Compiler Design VTU CBCS Question Paper Set 2018

USN □□□□□□□□□□ **10CS63**

Sixth Semester B.E. Degree Examination, Dec. 2013/Jan. 2014

Complier Design

Time: 3 hrs. Max. Marks:100

Note: *Answer FIVE full questions, selecting atleast TWO questions from each part.*

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – A

1 a. Explain the various phases of complier. Show the translations for an assignment statement. Position = initial + rate * 60, clearly indicate the output of each phase. **(12 Marks)**

b. Write the regular definition for an unsigned number. Also write the transition diagram. **(06 Marks)**

c. What is printed by the following C code?

```
# define a (x + 1)
  int x = 2 ;
  void b( ) {int x  =1;printf("%d ln", a)'}
  void c( ){printf("%d ln", a); }
  void main( ) {b( ) ; c( );}.
```

(02 Marks)

2 a. Describe an algorithm used for eliminating the left recursion. Eliminate left recursion from the grammar :

S → Aa |b A → Ac | Sd | a. **(06 Marks)**

b. Show that the following grammar is ambiguous :

E → E + E | E * E | (E) | id. Write an equivalent unambiguous grammar for the same. **(06 Marks)**

c. What are the key problems with top down parse? Write a recursive descent parser for the grammar :

S → cAd A → ab | a. **(08 Marks)**

3 a. Given the grammer :

S → aABb
A → c | ε
B → d | ε

i) Compute FIRST and FOLLOW sets
ii) Construct the predictive parsing table
iii) Show the moves made by predictive parser on the input ; acdb. **(10 Marks)**

b. Explain with a neat diagram, the model of a table driven predictive parser. **(05 Marks)**

c. What is handle pruning? Give a bottom – up parse for the input : aaa * a++ and grammar :

S → SS + | SS * | a. **(05 Marks)**

4 a. Given the grammar :

S → CC
C → cC | d

i) Obtain the sets of canonical collection of sets of valid LR(0) items
ii) Design SLR parsing table. **(10 Marks)**

b. Write an algorithm used to compute LR (1) sets of items. **(06 Marks)**

c. Write a note on the parser Generator – Yacc. **(04 Marks)**

PART – B

5 a. Explain the concept of syntax – directed definition. (05 Marks)

b. The SDD to translate binary integer number into decimal is shown below :

Productions	Semantic rules
BN → L	BN.val= L. val
L → L_1 B	L. val =2 ×L_1.val + B. val
L → B	L. val = B .val
B → 0	B. val = 0
B → 1	B. val = 1

Construct the parse tree and annotated parse tree for the input string : 11001. (05 Marks)

c. Give a SDT for desktop calculator and show its parser stack implementation. (10 Marks)

6 a. Translate the arithmetic expression : a + - (b + c) into quadruples, triples and indirect triples. (06 Marks)

b. Give a semantic action for : S → if (B) S_1 else S_2. (06 Marks)

c. Develop SDD to produce directed a cyclic graph for an expression. Show the steps for constructing the directed acyclic graph for the expression : a + a * (b – c) + (b – c) * d. (08 Marks)

7 a. Describe the general structure of an activation record. Explain the purpose of each field in the activation record. (08 Marks)

b. A C – code to compute Fibonacci numbers recursively is shown below :

```
int f(int n)
 { int t, s ;
   if(n < = 2) return 1 ;
    s = f(n -1) ;
    t = f(n -2) ;
    return (s+t);
 }
```

i) Draw the activation tree for the call : f(5)

ii) What is the largest number of activation records that ever appear together on the stack? (06 Marks)

c. Explain the performance metrics to be considered while designing a garbage collector. (06 Marks)

8 a. Discuss the issues in the design of a code generator. (10 Marks)

b. Write the tree address code and construct the basic blocks for the following program segment.

```
sum = 0 ;
for(i = 0; i < = 10 ; i ++)
    sum = sum +a[i] ;
```

(05 Marks)

c. Give the code generation process for operations. (05 Marks)

* * * * *

USN □□□□□□□□□□ **10CS63**

Sixth Semester B.E. Degree Examination, Dec.2014/Jan.2015
Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

PART – A

1 a. Explain with neat diagram, the phases of compiler with example. **(10 Marks)**

b. Construct a transition diagram for recognizing relational operators. Sketch the program segment to implement it, showing the first state and one final state. **(10 Marks)**

2 a. Briefly explain the problems associated with top down parser. **(03 Marks)**

b. Show that following grammar is ambiguous; S → S + S ¦ S ∗ S ¦ id. Give an unambiguous grammar for the above grammar such that '+' has highest priority and ∗ has less priority and both are left associative. **(07 Marks)**

c. Given the grammar A → (A) / a

i) Construct predictive parser table.

ii) Check the grammar is LL(1) or not.

iii) Show the parser steps for the input ((a)). **(10 Marks)**

3 a. Obtain LR(0) items for the following grammar:

S → L = R ¦ R L → ∗ R ¦ id R → L. **(08 Marks)**

b. Obtain FIRST and FALLOW sets for the grammar shown in Q.3(a) and obtain SLR parsing table. Is the grammar SLR? **(12 Marks)**

4 a. Given the grammar:

A → CC C → aC ¦ b

i) Construct sets of LR(1) items.

ii) Construct canonical LR(1) parsing table. **(12 Marks)**

b. Write a note on the parse generator – YACC. **(03 Marks)**

c. Write the YACC specification of a simple desk calculator with following grammar for arithmetic expression:

E → E + T ¦ T

T → T ∗ F ¦ F

F → (E) ¦ digit where digit between 0 to 9. **(05 Marks)**

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – B

5 a. Explain type of attributes for non terminal with example. **(04 Marks)**

b. Write annotated parse tree for expression 5 + 4 * 3n where grammar is

L → En

E → E + T ¦ T

T → T * F ¦ F

F→ (E) ¦ digit **(06 Marks)**

c. How different classes of SDD's that guarantee evaluation order? **(06 Marks)**

d. Obtain postfix SDT for simple desk calculator. **(04 Marks)**

6 a. Obtain the directed acyclic graph for the expression x + x * (y + z) + (y + z) * w. **(06 Marks)**

b. Explain the following with example:

i) Quadraples ii) Triples iii) Indirect triples. **(06 Marks)**

c. Explain SDT of switch statement. **(08 Marks)**

7 a. What is activation record? Explain structure and purpose of each field in the activation record. **(06 Marks)**

b. Explain tasks of caller and callee when procedure called and exit. **(08 Marks)**

c. Explain briefly the performance metrics to be considered while designing garbage collector. **(06 Marks)**

8 a. Write intermediate code for the following source code:

```
for i from 1 to 10 do
for j from 1 to 10 do
    a [i, j] = 0.0;
for i from 1 to 10 do
    a [i, i] = 1.0;
```

and identify basic blocks. **(10 Marks)**

b. Discuss the issues in the design of a code generator. **(10 Marks)**

* * * * *

USN □□□□□□□□□□ **10IS662**

Sixth Semester B.E. Degree Examination, Dec.2014/Jan.2015

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. Describe the various phases of a compiler. Write down the translation for an assignment statement like: position = initial + rate ∗ 60. Indicate the outputs of each phase. **(12 Marks)**

b. Construct the transition diagrams for a set of keywords like begin, end, if, then and else, and identifiers and constants along with a minimum set of relational operators. **(08 Marks)**

2 a. How left recursion can be eliminated from grammars? Write down the simple arithmetic expression grammar and rewrite the grammar after removing left recursion. **(05 Marks)**

b. What is left factoring? Rewrite the following grammar after "left factored":

$S \rightarrow iEtS \mid iEtSeS \mid a$

$E \rightarrow b$ **(05 Marks)**

c. Broadly classify parsers. How top-down parsing and bottom-up parsing be subdivided? Briefly explain the merits of each. **(05 Marks)**

d. Briefly explain the implementation of a table driven predictive parser. **(05 Marks)**

3 a. Define LL(1) grammars. Test whether the following grammar is LL(1) or not, and construct a predictive parsing table for it.

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \in$

$B \rightarrow \in$ **(10 Marks)**

b. Construct the LR(0) item set for the grammar:

$E \rightarrow E + T / T$

$T \rightarrow T * F / F$

$F \rightarrow (E) / id$

Also draw the DFA for the et of LR(0) items found. **(10 Marks)**

4 a. Construct the canonical LR(1) items and the GOTO graph as well as the canonical LR(1) parsing table for the following augmented grammar:

$S' \rightarrow s$

$S \rightarrow CC$

$C \rightarrow cC \mid d$ **(16 Marks)**

b. For the table in question 4(a), construct the LALR parsing table. **(04 Marks)**

PART – B

5 a. Explain the concept of syntax-directed definitions. What is synthesized attributes and inherited attributes? **(06 Marks)**

b. Construct the syntax tree, parse tree, and the annotated parse tree for the input string say 5 ∗ 6 + 7; using suitable context-free grammar. **(06 Marks)**

c. Construct a dependency graph for the declaration float id1, id2, id3. **(08 Marks)**

6 a. Draw the syntax tree and DAG for the expression (a ∗ b) + (c – d) ∗ (a ∗ b) + b. **(08 Marks)**

b. Represent the following assignment namely a = b ∗ – c + b ∗ – c; in its syntax tree form, three-address code, quadruples and triples representation. **(12 Marks)**

7 a. Discuss the general structure of activation record. **(08 Marks)**

b. What is meant by calling sequence and return sequence? List the calling sequence design principles. **(08 Marks)**

c. Write a note on garbage collection. **(04 Marks)**

8 a. List and briefly explain the design issues of a code generator. **(10 Marks)**

b. With example explain common subexpression and dead code elimination methods. **(10 Marks)**

* * * * *

USN □□□□□□□□□□

10IS662

Sixth Semester B.E. Degree Examination, Dec.2015/Jan.2016
Compiler Design

Time: 3 hrs. Max. Marks: 100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

PART – A

1 a. With a neat diagram, explain the various phases of compiler with example. **(10 Marks)**

b. Explain the input buffering strategy used in lexical analysis phase. **(05 Marks)**

c. Write the regular definition for an unsigned number, also write the transition diagram. **(05 Marks)**

2 a. Define left recursion and left factoring? Consider the Grammar :

$E \to E + T \mid T$
$T \to id \mid id\,[\;] \mid id\,[X]$
$X \to E, E \mid E$

i) Eliminate left recursion
ii) For the obtained result of i) do the left factoring. **(10 Marks)**

b. Construct LL(1) parsing table for the grammar

$S \to aB \mid aC \mid Sd \mid Se$
$B \to bBc \mid f$
$C \to g$

and verify the above grammar is LL (1) or not. **(10 Marks)**

3 a. Define Handle and Handle pruning. Consider the grammar :

$E \to E + T \mid T$
$T \to T * F \mid F$
$F \to (E) \mid id$

Indicate the handle for the following right sentential form $id_1 * id_2$. **(06 Marks)**

b. With a neat diagram, explain the general structure of LR parser. **(06 Marks)**

c. Consider the grammar :

$S \to L = R \mid R$
$L \to *R \mid id$
$R \to L$

Verify the grammar is SLR (1) or not. **(08 Marks)**

4 a. Given the grammar :

$A \to (A) \mid a$

Construct LR(1) set of items, parsing table and also parse the input string ((a)) using canonical LR parsing method. **(12 Marks)**

b. Write the procedure used to compute LR(1) items using LALR parser. **(04 Marks)**

c. Write a note on the parser generator - Yacc. **(04 Marks)**

PART – B

5 a. Define synthesized and inherited attributes with examples. **(04 Marks)**

b. Consider the grammar that is used for simple desk calculator. Obtain the semantic action and also the annotated parse tree for the string. (3 + 4) * (5 + 6)n

L → En
E → E + T
E → T
T → T *F
T → F
F → (E)
F → digit. **(10 Marks)**

c. Consider the grammar :

$T \rightarrow F\,T^1$
$T^1 \rightarrow *\,F\,T^1$
$T^1 \rightarrow E$
F → digit

Write the semantic action and obtain the dependency graph and the order of execution for the input string 3 * 5. **(06 Marks)**

6 a. Define DAG? Develop SDD to produce DAG for the expression :

E → E + T
E → E – T
E → T
T → (E)
T → id
T → num

and show the steps for constructing DAG for the expression a + a * (b – c) + (b – c) * c. **(12 Marks)**

b. Explain the quadruples, triples and indirect triples for the example a : = b * – c + b * c. **(08 Marks)**

7 a. Describe the general structure of a activation record. Explain the purpose of each field in the activation record and construct activation tree for Quicksort. **(10 Marks)**

b. Explain Heap management in detail. **(10 Marks)**

8 a. Briefly explain the main issues in code generation. **(10 Marks)**

b. Briefly explain any five kinds of code optimization with an example each. **(10 Marks)**

* * * * *

USN □□□□□□□□□□ **10CS63**

Sixth Semester B.E. Degree Examination, Dec.2015/Jan.2016

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – A

1 a. With the help of a diagram, explain the various phases of a compiler. **(10 Marks)**

b. What is meant by input buffering? Write an algorithm for look ahead code with sentinels. **(04 Marks)**

c. Construct transition diagram to recognize the tokens below
i) identifier ii) Relational operator iii) unsigned number. **(06 Marks)**

2 a. With a neat diagram explain the role of a parser. **(05 Marks)**

b. Explain different error recovery strategies. **(08 Marks)**

c. Consider the context free grammar S → SS + |SS * | a
And the string aa + a*
i) Give a left most derivation for the string
ii) Give a right most derivation for the string
iii) Give a parse tree for the string
iv) Is the grammar ambiguous or unambiguous? Justify.
v) Describe the language generated by this grammar
vi) Remove the left recursion from the grammar?
vii) Left factor this grammar. **(07 Marks)**

3 a. Given the grammar
S → a | (L), L → L, S | S
i) Do the necessary changes to make it suitable for LL(1) parser
ii) Check the resultant grammar is LL (1) or not
iii) Show the moves made by the predictive parser on the input (a, (a, a)). **(12 Marks)**

b. What is meant by handle pruning? List the actions of a shift reduce parser. Consider the following grammar
S → TL ;
T → int | float
L → L, id | id parse the input string int id, id; using shift reduce parser. **(08 Marks)**

4 a. Given the grammar
S → AA
A → Aa | b
i) Construct sets of LR(1) items
ii) Construct canonical LR(1) parsing table **(12 Marks)**

b. How LALR parsing table is constructed? Develop an algorithm for the same. **(08 Marks)**

PART – B

5 a. Give the syntax directed definition to process a sample variable declaration in C and construct dependency graph for the input float x, y, z. **(10 Marks)**

b. Write the grammar and syntax directed definitions for a simple desk calculator and show annotated parse tree for the expression 3*5 + 4n. **(10 Marks)**

6 a. Draw the DAG for the arithmetic expression
a + a * (b – c) + (b – c)*d. Show the steps for constructing the DAG. **(10 Marks)**

b. What are three address codes? Explain different ways of representing three address codes, with examples. **(10 Marks)**

7 a. Distinguish between static scope and dynamic scope. Briefly explain access to non – local names in static scope. **(10 Marks)**

b. Explain in detail, the strategy for reducing fragmentation in heap memory. **(10 Marks)**

8 a. Discuss the following terms :
i) Basic blocks ii) Next use information iii) Flow graph. **(10 Marks)**

b. With example, explain common subexpression and dead code elimination methods. **(10 Marks)**

* * * * *

USN □□□□□□□□□□ **10IS662**

Sixth Semester B.E. Degree Examination, Dec.2016/Jan.2017

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.

PART – A

1 a. Differentiate between computer and Interpreter. List and explain the various phases of a compiler and show the output of each phase for the expression a : = b + c * 25. **(10 Marks)**

b. Explain the technique of input buffering used by the lexical analyser. **(06 Marks)**

c. Construct transition diagram for the following :

i) Unsigned numbers

ii) Relational operators. **(04 Marks)**

2 a. Write the algorithm used of eliminating left recursion. Use this algorithm and eliminate left recursion on the given grammar

S → C | a

C → Dd | c

D → Cc | d **(07 Marks)**

b. Write the "dangling else" grammar and show that it is ambiguous. Rewrite the grammar incorporating the rule "Match each else with the closest unmathed then". **(08 Marks)**

c. List and explain the various error recovery strategies. **(05 Marks)**

3 a. Construct the predictive parsing table for the given grammar

E → TE′

E′ → + E | ε

T → FT′

T′ → T | ε

F → PF′

F′ → *F′ | ε

P → (E) | a | b | ep **(10 Marks)**

b. Discuss the conflicts that can arise during shift reduce parsing giving one example for each type. Do shift reduce parse for the string (a, (a, a)) and indicate the presence of conflicts if any. Use the given grammar

S → (L) | a

L → L, S | S **(10 Marks)**

4 a. Show that the following grammar is LR(1).

S → Aa | bAc | Bc | bBa

A → d

B → d **(10 Marks)**

b. Explain the procedure for the construction of an LA LR parser. **(10 Marks)**

PART – B

5 a. Define :

i) Inherited attribute

ii) Synthesized attribute. (02 Marks)

b. Develop the grammar and SDD for a simple desk calculator and show the annotated parse tree for the expression 1 * 2 * 3* (4+5) n. (10 Marks)

c. Rewrite the actions of desk calculator SDD so that they manipulate the parser stack explicitly. Illustrate the parser stack implementation. (08 Marks)

6 a. Construct a DAG for the arithmetic expression 2* x+y * (2*x – y). Show the steps for constructing the DAG. (08 Marks)

b. Generate intermediate code for the statement "if (x < 100 || x > 200 && xi = y) x = 0", along with the required syntax directed translation scheme. Avoid redundant Gotos. (12 Marks)

7 a. Show the structure of activation record. Explain the purpose of each item on the activation record. (08 Marks)

b. Explain the strategy for reducing fragmentation in heap memory. (08 Marks)

c. List and explain the performance metrics to be considered when designing a garbage collector. (04 Marks)

8 a. Generate the code for the expression

X = (a – b) + (a + c) (04 Marks)

b. What are basic blocks? Explain the DAG representation of basic blocks. (08 Marks)

c. Describe the next use information. Write an algorithm to determine the liveness and next use information for each statement in a basic block. (08 Marks)

* * * * *

USN □□□□□□□□□□ **10CS63**

Sixth Semester B.E. Degree Examination, Dec.2017/Jan.2018
Complier Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – A

1 a. With a neat diagram, explain various phases of complier. **(10 Marks)**

b. Write the look ahead code with sentinels for input buffering strategy used in lexical analysis phase. **(05 Marks)**

c. Construct a transition diagram for recognizing unsigned numbers. **(05 Marks)**

2 a. Explain panic mode and phrase – level error recovery strategies. **(08 Marks)**

b. Write an algorithm to left factor a grammar. Give the left factored grammar for the following :

$S \rightarrow iEtS/iRtSeS/a$

$E \rightarrow b$ **(06 Marks)**

c. Give the rules for constructing FIRST and FOLLOW sets. **(06 Marks)**

3 a. Write an algorithm to construct a predictive parsing table. Construct the predictive parsing table, considering the grammar :

$E \rightarrow E + T/T$

$T \rightarrow T * F/F$

$F \rightarrow (E) /id$ **(08 Marks)**

b. Explain the working of a shift reduce parser. **(04 Marks)**

c. Explain the conflicts of shift reduce parsing with suitable examples. **(08 Marks)**

4 a. Write an algorithm for constructing SLR parsing table. **(06 Marks)**

b. Construct LALR parsing table, considering the following augmented grammar :

$S' \rightarrow S$

$S \rightarrow CC$

$C \rightarrow cC/d$ **(10 Marks)**

c. Write a note on the use of ambiguous grammars. **(04 Marks)**

PART – B

5 a. Explain the concept of syntax–directed definition. **(06 Marks)**

b. Construct a dependency graph for the declaration float id1, id2, id3. **(06 Marks)**

c. Explain the parser stack implementation of postfix SDT with an example. **(08 Marks)**

6 a. Obtain the directed acyclic graph for the expression :

$a + a * (b - c) + (b - c) * d$. **(06 Marks)**

b. List any four common three address instruction forms. **(04 Marks)**

c. Write syntax directed definition for flow of control statements. **(10 Marks)**

7 a. With a neat diagram, explain the typical subdivision of runtime memory. **(08 Marks)**

b. Explain the desirable properties of memory manager. **(06 Marks)**

c. Explain the design goals for garbage collector. **(06 Marks)**

8 a. Write an algorithm to partition three-address instructions into basic blocks. **(06 Marks)**

b. Define flow graph. How it is constructed? **(04 Marks)**

c. With an example, explain common sub-expression and dead code elimination methods. **(10 Marks)**

* * * * *

USN □□□□□□□□□□ **10IS662**

Sixth Semester B.E. Degree Examination, June/July 2013

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. Explain, with a neat diagram, the phases of a compiler. **(10 Marks)**

b. Construct the transition diagram to recognize the tokens given below:

i) Identifier

ii) Relational operator

iii) Unsigned number **(10 Marks)**

2 a. What is left-recursion? Eliminate left recursion from the following grammar:

$E \rightarrow E + T/T$

$T \rightarrow T * F/F$

$F \rightarrow (E)/id$ **(08 Marks)**

b. Given the grammar:

$S \rightarrow (L)/Q$

$L \rightarrow L, S/S$

i) Make necessary changes to make it suitable for LL (1) parsing.

ii) Construct FIRST and Follow sets

iii) Construct the predictive parsing table

iv) Show the moves made by the predictive parser on the input (a, (a, a)). **(12 Marks)**

3 a. What is shift reduce parser? Explain the conflicts that may occur during shift reduce parsing. **(04 Marks)**

b. Given the grammar:

$A \rightarrow (A)/a$

i) Find LR (0) items

ii) Construct SLR parsing table.

iii) Write SLR parsing algorithm.

iv) Show the parsing of input string ((a)). **(16 Marks)**

4 a. Given the following grammar:

$S \rightarrow CC$

$C \rightarrow cC/d$

i) Construct sets of LR (1) items.

ii) Construct canonical LR (1) parsing table. **(12 Marks)**

b. Construct LALR parsing tables for the grammar shown in Q.No.4(a) using LR (1) items. **(08 Marks)**

PART – B

5 a. Explain the concept of syntax directed translation, with examples. **(06 Marks)**

b. Define inherited and synthesized attributes. **(04 Marks)**

c. Give SDD of a simple desk calculator. **(04 Marks)**

d. Write the annotated parse tree for 3*5 + 4n. **(06 Marks)**

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

6 a. Draw the DAG for the arithmetic expression, a + a ∗ (b – c) + (b – c) ∗ d. Show the steps for constructing the DAG. **(10 Marks)**

b. Translate the arithmetic expression a + –(b + c) into quadruples, triples and indirect triples. **(06 Marks)**

c. Write the tree address code for switch statement. **(04 Marks)**

7 a. What is an activation record? Explain the purpose of each item in the activation record, with an example. **(08 Marks)**

b. What is meant by calling sequence and return sequence? List calling sequence design principles. **(08 Marks)**

c. Write a note on Garbage collection. **(04 Marks)**

8 a. List and explain design issues of a code generator. **(10 Marks)**

b. With an example, explain common sub-expression and dead code elimination methods. **(10 Marks)**

* * * * *

USN | | | | | | | | | | |

10CS63

Sixth Semester B.E. Degree Examination, June/July 2013

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – A

1 a. Explain three types of software productivity tools. **(06 Marks)**

b. Define sentinels. Give lookahead code with sentinels. **(04 Marks)**

c. Enlist algebraic laws for regular expressions. **(07 Marks)**

d. Give transition diagram for unsigned numbers. **(03 Marks)**

2 a. Write an algorithm to eliminate left recursion from a grammar, also give the syntax of the production. **(05 Marks)**

b. Consider the production:

$S \rightarrow aAb$

$A \rightarrow cd/C$.

Show that recursive-descent parsing fails for the input string "acdb", also explain recursive descent algorithm. **(07 Marks)**

c. Find First and Follow for the given grammars:

i) stmt_sequence $\rightarrow$ stmt stmt_sequence′

stmt_sequence′ $\rightarrow$; stmt_sequence/$\in$

stmt $\rightarrow$ s

ii) $S \rightarrow ,GH;$

$G \rightarrow aF$

$F \rightarrow bF/\in$

$H \rightarrow KL$

$K \rightarrow m/\in$

$L \rightarrow n/\in$ **(08 Marks)**

3 a. What are two types of conflicts during shift reduce parsing? Give examples. **(04 Marks)**

b. For the given grammar $E \rightarrow E + n/n$. Construct parsing table of LL(1). Verify 3 + 4 + 5 and show each step of verification with reference to parsing table. **(08 Marks)**

c. How to verify whether grammar is LL(1) or not? Show that:

$S \rightarrow AaAb/BbBa$

$A \rightarrow \in$

$B \rightarrow \in$

is LL (1), without constructing any table. **(08 Marks)**

4 a. Construct the DFA of LR(0) items and SLR parsing table for the grammar:

Stmt_sequence $\rightarrow$ stmt_sequence; stmt/stmt

Stmt $\rightarrow$ S

Identify Kernel and non Kernal items in state I_4. **(12 Marks)**

b. Discuss the behaviour of the LR parser. **(04 Marks)**

c. For the grammar $A \rightarrow (A)/a$, construct LR(1) set of items. **(04 Marks)**

PART – B

5 a. Write annotated parse tree for 3*5 + 4n using Top down approach. Write semantic rules for each step. **(08 Marks)**

b. Discuss S-attributes and L-attributes with respect to SDD (Syntax Directed Definition). **(04 Marks)**

c. By considering an array type int[3][3], write syntax directed translation with semantic rules. **(08 Marks)**

6 a. Enlist any four common three address instruction forms. **(04 Marks)**

b. Define quandruples, triples and static single assignment form. **(06 Marks)**

c. Write syntax directed definition for flow of control statements. **(10 Marks)**

7 a. Write a version of quick sort, in ML style using the nested functions. Give any four additional features of ML. **(08 Marks)**

b. "Most programs exhibit a high degree of locality", explain the statement. **(05 Marks)**

c. "Garbage collection is seldom used in real time applications", justify the statement. How language design affects the characteristics of memory usage. **(07 Marks)**

8 a. How register allocation and evaluation order plays an important role in a code generation? Discuss. **(06 Marks)**

b. Write an intermediate code to set a 10 × 10 matrix to an identity matrix. **(10 Marks)**

c. Define flow graph. How it is constructed? **(04 Marks)**

* * * * *

USN □□□□□□□□□□ **10CS63**

Sixth Semester B.E. Degree Examination, June / July 2014

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – A

1 a. Explain the various phases of a compiler with the help of neat diagram. **(08 Marks)**

b. Give the formal definations of operations on languages with notations. **(04 Marks)**

c. Write the transition diagram to recognize the token below:
i) relop (relational operations)
ii) unsigned number. **(08 Marks)**

2 a. Give the rules for constructing FIRST and FOLLOW sets. **(06 Marks)**

b. Construct the predictive parsing table by making necessary changes to the grammar given below:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$ **(10 Marks)**

c. Give the formal defination of CFG with an example. **(04 Marks)**

3 a. What is a shift-reduce parser? Explain the conflicts that may occur during shift-reduce parsing. List the actions of shift-reduce parser. **(06 Marks)**

b. Form the Action / Goto table for the following grammar:

$S \rightarrow Aa \mid bAc \mid Ba \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

Justify whether the grammar is LR(0) or not. **(14 Marks)**

4 a. Construct the canonical LR(1) Item sets for the following grammar:

$S \rightarrow AA$

$A \rightarrow aA \mid b$ **(10 Marks)**

b. Construct LALR parsing table for the grammar shown in Q4 (a) using LR(1) items. **(10 Marks)**

PART – B

5 a. Define inherited and synthesized attributes. Give examples. **(06 Marks)**

b. Give the SDD for simple desk calculator and draw dependency graph for expression, $1 * 2 * 3 * (4 + 5)n$ **(10 Marks)**

c. Write SDD that generates either a basic type or an array type. **(04 Marks)**

6 a. Draw the DAG for the expression, $a + a * (b - c) + (b - c) * d$. Show the steps for constructing the same. **(10 Marks)**

b. Explain the following with examples: i) Quadraples ii) Triples. **(06 Marks)**

c. Write the three address code for the expression:
$a + a * (b - c) + (b - c) * d$ **(04 Marks)**

7 a. Give the general structure of an activation record. Explain the purpose of each component. **(08 Marks)**

b. Explain the performance metrics that must be considered while designing garbage collector. **(08 Marks)**

c. Give the memory hierarchy configurations of modern computer highlighting size and access times. **(04 Marks)**

8 a. Explain the main issues in code generation. **(10 Marks)**

b. For the following program segment:

```
for i = 1 to 10 do
    for j = 1 to 10 do
        a[i, j] = 0 . 0
    for i = 1 to 10 do
    a[i, i] = 1.0
```

Generate intermediate code and identify basic blocks. **(10 Marks)**

* * * * *

USN □□□□□□□□□□ **10IS662**

Sixth Semester B.E. Degree Examination, June/July 2014

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – A

1 a. Draw diagram for language processing system. **(04 Marks)**

b. Explain general structure of a compiler. **(08 Marks)**

c. Write an algorithm for "look ahead code with sentinels". **(04 Marks)**

d. Draw a transition diagram for identifiers and keywords. How do you handle reserve words that look like identifiers? **(04 Marks)**

2 a. For the following grammar

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

Make necessary changes to make it suitable for LL(1) parser. **(02 Marks)**

b. After doing necessary changes in Q2(a) grammar check whether it is LL(1) grammar or not. If yes, parse the string (a, a). **(10 Marks)**

c. List all error recovery methods in LL(1) parser. Explain one method suitable for the grammar given below.

$S \rightarrow AbS \mid e \mid \in$

$A \rightarrow a \mid cAd$ string is "ceadb" **(08 Marks)**

3 a. What is a shift reduce parser? Explain the conflict that may occur during shift reduce parsing. **(04 Marks)**

b. What is handle pruning? Explain with the help of the grammar $S \rightarrow SS + \mid SS* \mid a$ and input string aaa*a++. **(08 Marks)**

c. Give Bottom-up parsing for the strings 000111 and grammar $S \rightarrow OS1 \mid 01$ and construct parse tree in each step of deviation. **(08 Marks)**

4 a. Write algorithm for construction of canonical LR(1) parsing table. **(10 Marks)**

b. Construct LALR parsing tables for the grammar shown below using LR(1) items.

$S \rightarrow CC$

$C \rightarrow c\,C \mid d$ **(10 Marks)**

PART – B

5 a. Write a SDD for desktop calculator. **(04 Marks)**

b. Assume suitable SDD to construct a syntax tree for the expression a – 4 + c and what are the steps involved in construction of that syntax tree. **(08 Marks)**

c. Construct annotated parse tree for 3*5 and write dependency graph for the constructed parse tree. **(08 Marks)**

6 a. List various 3 address instruction forms. Give one example for each. **(10 Marks)**

b. Construct DAG for the expression

$(x + y) - ((x + y) * (x - y)) + ((x + y) * (x - y))$ **(06 Marks)**

c. Write case 3 address code instructions used to translate a switch statement. **(04 Marks)**

7 a. Discuss about the various components and their use in an activation record. **(08 Marks)**

b. What do you mean by calling sequence? Explain the actions performed during
(i) function call (ii) return. **(08 Marks)**

c. Draw subdivision of run-time memory into code and data areas. **(04 Marks)**

8 a. For the following program segment generate intermediate code and flowgraph of that code.

```
for i from 1 to 10 do
    for j from 1 to 10 do
        a[i, j] = 0.0 ;
    for i from 1 to 10 do
        a[i, i] = 1.0 ;
```

(10 Marks)

b. What are the steps involved in optimization of Basic blocks. Explain any 2 steps. **(10 Marks)**

* * * * *

USN

10CS63

Sixth Semester B.E. Degree Examination, June/July 2015
Compiler Design

Time: 3 hrs. Max. Marks: 100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – A

1 a. Explain with a diagram, the phases of complier. **(08 Marks)**

b. Write regular definitions for the following using extended regular expression notation :
  i) identifier
  ii) unsigned number. **(06 Marks)**

c. Write a program for look ahead code with sentinels. **(06 Marks)**

2 a. Define left – recursive grammer. Eliminate left recursion from the following grammer :
$E \rightarrow E + T \mid T$
$T \rightarrow T * F \mid F$
$F \rightarrow (E) \mid id.$ **(05 Marks)**

b. Given the grammer :
$S \rightarrow AaAb \mid BbBa$
$A \rightarrow \epsilon$
$B \rightarrow \epsilon$
  i) compute FIRST( ) and FOLLOW( ) functions
  ii) construct predictive parsing table
  iii) parse the input string w = ab. **(09 Marks)**

c. Show that the following grammer is ambigious $E \rightarrow E + E \mid E * E \mid (E) \mid id$, write an equivalent un-ambigious grammer for the same. **(06 Marks)**

3 a. What is meant by handle pruning? construct Bottom – up parse tree for the input string w = aaa * a ++. Using the grammer :
$S \rightarrow SS + \mid SS * \mid a.$ **(06 Marks)**

b. Explain the working of shift reduce parser. Parse the input string id * id. Using the grammer of question no, 2(a). **(08 Marks)**

c. With a diagram, explain the model of an LR parser. **(06 Marks)**

4 a. Write an algorithm to construct LALR parsing table. **(06 Marks)**

b. Construct the parsing table for LALR(1) parser using the grammer :
$S \rightarrow CC$
$C \rightarrow aC$
$C \rightarrow d.$ **(10 Marks)**

c. Compare LALR and canonical LR parsers. **(04 Marks)**

PART – B

5 a. Explain the concept of syntax directed definition. **(04 Marks)**

b. Consider the context free grammer given below :

$S \rightarrow EN$

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T/F \mid F$

$F \rightarrow (E) \mid \text{digit}$

$N \rightarrow ;$

i) Obtain SDD for the above grammer

ii) Annotated parse tree for the input string $5 * 6 + 7$. **(10 Marks)**

c. Define :

i) Synthesized attribute

ii) Inherited attribute. **(06 Marks)**

6 a. Construct DAG and three address code for the following expression :

$a + a * (b - c) + (b - c) * d$. **(08 Marks)**

b. Explain the following with an example : i) quadruples ii) triples. **(08 Marks)**

c. Generate three address code for the following statement :

```
switch (ch)
{ case 1 : c = a + b ; break ;
  case 2 : c = a – b ; break ;
}
```

(04 Marks)

7 a. With a neat diagram, describe the general structure of an activation record. **(06 Marks)**

b. Explain in the strategy for reducing fragmentation in heap memory. **(08 Marks)**

c. Explain briefly the performance metrics to be considered while designing a garbage collector. **(06 Marks)**

8 a. Discuss the various issues in the design of a code generator. **(10 Marks)**

b. What are basic blocks and flow graphs? Write an algorithm to partition the three address instructions into basic blocks. **(06 Marks)**

c. List the characteristics of a peephole optimization. **(04 Marks)**

* * * * *

USN □□□□□□□□□□ **10IS662**

Sixth Semester B.E. Degree Examination, June/July 2015

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

PART – A

1 a. Explain the various phases of compiler. Show the translations for an assignment statement position = initial + rate * 60, clearly indicate the output of each phase. **(12 Marks)**

b. Construct transition diagram for the following: i) Relational operators; ii) Unsigned number. **(08 Marks)**

2 a. Show that the following grammar is ambiguous: E → E + E / E * E / (E) / id. Write an equivalent unambiguous grammar for the same. **(06 Marks)**

b. Write a recursive descent parser for the grammar: S → cAd A → ab/a and for the input "cad" trace the parser. **(04 Marks)**

c. Consider the grammar:

E → 5 + T / 3 – T

T → V / V * V / V + V

V → a / b

i) Do the left factoring for the above grammar.

ii) Obtain FIRST and FOLLOW table for the above grammar.

iii) Construct predictive parsing table for the above grammar. **(10 Marks)**

3 a. What is handle pruning? Explain with the help of the grammar. S → SS + / SS * / a and input string aaa * a++, give a bottom-up parse of the given input string. **(10 Marks)**

b. For the following grammar S → 0S1 / 01 indicate the handle in the following right sentential form 00001111. **(04 Marks)**

c. Show that the following grammar is not LL(1) without constructing parsing table:

S → iCtSS′ / a

S′ → eS / ∈

C → b **(06 Marks)**

4 a. Consider the following grammar

S → CC

C → cC

C → d

i) Obtain canonical collection of LR (0) items.

ii) Construct SLR (1) parsing table.

iii) Show the sequence of moves made by the parser for the string ccdd. **(12 Marks)**

b. Consider the following augmented grammar

S′ → S

S → AA

A → Aa/b

Obtain LR(1) items. **(08 Marks)**

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

PART – B

5 a. Obtain SDD for simple type declaration. Construct a dependency graph for the declaration int a, b, c along with evaluation order. **(08 Marks)**

b. For the given productions shown below, write semantic rules and construct annotated parse tree for

3 * 5 + 4n

L → En E → E1 + T E → T

T → T1 * F T → F F → (E) F → digit. **(08 Marks)**

c. Define S-attributed and L-attributed definitions with examples. **(04 Marks)**

6 a. Explain how DAG will help in intermediate code generation? Construct a DAG and a 3-address code for the expression a + a * (b – c) + (b – c) * d. **(08 Marks)**

b. Explain the following with an example:

i) Quadruples ii) Triples iii) Indirect triples. **(06 Marks)**

c. Explain syntax directed translation of switch statement. **(06 Marks)**

7 a. Describe the general structure of an activation record. Explain the purpose of each item in the activation record. **(10 Marks)**

b. What is garbage collection? Explain the design goals of garbage collector. **(10 Marks)**

8 a. Briefly discuss the various issues in code generation phase. **(10 Marks)**

b. Explain the following code optimization with examples:

i) Finding local common sub expression.

ii) Dead code elimination. **(10 Marks)**

* * * * *

USN □□□□□□□□□□

10IS662

Sixth Semester B.E. Degree Examination, June/July 2016

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. With a neat diagram, explain various phases of compiler. **(10 Marks)**

b. Construct a transition diagram for recognizing relational operators. Sketch the program segment to implement it, showing the first state and one in final state. **(10 Marks)**

2 a. Write an algorithm to eliminate left recursion from a grammar. Eliminate left recursion from the grammar: $S \rightarrow Aa \mid b,\ A \rightarrow Ac \mid sd \mid a$. **(08 Marks)**

b. Show that the following grammar is ambiguous:

$\text{Stmt} \rightarrow$ if expr then stmt
| if expr then stmt else stmt
| other

write an unambiguous grammar for the same. **(06 Marks)**

c. Give the rules for constructing FIRST and FOLLOW sets. **(06 Marks)**

3 a. What is meant by handle pruning? How it helps in shift reduce parsing? List the actions of a shift reduce parser. **(10 Marks)**

b. For the grammar: $S \rightarrow SS+ \mid SS* \mid a$. Give a bottom-up parse for the input: aaa * a++. **(06 Marks)**

c. What are two types of conflicts during shift-reduce-parsing? Give examples. **(04 Marks)**

4 a. What is the meaning of 'L' and 'R' in LR grammars? Why LR parsing is attractive? **(04 Marks)**

b. Construct canonical LR(1) items for the augmented grammar: $s' \rightarrow s;\ s \rightarrow Cc;\ c \rightarrow cC \mid d$. **(10 Marks)**

c. Write the yacc specification of a simple desk calculator with the following grammar for arithmetic expressions:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$. **(06 Marks)**

PART – B

5 a. Define the following with examples:
i) Synthesized attribute
ii) Inherited attribute
iii) S-Attributed definitions
iv) L-Attributed definitions. **(08 Marks)**

b. Explain the parser stack implementation of postfix STD with an example. **(08 Marks)**

c. Define syntax directed definition for a simple type declaration. **(04 Marks)**

6 a. List any four common three address instruction forms. (04 Marks)

b. Discuss quadruples, triples and indirect triples representation. (06 Marks)

c. Write syntax directed definition for flow of control statements. (10 Marks)

7 a. Explain the desirable properties of memory manager. (06 Marks)

b. Explain in detail, the strategy for reducing fragmentation in heap memory. (08 Marks)

c. Explain the design goals for garbage collector. (06 Marks)

8 a. Discuss the issues in the design of code generator. (10 Marks)

b. Write intermediate code for the following source code

```
for i from 1 to 10 do
for j from 1 to 10 do
    a [i, j] = 0.0;
for i from 1 to 10 do
    a [i, i] = 1.0.
```

(10 Marks)

* * * * *

USN □□□□□□□□□□

10CS63

Sixth Semester B.E. Degree Examination, June/July 2017

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.

PART – A

1 a. List the phases of compiler in order. Use these phases to translate a = bc*cd + 50.00 into the target code in assembly language. **(08 Marks)**

b. What are the applications of compiler? Explain. **(08 Marks)**

c. Write the regular definition and transition diagram for valid unsigned number. **(04 Marks)**

2 a. Why it is necessary for regular expression to define the lexical syntax of a languages? Give reasons. **(04 Marks)**

b. Define ambiguity. Is the following grammar ambiguous? If yes remove the ambiguity and rewrite the grammar

<stmt> → if <expr> then <stmt>

| if <expr> then <stmt> else <stmt>

| a

<expr> → b **(08 Marks)**

c. Find the FIRST and FOLLOW set for the following grammar **(05 Marks)**

E → TX

T → (E)/ int Y

X → +E / ∈

Y → *T / ∈

Fig. Q2 (c)

d. When we say that the grammar G is LL(1) grammar? **(03 Marks)**

3 a. Write an algorithm to construct predictive parser table. Construct a predictive parser table for grammar given in Fig. Q2 (c), and parse the string w = int. **(12 Marks)**

b. Define handle, handle pruning with example. **(03 Marks)**

c. What are the actions a shift-reduce parser makes? Write the parse tree and shift-reduce configurations for the derivation $S \overset{*}{\Rightarrow} \alpha BxAz \Rightarrow \alpha Bxyz \Rightarrow \alpha rxyz$. **(05 Marks)**

4 a. Write a schematic of LR parser. Write the canonical collection of set of LR(0) items and SLR parsing table for the following grammar:

E → E + T / T

T → T * F / F

F → (E) / id **(14 Marks)**

b. Construct LR(1) goto graph for below grammar:

X → YZ / a

Y → bZ / ∈

Z → c **(06 Marks)**

PART – B

5 a. Define synthesized attribute, inherited attributes and attribute grammar. **(03 Marks)**

b. Write a SDD and annotated parse tree for u*s for below grammar suitable for top-down parser.

$T \rightarrow T * F / F$

$F \rightarrow \text{digits}$ **(07 Marks)**

c. Construct a syntax tree for expression a+b–c using the grammar

$E \rightarrow E + T / E - T / T$

$T \rightarrow (E) / \text{id} / \text{num}$ **(06 Marks)**

d. What is the need for eliminating left –recursion? Eliminate left recursion from SDT

$E \rightarrow E + T\{\text{print}('+')\}$

$E \rightarrow T$ **(04 Marks)**

6 a. Which are the common three address instruction forms? Explain. **(09 Marks)**

b. Define jumping code. Translate the following code to jumping code:

if (X < 10 || X > 20 && X = Y) X = 1 **(05 Marks)**

c. Translate the following switch statement to intermediate code.

```
Switch (E) {
                Case V1 : S1 break ;
                Case V2 : S2 break ;
                .
                .
                .
                Case Vn-1 : Sn-1 break ;
                Default : Sn
              }
```

(06 Marks

7 a. Write the possible activations and activation tree corresponding to quick sort call quicksor (1, 9). **(06 Marks**

b. What are the basic functions and properties of memory management? Explain locality i program in detail. **(08 Marks**

c. What is garbage collection? What are the performance metric that must be considered whe designing a garbage collector? **(06 Marks**

8 a. Write intermediate code and flow graph for below code

```
for i from 1 to 10 do
    for J from 1 to 10 do
        a[i, J] = 0.0
    for i from 1 to 10 do
        a[i, i] = 1.0
```

(10 Mark

b. What is the need for optimization? List and explain any three local optimization methods. **(10 Mark**

* * * * *