

Code No: R1631051

R16**SET - 1****III B. Tech I Semester Supplementary Examinations, May - 2019****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. Answer **ALL** the question in **Part-A**
 3. Answer any **FOUR** Questions from **Part-B**
- ~~~~~

PART -A

1. a) What is the role of compiler in bootstrapping operation? [2M]
- b) Write context free grammar for polish notation of arithmetic expressions. [2M]
- c) Construct parse tree and syntax tree for $4-6/3*5+7$. [2M]
- d) Apply translation scheme to generate three-address code $a < b$ or $c < d$. [3M]
- e) Write in detail about the sub-division of run-time memory. [3M]
- f) Copy propagation leads to dead-code elimination, justify this with example. [2M]

PART -B

2. a) Write short notes on hierarchical and linear analysis operations. [7M]
- b) Regular expressions are important for lexical analysis? Explain the reason with examples. [7M]
3. a) **G: $S \rightarrow (L) | a$ $L \rightarrow L, S | R$, $R \rightarrow b$** for the given grammar find LR(0) items. [7M]
- b) For the above grammar G construct LR parsers and explain how shift, reduce accept or reject operations are performed. [7M]
4. a) Write a short note on error recovery with LR parsers. How it is different from LL parsers? [7M]
- b) List and explain the algorithmic steps to construct LALR parser for grammar **$S \rightarrow L = R | R L \rightarrow * R | id R \rightarrow L$** . [7M]
5. a) Explain the role of type checking in error detection and recovery. [7M]
- b) Write various semantic routines used to construct abstract syntax tree with an example. [7M]
6. a) Write pseudocode for finding sum of 'n' numbers. And identify basic blocks then construct the flow graph for it. Explain the rules used for this. [7M]
- b) How to access non-local data? Explain implication details with example. [7M]
7. Explain the following two classes of local machine independent transformations [14M]
 - i) Structure preserving transformations
 - ii) Algebraic transformations.
