

Code No: 125DR

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD****B. Tech III Year I Semester Examinations, November/December - 2017****AUTOMATA AND COMPILER DESIGN****(Information Technology)****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 a context free grammar? Design a CFG that generates floating-point literal. [2]
- b) Considering the following grammar, remove left recursion and left factor the grammar to be LL (1). [3]  
 $L \rightarrow L ; S \mid S$   
 $S \rightarrow id = E \mid id (E)$   
 $E \rightarrow id \mid num$
- c) What is syntax directed definition? Give a suitable example? [2]
- d) Find the canonical collection of sets of LR(1) items [3]  
 $S \rightarrow AaAb$   
 $A \rightarrow BbBa$   
 $A \rightarrow \epsilon$   
 $B \rightarrow \epsilon$
- e) What is the difference between type conversion and type checking? Explain with the help of a high level language? [2]
- f) What is unit production in CFG? What is the procedure to eliminate Unit productions? [3]
- g) Explain the principle sources of optimization. What are the patterns used for code optimization? What are the 3 areas of code optimization? [2]
- h) What are the rules to determine the leaders of basic blocks? What is the use of algebraic transformation? [3]
- i) What is a flow graph? Mention the applications of DAGs. [2]
- j) Construct a DAG for the expression  $a=b*-c + b*-c$ . [3]

**PART - B****(50Marks)**

- 2.a) Consider the following grammar  
 $lexp \rightarrow atom \mid list$   
 $atom \rightarrow number \mid identifier$   
 $list \rightarrow ( lexp\_seq )$   
 $lexp\_seq \rightarrow lexp\_seq lexp \mid lexp$   
 Answer the following:
- b) Remove left recursion.
- c) Construct FIRST and FOLLOW set for the non-terminals of the resulting grammar.
- d) Construct LL(1) parsing table for the resulting grammar.
- e) Show that the resulting grammar is LL(1).
- f) Show the parsing action of LL(1) parser for the input string (a (b (2) (c)). [10]

**OR**

3.a) Construct an LL(1) parsing Table for the following grammar

**S** → **aBDh**

**B** → **cC**

**C** → **bC** |  $\epsilon$

**D** → **EF**

**E** → **g** |  $\epsilon$

**F** → **f** |  $\epsilon$

b) Construct DFA equivalent to the NFA ( $\{p, q, r, s\}, \{0,1\}, \delta, p, \{s\}$ ) where the transition function  $\delta$  is given in following table: [5+5]

States	0	1
p	p, q	p
q	r	r
r	s	-
s	s	s

4.a) Discuss the construction of **LR** parser. What are the various data structures used in **LR** parser design? Discuss the construction of **ACTION[ ]** and **GOTO[ ]** table.

b) Write syntax directed definition to implement of a desk calculator with an LR parser and show the evaluation of expression '95\*4 +5'. [5+5]

**OR**

5.a) Describe the syntax directed translation procedure for assignment statements with integers and mixed types and explain.

b) Construct LALR(1) parsers for the following grammar. [5+5]

**S** → **L = R**

**S** → **R**

**L** → **\* R**

**L** → **id**

**R** → **L**

6.a) What is phase structure grammar? State and explain with suitable examples, Chomsky classification of grammar. Give the corresponding language generated by each grammar and their relation.

b) **S** → **aAa** | **bBb** |  $\epsilon$

**A** → **C** | **a**

**B** → **C** | **b**

**C** → **CDE** |  $\epsilon$

**D** → **A** | **B** | **ab**

For the above grammar eliminate the useless symbols(if any). Eliminate  $\epsilon$ -production, Eliminate unit production, then convert the grammar to Chomsky normal form. [5+5]

**OR**

7.a) What is phase structure grammar? What is Chomsky normal form of grammar? Explains the steps used to reduce a CFG to CNF.

b) Convert the following grammar into Chomsky normal form. [5+5]

**S** → **AAA** | **B**

**A** → **aA** | **B**

**B** →  $\epsilon$

- 8.a) Give the block diagram of organization of code optimizer. What are the advantages of the organization of code optimizer?  
b) What is dynamic storage allocation? Why is not static allocation sufficient for everything? [5+5]

**OR**

- 9.a) What is activation record? Discuss the structure of a typical activation record. What do you mean by a layout of a activation record?  
b) Explain the machine dependent and machine independent code optimization? What are their advantages? How to represent the dummy blocks with no statements indicated in global dataflow analysis? [5+5]

- 10.a) Define a Directed Acyclic Graph. Construct a DAG and write the sequence of instructions for the expression  $a + a * (b - c) + (b - c) * d$ .  
b) Give the sequence of three-address code instructions corresponding to the following fragment using the above attribute grammar [5+5]

**for i= 1 to n step m+k do**  
**s:= s+i**

**OR**

- 11.a) Explain the code generation process involving the environment of the code generator. Explain the steps in code generation of the expression  $(A + B) / C + D$ . Assuming two machine registers are available.  
b) Explain the various issues in the design of code generation. Construct the DAG for the following basic block. [5+5]

$d := b * c$   
 $e := a + b$   
 $b := b * c$   
 $a := e - d$

---ooOoo---