6. Can a sentence have a single right most derivation but two leftmost derivation? Justify your answer *with example.*

7. What do you mean by handle pruning? Illustrate with *example.*

8. Verify whether grammar is (operator precedence grammar) OPG or not?
$E \rightarrow E + \frac{T}{T}, T \rightarrow T * \frac{F}{F}, F \rightarrow (E)/id$

9. Construct a parse tree for string w=ibt ibt aea, using following grammar:
(1) $S \rightarrow icts$ (2) $S \rightarrow ictses$
(3) $S \rightarrow a$ (4) $C \rightarrow b$

10. Briefly explain predictive parser.

11. Construct the LALR parse table for the following grammar:
$S \rightarrow AA$
$A \rightarrow aA/b$

12. Construct a SLR parsing table for the following grammar:
$E \rightarrow E + T/T$
$T \rightarrow TF/F$
$F \rightarrow F*/a/b$

13. What is syntax directed translation scheme (SDTS)?

14. What are the features and capabilities of symbol table?

15. What is global data flow analysis?

Section-B
(Long Answer Type Questions)

Note: Attempt any three questions. Each question carries 20 marks. **(20x3=60)**

1. (a) How is the source program interpreted? What are similarities and difference between compiler and interpreters? Give example of both, which is preferred over other and why? What do you mean by passes of compiler?

(b) What do you mean by regular expression? Write some properties of a r.e. Write a r.e. over alphabet Σ=(a,b,c) that represents all strings of a length three.

2. (a) Difference between operator grammar and operator precedence grammar. Computer operator precedence relation for the following grammar:
$E \rightarrow E + T/T$; $T \rightarrow T * F/F$; $F \rightarrow (E)/id$;
Show the moves of operator precedence parser for input string id+id*id.

(b) Eliminate the immediate left recursion in the following grammar:
$E \rightarrow E + T/T$, $T \rightarrow T * F/F$, $F \rightarrow (E)/id$

CS-502/3402

B. Tech. (Comp. Engg.) (Semester-V)
Examination- 2012
Compiler Designer

Time: Three Hours
Maximum Marks: 100

Note: Attempt questions from all the sections.

Section –A

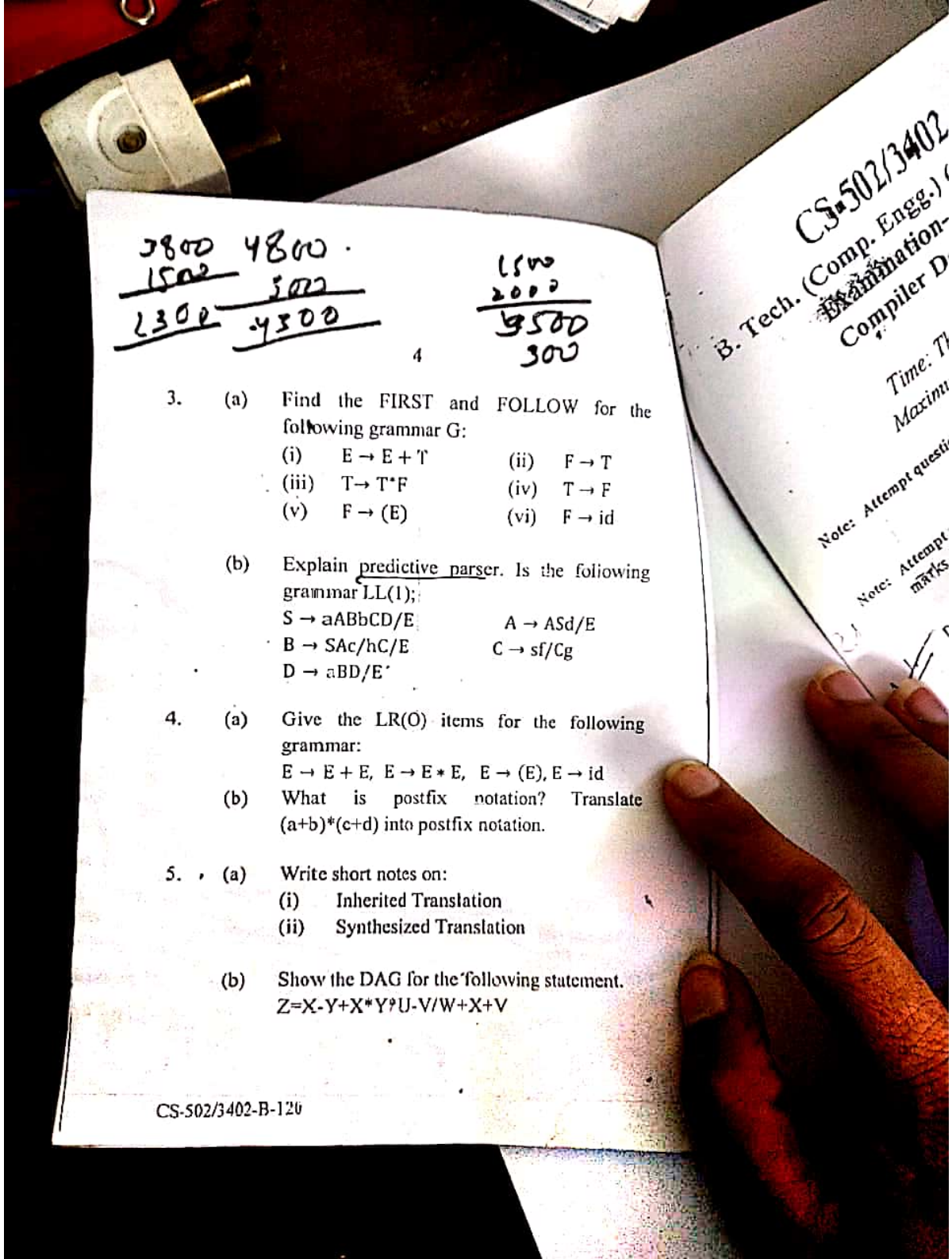
Note: Attempt any ten questions. Each question carries four marks. (4x10=40)

1. Discuss the boot strapping of cross-compilers.

2. Write down the regular expressions to describe the correct syntax for the E-mail address.

3. Eliminate left recursion from the following grammar:
   $S \rightarrow Aa/b$
   $A \rightarrow Ac/Sd/\in$

4. Discuss various data structures used for symbol table with suitable example.



4

3. (a) Find the FIRST and FOLLOW for the following grammar G:

(i) $E \rightarrow E + T$ (ii) $F \rightarrow T$

(iii) $T \rightarrow T^*F$ (iv) $T \rightarrow F$

(v) $F \rightarrow (E)$ (vi) $F \rightarrow id$

(b) Explain predictive parser. Is the following grammar LL(1);

$S \rightarrow aABbCD/E$ $A \rightarrow ASd/E$

$B \rightarrow SAc/hC/E$ $C \rightarrow sf/Cg$

$D \rightarrow aBD/E$

4. (a) Give the LR(O) items for the following grammar:

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

(b) What is postfix notation? Translate (a+b)*(c+d) into postfix notation.

5. (a) Write short notes on:

(i) Inherited Translation

(ii) Synthesized Translation

(b) Show the DAG for the following statement.

Z=X-Y+X*Y?U-V/W+X+V

CS-502/3402-B-120

CS-502/3402

B. Tech. (Comp. Engg.)

Examination-

Compiler D

Time: Th

Maximu

Note: Attempt questi

Note: Attempt

marks

2

Explain logical phase error and syntactic phase error

What do you understand by preliminary scanning? Describe the ways how lexeme are grouped to make a pass.

Explain all the necessary phases and passes in a compiler design.

Consider the following grammar

$S \rightarrow TL;$

$T \rightarrow int/float$

$L \rightarrow L,id/id$

Parse the input string int id, id; using shift reduce parser.

How do you implement LR parsing tables? Why do we need LR parsing tables?

Differentiate between synthesized translation and inherited translation

Generate the three address code for the following program fragment

While (A< C and B> D) do

if A=1 then C=C+1

else while (A<=D) do

A=A+B

Explain code optimization with suitable example.

Explain following terms:

(i) YACC (ii) LALR (iii) SDTS

CS-502/7402-M-120

3

Discuss conversion of NFA into a DFA. Also give the algorithms used in this conversion

Differentiate between top-down parser and bottom-up parser.

Section –B

Note: Attempt any three questions. Each question carries twenty marks. (20x3=60)

1. Give the grammar

$A \rightarrow BB$

$B \rightarrow bB/c$

Construct LR (1) parsing table and also draw the LALR parsing table.

2. Describe the output for the various phases of complier with respect to following statements

Position:= initial +Rate *60

(ii) Write the algorithm for First and Follow function

3. (i) Optimize the codesequence by applying function preserving transformation and loop optimization techniques.

(ii) Discuss different storage allocation strategies used at the time of execution of the programs

CS-502/7402-M-120

CS–3402

B. Tech. (Comp. Engg.) (Fifth Semester)

EXAMINATION, 2019

COMPILER DESIGN

Time : Three Hours

Maximum Marks : 100

Note : Attempt questions from both Sections as directed.

Section—A

(Short Answer Type Questions)

Note : Attempt any *ten* questions. Each question carries 4 marks. 10×4=40

1. Consider the following productions. :

$$S \rightarrow 0B|1A$$

$$A \rightarrow |\ 0S\ |1AA$$

$$B \rightarrow 1S\ |\ 0BB\ |1$$

For the string 00110101, find :

(i) the leftmost derivation

(ii) the rightmost derivation

2. Construct a DFA with reduced states equivalent to the following regular expression :

$$(0+1)^*(00+11)(0+1)^*$$

3. Consider the following left recursive grammar and eliminate the left recursion. Also construct the predictive parsing table :

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

$$T \rightarrow T^*F/F$$

$$F \rightarrow a/b$$

4. What is Syntax-directed-translation ? How are semantic actions attached to the productions ? Explain with example.

5. What is postfix notation ? Translate

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

into postfix using syntax directed translation scheme.

6. Explain the following categories of Intermediate codes :

(i) Quadruples

(ii) Triples

7. Construct the operator precedence parse table for the following Grammar :

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

8. Differentiate synthesized and inherited attributes with example.

9. Verify whether the following grammar is LL(1) *or* not ?

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

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

$$F \rightarrow (F) \mid a \mid b$$

10. What do you mean by passes of compiler ? Explain advantage and disadvantages of single-pass and multi-pass compiler over each other.

11. State the problems associated with Top-Down parsing. State and eliminate the problem

associated with the following grammar for Top-Down parsing :

$$E \rightarrow E + T|T$$

$$T \rightarrow T^* F|F$$

$$F \rightarrow (E)|id$$

12. Construct the predictive parse table for the following grammar :

$$S \rightarrow iEtSS_1$$

$$S_1 \rightarrow eS|E$$

$$E \rightarrow b$$

13. What are different types of symbol table organization possible ?

14. Explain the six phases of compiler with diagram.

15. What is hashing ? What are different types of hashing technique available ?

Section—B

(Long Answer Type Questions)

Note : Attempt any *three* questions. Each question carries 20 marks. 3×20=60

1. Consider the following grammar :

$$S \rightarrow iSeS$$

$$S \rightarrow iS$$

$$S \rightarrow a$$

(i) List the canonical collection of sets of LR (0) items for the given grammar.

(ii) Construct SLR parsing table for the grammar.

2. Construct the Canonical LR parsing table for the following grammar :

$$E \rightarrow E + T|T$$

$$T \rightarrow T^*F|F$$

$$F \rightarrow (E)\,|id$$

3. Explain the need of code optimization. With example, illustrate loop optimization.

4. What are the various storage management techniques available ? What are their importance in compiler design ?

5. What do you understand by Lexical phase error and syntactic phase error. Also suggest method for recovery of errors.

6. Differentiate between LR and LALR parsers. Construct CLR parse table for :

$$S \rightarrow AA$$

$$A \rightarrow aA|d$$