Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 20374

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2018.

Sixth Semester

Computer Science and Engineering

CS 6660 — COMPILER DESIGN

(Common to Information Technology)

(Regulations 2013)

(Also common to PTCS 6660 — Compiler Design – for B.E. (Part-Time) Fifth Semester – Computer Science and Engineering – Regulations 2014)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Recall the basic the two parts of a compilation process.

2. How a source code is translated to machine code?

3. State the rules to define regular expression.

4. Construct Regular expression for the language $L = \{w \in \{a, b\} / w$ ends in $abb\}$.

5. What are the different stages that a parser can recover from a syntactic error?

6. Define LR (0) item.

7. List three kinds of intermediate representation.

8. When procedure call occurs, what are the steps taken?

9. State the problems in code generation.

10. Define common sub expression.

PART B — (5 × 13 = 65 marks)

11. (a) Write short notes about :

(i) Compiler Construction Tools. (7)

(ii) Lexeme, token and pattern. (6)

Or

(b) Discuss in detail about the operations of compiler which transforms the source program from one representation into another. Illustrate the output for the input : (13)

$a = (b + c) * (b + c) * 2$.

12. (a) Write briefly about :

(i) the role of Lexical analyzer with the possible error Recovery actions. (5)

(ii) recognition and specification of tokens. (8)

Or

(b) Construct the minimized DFA for the regular expression $(0+1)*(0+1)01$. (13)

13. (a) Show that the following grammar

$S \rightarrow Aa \mid bAc \mid dc \mid bda$

$A \rightarrow a$

is LALR(1) but not SLR(1). (13)

Or

(b) Show that the following grammar

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

$A \rightarrow d$

$B \rightarrow d$

is LR(1) but not LALR(1). (13)

14. (a) Apply the S-attributed definition and constructs syntax trees for a simple expression grammar involving only the binary operators + and −. As usual, these operators are at the same precedence level and are jointly left associative. All nonterminal have one synthesized attribute node, which represents a node of the syntax tree.

Production: $E \rightarrow E_1 + T$, $E \rightarrow T$, $T \rightarrow (E)$, $T \rightarrow$ id/num . (13)

Or

(b) Discuss in detail about :

(i) Storage allocation strategies. (7)

(ii) Parameter passing methods. (6)

15. (a) Discuss in detail about optimization of basic blocks. (13)

Or

(b) Explain in detail about issues in the design of a code generator. (13)

PART C — (1 × 15 = 15 marks)

16. (a) Suppose we have a production $A \rightarrow B\ C\ D$. Each of the four nonterminals has two attributes s, which is synthesized, and i, which is inherited. For each set of rules below, check whether the rules are consistent with (i) an S-attributed definition, (ii) an L-attributed definition (iii) any evaluation order at all.

(1) $A.s = B.i + C.i$

(2) $A.s = B.i + C.s$ and $D.i = A.i + B.s$

(3) $A.s = B.s + D.s$

(4) $A.s = D.i$
$B.i = A.s + C.s$
$C.i = B.s$
$D.i = B.i + C.i$. (15)

Or

(b) Construct a Syntax-Directed Translation scheme that translates arithmetic expression from infix into postfix notation. (15)