

III B. Tech I Semester Supplementary Examinations, October/November - 2018

COMPILER DESIGN

(Computer Science and Engineering)

Time: 3 hours

Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)

2. Answering the question in **Part-A** is compulsory

3. Answer any **THREE** Questions from **Part-B**

PART -A

- | | | | |
|---|----|---|------|
| 1 | a) | Write the regular definition and transition diagram for identifiers and reserved words. | [3M] |
| | b) | Differentiate Parse tree and Syntax tree with an example. | [4M] |
| | c) | What is the significance of Operator precedence | [4M] |
| | d) | What is semantic rule? How to evaluate the semantic rules? | [4M] |
| | e) | Write a short note on peephole optimization. | [4M] |
| | f) | What is strength reduction? Give an example. | [3M] |

PART -B

- | | | | |
|---|----|--|-------|
| 2 | a) | Explain the role of assembler, compiler, loader and linker in the language processing system. | [8M] |
| | b) | Write about the following with respect to lexical analyzer.
i) Relationship with regular expressions and regular definitions ii) Lexical errors. | [8M] |
| 3 | a) | Explain the structure of predictive parser. How to handle error in it? | [6M] |
| | b) | Construct the non recursive predictive parse table for the given grammar and check the acceptance of input string abfcg
$S \rightarrow A$ $A \rightarrow aB/Ad$ $B \rightarrow bBC/f$ $C \rightarrow cg$ | [10M] |
| 4 | a) | Explain the working principle of CLR(1) parser and construct the parse table for the given grammar $S \rightarrow L=R/R$ $R \rightarrow L$ $L \rightarrow *R/id$ | [10M] |
| | b) | Using the CLR (1) table constructed above check the acceptance of input string id=id/id and also explain the algorithm for this. | [6M] |
| 5 | a) | What is intermediate code? Translate the expression (a+b)/(c+d)*(a+b/c)-d into quadruples, triples and indirect triples. | [8M] |
| | b) | Write and explain the Syntax Directed definition for the grammar
$E \rightarrow E1+T/E1-T/T$ $T \rightarrow (E)/id/num.$ | [8M] |
| 6 | a) | Consider the C program and generate the code and Write different object code forms
Main() { int i, a[10]; while (i<=10) a[i]=i*5; } | [8M] |
| | b) | What is Activation Record? Explain its usage in stack allocation strategy. How it is different from heap allocation? | [8M] |
| 7 | | Explain the following machine independent optimization techniques. | |
| | a) | Common sub expression and dead code elimination | [6M] |
| | b) | Copy propagation, constant folding. | [5M] |
| | c) | Instruction scheduling. | [5M] |
