T$

(c) $T \rightarrow T*F$

(d) $T \rightarrow F$

(e) $F \rightarrow (E)$

(f) $F \rightarrow id$.

4. (a) What are FIRST and FOLLOW functions? State the rules to construct FIRST and FOLLOW sets.

(b) Construct predictive parsing table for the following grammar :

$S \rightarrow iEtss'|a$

$S' \rightarrow es|\varepsilon$

$E \rightarrow b.$

5. (a) Explain about S–attributed definitions and L–attributed definitions.

(b) What is Heap management? What is the role of Heap management in implementing a compiler?

6. What is type conversion? Explain the methods used for type conversion with pseudo code for each method.

7. (a) How can you partition the Three–Address instructions into basic blocks? Construct intermediate code to set a 10×10 matrix to an identify matrix.

(b) Explain the method of register allocation by using Graph coloring.

8. Discuss about Simple Code Generator.

[07 BENG – 3209] (C-19)

III/IV B.Tech. DEGREE EXAMINATION.

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Common with Information Technology)

(Effective from the admitted batch of 2015–2016)

(For the Academic Year 2020-2021 batch only)

Time : Three hours Maximum : 70 marks

First Question is compulsory

Answer any FOUR questions from the remaining.

Write all parts of any question at one place.

1. (a) Define bootstrapping of a compiler.

   (b) How does lexical analyzer help in the process of compilation?

   (c) Give the structure of LEX program.

   (d) Define left recursion. How to eliminate left recursion from CFG?

(e) Show that the grammar G : S → SS | aSb | bsa | ε is ambiguous.

(f) Define syntax directed translation.

(g) Define loop invariant. Give an example.

2. (a) What are different analysis phases of compiler? Explain the reasons for separation of lexical analysis from syntax analysis.

(b) Describe compiler construction tools with example.

3. (a) Construct DFA for the regular expression: $(0 + 1)^* 011$.

(b) Construct finite automata that accepts tokens: identifiers, decimal constants and integer constants.

4. (a) What is input buffering? Describe different input buffering schemes.

(b) Write a procedure to compute FIRST and FOLLOW of the grammar.

5. Construct predictive parsing table for the following grammar and verify the string (a + a) is accepting or not

$E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F \quad F \rightarrow (E) \mid a$

6. (a) What are the basic operations in Shift reduce parser? Find the shift reduce parser algorithm for the input string (a, (a, a)) using following grammar:

$S \rightarrow (L) \mid a, \quad L \rightarrow L, S \mid S$

(b) Write a procedure to construct parsing table in SLR parser.

7. (a) Explain about s–attributes and l– attributes with suitable example.

(b) Translate the expression: x = – (a + b) * (c + d) + (a + b + c) into (i) Quadruple (ii) Triple (iii) Indirect triple.

8. (a) What is Peephole optimization? Explain different Peep – hole optimization methods.

(b) What is a leader of basic block? Write an algorithm to find leaders.

[07 BENG – 3212]

III/IV B.Tech. DEGREE EXAMINATION

Second Semester

Computer Science and Engineering

COMPILER DESIGN

(Effective from the admitted batch of 2019–2020)

Time : Three hours Maximum : 70 marks

First question is compulsory.

Answer any FOUR from the remaining questions.

All questions carry equal marks.

Answer ALL parts of any question at one place.

1. Answer the following in brief :
   (a) List and explain few compiler construction tools.
   (b) Define the term 'Tokens' in lexical analysis phase.
   (c) What are the grass of error handless in a parser?
   (d) Write the advantages and disadvantages of operator precedence parsing.
   (e) How can you generate 3-Address code?

(f) List various forms of Target programs.

(g) What are the patterns used for code optimization?

2. (a) Explain Single pass and multipass compiler with examples.

(b) Describe how various phases could be combined as a pass in a compiler.

3. (a) Explain Shift-reduce parsing. What are the conflicts that may occur during shift-reduce parsing?

(b) For the grammar given below, calculate the operator precedence relation and precedence functions.

$$E \rightarrow E+E \mid E-E \mid E*E \mid E/E \mid E\ E \mid (E) \mid -E \mid id$$

4. Explain 3-addres codes and mention its types. How would you implement the three-address statement? Explain with suitable examples.

5. (a) Discuss about allocation strategies of Heap allocation and stack allocation.

(b) What is Activation record? Explain the purpose of different fields in an activation record.

6. (a) How to trace the data flow analysis of structured program? Explain.

(b) Explain about Peephole optimization technique.

7. (a) Explain the concept of Syntax directed translation with an example.

(b) Explain the syntax directed translation of switch statement.

8. Write short notes on the following :

(a) Code generation algorithm

(b) Error handling routines in compliers.

(c) Machine dependent optimization.
