MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

B.Tech CSE III Year I Semester Examinations

COMPILER DESIGN (A50514)

Time : 3.00 Hours **MODEL PAPER -1** **Max.Marks: 70**

Answer any 5 questions (5x14 = 70 Marks)

Section -1

1. Discuss the phases of a compiler indicating the inputs and outputs of each phase in translating the statement "amount = principle + rate * 36.0 " [14M]

OR

2. Define an LL(1) grammar. Is the following grammar LL(1). G:S → iEtS | iEtSes |a , E → b. Also write the rules for computing FIRST() and FOLLOW(). [14M]

Section -2

3. What is an LALR(1) grammar?. Construct LALR parsing table for the following grammar: S→ CC, C → cC , C → c|d . [14 M]

OR

4. Explain the usage of YACC parser generator in construction of a Parser. [14M]

Section -3

5. What are different intermediate code forms? Discuss different Three Address code types and implementations of Three Address statements. [14 M]

OR

6. With a neat diagram explain the format of the Symbol Table. And discuss the tree structures representation of scope information. [14 M]

Section -4

7. Explain the following code optimization techniques with examples . [5M+4M+5M]

a) Constant propagation b) Strength reduction c) Code Motion

8. a) What is an induction variable? Explain with an example. [7 M]

b) Discuss how induction variables can be detected and eliminated from the given intermediate code [7 M]

```
B2:   i:= i+1
      t1:=4*j
      t2:=a[t1]
      if t2<10 goto B2
```

Section -5

9. Explain various issues in the design of the code generation. [14M]

OR

10. a). Explain the code generation algorithm in detail. [7M]

b). Write short notes on peephole optimization. [7M]

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

B.Tech CSE III Year I Semester Examinations

COMPILER DESIGN (A50514)

Time : 3.00 Hours **MODEL PAPER -2** **Max.Marks: 70**

Answer any Five questions (5x14 = 70 Marks)

Section -1

1. List out the functions of a Lexical Analyzer? State the reasons for the Separation of Analyses programs into Lexical, Syntax, and Semantic Analyses. [14 M]

OR

2. Define a Parser. What is the role of grammars in Parser construction? Construct the Predictive parsing table for the grammar G : E → E+T |T , E →T*F |F , F →(E) |id. [14M]

Section -2

3. What is an LR(0) item? Construct an SLR parsing table for the grammar G: S→ L=R |R, L → *R | id, R → L. Is it SLR(1) grammar? [14 M]

OR

4. Construct SLR parsing table for the following grammar: R-> R'|'R|RR|R*|(R)|al b [14 M]

Section -3

5. What do you mean by attributed grammars? Discuss the a translation scheme for Converting an infix expression to its equivalent postfix form. [14 M]

OR

6.a) Define activation records.

b)Explain how it is related with runtime storage allocation. [7 M+7M]

Section -4

7. What is the role of code Optimizer in compiler? Is it a mandatory phase? [14 M]

8. What is DAG and flow graph? Explain their role in compilation process. [14 M]

Section -5

9 a) Explain why next-use information is required for generating object code? [7M+7M]

b) Explain the main issues in code generation.

OR

10. a) Define code generation . [4 M+10M]

b) Generate code for the following C program using any code generation algorithm.

```
main()
{
Int I;
Int a[10];
while(i<=10)
a[i]=0;
}
```

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

B.Tech CSE III Year I Semester Examinations

COMPILER DESIGN (A50514)

Time : 3.00 Hours **MODEL PAPER -3** **Max.Marks: 70**

Answer any Five questions (5x14 = 70 Marks)

Section -1

1.a)What is LEX? [4M+10M]

b)Discuss the usage of LEX in Lexical Analyzer generation.

OR

2.a) Write a note on the parse generator '_ YACC. [7M+7M]

b) Write the YACC specification of a simple desk calculator as given:
E->E+T/T T->T*F /F F -> (E) | digit where digit between 0 to 9.

Section -2

3. a) Obtain the directed acycJic graph for the expression : x+x*(y+ z)+(y+z)* w [7M+7M]
.

b) Explain the following with example: i) Quadqaples ii) Triples iii) Indirect triple

OR

4. Compare and contrast SLR with LALR. Define Kernel items and Non-kernel items. [14 M]

Show the following grammar is LALR(1)

S-> Aa | bAc | dc | bda

A-> d

Section -3

5. Generate the three address code for the following code fragment. [14 M]

a = b + 1 x = y + 3 y = a / b a = b + c

OR

6. Write a note on simple type checker and list the different types of type checking [14 M]

Section -4

7. What is the role of code Optimizer in compiler? Is it a mandatory phase? [14 M]

OR

8. a) Construct a DAG for the expression: a+a*(b-c)+(b-c)*d [7M+7M]

b) Explain various machine independent code optimization techniques.

Section -5

9. a) Discuss Global Register Allocation in code generation. [7 M+7M]

b) Generate code for the following C statements: i) x=f(a)+f(a) ii) y=x/5;.

OR

10. Consider the following basic block of 3-address instructions: [14M]

a:=b+c x:=a+b b:=a-d c:=b+c d=a-d y=a-d

Write the next-use information for each line of the basic block.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

B.Tech CSE III Year I Semester Examinations

COMPILER DESIGN (A50514)

Time : 3.00 Hours **MODEL PAPER -4** **Max.Marks: 70**

Answer any Five questions (5x14 = 70 Marks)

Section -1

1. What is LEX? Discuss the usage of LEX in Lexical Analyzer generation. [14 M]

OR

2. State the various phases of a compiler and explain them in detail. [14 M]

Section -2

3. Prepare a canonical parsing table for the given grammar: [14 M]

S-> CC

C-> cC / d OR

4. Compare and contrast SLR with LALR. Define Kernel items and Non-kernel items. [14 M]

Show the following grammar is LALR(1)

S-> Aa | bAc | dc | bda

A-> d

Section -3

5. Generate the three address code for the following code fragment. [14 M]

while(a>b)

{

if(c<d)

x=y

+z;

else

x=y

-z;

}

OR

6. Explain the use of symbol table in compilation process. List out the various attributes for implementing the symbol table. [14 M]

Section -4

7. Explain the different storage allocation strategies. [14 M]

OR

8. Explain the role of DAG in optimization with example. [14 M]

Section -5

9. Generate code for the following: a) x=f(a)+f(a)+f(a) ii) x=f(f(a))

b) i) x=++f(a) ii) x=f(a)/g(b,c) [7M+7M]

OR

10. Explain the following terms: i) Register Descriptor ii) Address Descriptor iii) Instruction Costs [14 M]

Code No: 115AP **R13**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B.Tech III Year I Semester Examinations, February/March - 2016
COMPILER DESIGN
(Computer Science and Engineering)

Time: 3 hours **Max. Marks: 75**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

Part- A

(25 Marks)

1.a) Define Boot strapping. [2]
b) What is Context free grammar? [3]
c) Ware the actions performed by Shift reduce parser? [2]
d) Describe in brief about types of LR parsers? [3]
e) What is type expression? [2]
f) Define Type Equivalence? [3]
g) Define Basic Block? [2]
h) How can you identify the leader in a Basic block? [3]
i) Which graph is used for identifying the common sub expression in an expression? [2]
j) What is meant register allocation and assignment? [3]

Part-B

(50 Marks)

2.a) Define Regular Expression? Explain about the Properties of Regular Expressions.
b) Differentiate between Top down and bottom up parsing techniques. [5+5]

OR

3.a) Define Compiler? Explain in brief about the syntax and semantic analysis of a compiler with an example?
b) Construct a Predictive parsing table for the Grammar [5+5]
E→E+T/T, T→T*F/F, F→(E)/id .

4.a) Construct SLR parsing table for the following grammar.
E → E+T/T T → T*F/F F →(E)/id .
b) Discuss in brief about Yacc. [5+5]

OR

5.a) Construct CLR Parsing table for the grammar S→L=R/R, L→*R/id, R→L
b) Define Ambiguous Grammar? Check whether the grammar S→aAB [5+5]
A→bC/cd, C→cd, B→c/d Is Ambiguous or not.

6.a) Explain in detail about Polymorphism.
b) Explain in brief about Heap Storage allocation strategy. [5+5]

OR

7.a) Construct an annotated parse-tree for 6 3 2.
b) Explain in brief about equivalence of type expressions. [5+5]

8.a) Explain in brief about different Principal sources of optimization techniques with suitable examples.

b) Define Flow Graph? Explain how a given program can be converted in to flow graph. [5+5]

OR

9.a) What is DAG? Construct DAG for the following Basic block.

D: = B*C; E:= A+B; B:=B+C; A:=E-D;

b) Explain how copy propagation can be done using data flow equation. [5+5]

10. Explain in detail the procedure that eliminates global common sub expression. [10]

OR

11.a) What are the object code forms? Explain the issues in code generation.

b) Explain about machine dependent code optimization. [5+5]

--ooOoo--

Code No: 115AP **R13**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech III Year I Semester Examinations, March - 2017
COMPILER DESIGN
(Computer Science and Engineering)

Time: 3 hours **Max. Marks: 75**

Note: This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART – A

(25 Marks)

1.a) Write regular expression over alphabet {a, b, c} containing at least one 'a' and at least one 'b' [2]
b) What is input buffering? How is input buffering implemented? [3]
c) What is operator precedence grammar? Give an example. [2]
d) What is significance of lookahead operator in LR parsing? [3]
e) What is the s – attributes and l – attributes? [2]
f) What is activation record? [3]
g) What is dead code elimination and reduction in strength? [2]
h) Define loop unrolling. Give an example. [3]
i) What is meant by register descriptor and address descriptor? [2]
j) How to allocate registers to instruction? [3]

PART – B

(50 Marks)

2.a) Explain the concept of bootstrapping with example.
b) Consider the following Conditional statement:
if (x > 3) then y = 5 else y = 10;
How does lexical analyzer help the above statement in process of compilation? [4+6]

OR

3. Construct predictive parsing table for the following grammar [10]
S → (L) | a
L → L, S | S

4. Find the LR (0) set of items for the following grammar. Describe state diagram and construct parse table of that [10]
S → CC
C → cC |d

OR

5.a) Write a procedure to construct LALR parsing table.
b) Write short notes on YACC. [5+5]

6. What is symbol table? Discuss various ways to organizing symbol table. [10]

OR

7. Translate the following expression:
(a + b) * (c + d) + (a + b + c) into
a) Quadruples b) Triples c) Indirect triples [3+3+4]

8.a) What is liveness? Explain liveness with suitable example.
b) Write a procedure to identify basic blocks. [5+5]

OR

9. Illustrate loop optimization with suitable example. [10]

10. Explain various method to handle peephole optimization. [10]

OR

11. Generate the code for the following expression: x = (a + b) – ((c + d) – e). Also Compute its cost. [10]

Code No: 115AP **R13**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech III Year I Semester Examinations, May/June - 2019
COMPILER DESIGN
(Computer Science and Engineering)

Time: 3 hours **Max. Marks: 75**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART - A

(25 Marks)

1.a)	Why lexical and syntax analyzers are separated?	[2]
b)	List the various error recovery strategies for a lexical analysis.	[3]
c)	Mention the types of LR parser.	[2]
d)	Define LR(0) items with examples.	[3]
e)	What are the benefits of intermediate code generation?	[2]
f)	Explain about hashing.	[3]
g)	What is a basic block?	[2]
h)	Discuss about common sub expression elimination.	[3]
i)	How do you calculate the cost of an instruction?	[2]
j)	List out the common issues in the design of code generator.	[3]

PART - B

(50 Marks)

2. Explain the various phases of a compiler in detail. Also write down the output for the following expression after each phase a: =b*cd. [10]

OR

3. What is FIRST and FOLLOW? Explain the steps to compute FIRST and FOLLOW with an example. [10]

4. Check whether the following grammar is SLR (1) or not. Explain your answer with Reasons. [10]
S→ L=R S→R L→*R L→id R→L.

OR

5. Consider the grammar.
E → E + T E → T T → T * F T → F F → (E) / id
Construct CLR parsing table for the above grammar. Give the moves of the CLR parser on id * id + id. [10]

6. What is a three address code? Mention its types. How would you implement the three address statements? Explain with examples. [10]

OR

7. Describe in detail the syntax directed translation of case statements. [10]

8. What are steps needed to compute the next use information? [10]

OR

9. Discuss about the following:
   a) Copy Propagation
   b) Dead code Elimination and
   c) Code motion. [10]

10. Write the code generation for the d:=(a-b)+(a-c)+(a-c). [10]

OR

11. Write a code generation algorithm. Explain about the descriptor and function getreg(). Give an example. [10]

---ooOoo---

Code No: 115AP **R13**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech III Year I Semester Examinations, November - 2015
COMPILER DESIGN
(Computer Science and Engineering)

Time: 3 hours **Max. Marks: 75**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART - A (25 Marks)

1.a)	Define Cross Compiler.	[2]
b)	Eliminate immediate left recursion for the following grammar: E->E+T | T T->T* F | F F-> (E) | id	[3]
c)	List the rules for computing FOLLOW SET.	[2]
d)	Define CLOSURE (I).	[3]
e)	What is a symbol table?	[2]
f)	What does a semantic analysis do?	[3]
g)	Define basic block in a flow graph.	[2]
h)	What is a DAG? Mention its applications	[3]
i)	Generate a object code for following statements a = b + c; d = a+e	[2]
j)	Mention the properties that a code generator should possess.	[3]

PART - B (50 Marks)

2. What are the various phases of the compiler? Explain each phase in detail. [10]

OR

3. Construct the predictive parser for the following grammar: [10]
S->(L)/a
L->L,S/S

4. Find the SLR parsing table for the given grammar:
E->E+E | E*E | (E) | id.
And parse the sentence (a+b)*c. [10]

OR

5. Construct an LALR Parsing table for the following grammar: [10]
E-> E+T |T
T-> T*F|F
F->id

6. Generate intermediate code for the following code segment along with the required syntax directed translation scheme:
if(a>b)
x=a+b
else
x=a-b
Where a and x are of real and b of int type data. [10]

OR

7. Give syntax directed translation scheme for simple desk circulator. [10]

8. Explain the following with an example:
a) Redundant sub expression elimination
b) Frequency reduction
c) Copy propagation. [3+3+4]

OR

9. Optimize the following code using various optimization techniques: [10]
i=1; s=0;
for (i=1; i<=3; i++)
for (j=1;j<=3;j++)
c[i][j]=c[i][j] + a[i][j] + b[i][j]

10. Explain in detail about machine dependent code optimization techniques. [10]

OR

11. Give an example to show how DAG is used for register allocation. [10]

–--ooOoo---

Code No: 115AP **R13**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech III Year I Semester Examinations, November/December - 2018
COMPILER DESIGN
(Computer Science and Engineering)

Time: 3 hours **Max. Marks: 75**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART - A

(25 Marks)

1.a)	Define Boot strapping?	[2]
b)	What are the draw backs of predictive parsing?	[3]
c)	What are the actions performed by Shift reduce parser?	[2]
d)	Differentiate between SLR, LALR and CLR parsers.	[3]
e)	What are the applications of DAG?	[2]
f)	What are the advantages of stack storage allocation strategy?	[3]
g)	Define Basic block. What are the rules for defining a basic block?	[2]
h)	What is common sub expression elimination?	[3]
i)	Define Dead code elimination?	[2]
j)	What is register allocation? Give a brief description.	[3]

PART - B

(50 Marks)

2.a) Define Regular Expression. Explain about the Properties of Regular Expressions.
b) Differentiate between Top down and bottom up parsing techniques. [5+5]

OR

3.a) Define Compiler? Explain in brief about the types of lexical errors with example.
b) Construct a Predictive parsing table for the Grammar [4+6]
E→E+T/T, T→T*F/F, F→ (E)/id?

4.a) Construct LALR Parsing table for the grammar
S→L=R/R
L→*R/id
R→L
b) Define Ambiguous Grammar? Check whether the grammar
S→aAB,
A→bC/cd,
C→cd,
B→c/d Is Ambiguous or not? [6+4]

OR

5.a) Construct SLR Parsing table for the grammar
S→ (L)|a
L→L,s|s

b) Discuss in brief about model of LR parser. [6+4]

6.a) Construct a Quadruple, Triple and Indirect Triple for the statement a+a*(b-c) +(b-c)*d?

b) Differentiate between Static and Dynamic Storage allocation Strategies. [5+5]

OR

7.a) Construct an annotated parse tree for real id1, id2, id3.

b) Explain in brief about equivalence of type expressions. [5+5]

8.a) Explain in brief about different Principal sources of optimization techniques with suitable examples.

b) Define Flow Graph? Explain how a given program can be converted into flow graph? [5+5]

OR

9.a) Explain in brief about function preserving transformations on basic blocks.

b) Explain in brief about optimization of basic blocks. [5+5]

10.a) Explain in detail about global common sub expression elimination technique.

b) Explain in brief about Induction variable elimination. [5+5]

OR

11.a) Explain in brief about the issues in the design of code generator.

b) Explain in detail about peep hole optimization. [5+5]

---oo0oo---

Code No: 115AP **R13**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech III Year I Semester Examinations, November/December - 2016
COMPILER DESIGN
(Computer Science and Engineering)

Time: 3 hours **Max. Marks: 75**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART - A

(25 Marks)

1.a) Write a brief note on bootstrap process. [2]
b) What are the differences between a compiler and an interpreter? [3]
c) Give the specification of the YACC parser generator. [2]
d) Construct the LR(0) items for the "dangling-else" grammar. [3]
e) How to check structural equivalence of two type expressions? [2]
f) Define and write the differences between synthesized attributes and inherited attributes. [3]
g) Write a short note on Flow graph. [2]
h) Write an algorithm for constructing a basic block. [3]
i) Define various possible outputs of the code generator. [2]
j) Construct DAG for the following basic block: [3]

T1=A+B
T2=C+D
T3=E – T2
T4=T1– T3

PART - B

(50 Marks)

2.a) Explain various error recovery strategies in lexical analysis.
b) Construct a Finite Automata and Scanning algorithm for recognizing identifiers, numerical constants in C language. [5+5]

OR

3. Explain the various phases of a compiler with an illustrative example. [10]

4. Construct the LR Parsing table for the following grammar: [10]
$E \rightarrow E + T \mid T$
$T \rightarrow T * F \mid F$
$F \rightarrow (E)/id$

OR

5.a) Write a YACC program that will take regular expression as input and produce its parse tree as output.
b) Write an algorithm for computing LR(k) item sets. [5+5]

6.a) Write an SDT to convert infix to postfix expression.
b) Explain briefly about polymorphic functions. [5+5]

OR

7. Explain various storage allocation strategies with its merits and demerits. [10]

8. Discuss various techniques of function preserving transformations for code optimization. [10]

OR

9. Explain how data flow equations are set up and solved for improving code. [10]

10. Explain the following peephole optimization techniques: [5+5]
a) Elimination of Redundant Code
b) Elimination of Unreachable Code.

OR

11. Explain in detail about machine dependent code optimization techniques with their drawbacks. [10]

---ooOoo---