www.rgpvonline.com

Unit-V

5. a) What is code optimization? How it is achieved?

   b) Explain local and global optimization.

   c) What are semantic errors?

   d) Explain code generation techniques.

OR

Explain optimizing transformations commonly used in compilers.

******

www.rgpvonline.com

Total No. of Questions :5] *[Total No. of Printed Pages :4*

Roll No ....

MCA - 405(B)

MCA. IV Semester

Examination, June 2015

Compiler Design

(Elective - I)

Time : Three Hours

Maximum Marks : 70

Note: i) Answer five questions. In each question part A, B, C is compulsory and D part has internal choice.

ii) All parts of each question are to be attempted at one place.

iii) All questions carry equal marks, out of which part A and B (Max. 50 words) carry 2 marks, part C (Max. 100 words) carry 3 marks, part D (Max. 400 words) carry 7 marks.

iv) Except numericals, Derivation, Design and Drawing etc.

Unit-I

1. a) Write various phases of complier.

   b) Distinguish between compiler and translator

   c) Explain boot strapping.

   d) Discuss the implementation of lexical analyser.

OR

Explain various tools for lexical analyser.

www.rgpvonline.com

Unit-II

2. a) Explain bottom up parsing?

   b) Differentiate between regular expression and context free grammars.

   c) Write about ambiguity of parse tree.

   d) Discuss the complication of top down parsing.

OR

What is predictive parsing? Explain why it is important?

Unit-III

3. a) Explain LR parsing.

   b) Write about syntax directed translation scheme.

   c) What is YACC?

   d) Given a grammar with the following rules:

$S \rightarrow A\#$

$A \rightarrow bB$

$B \rightarrow cC$

$c \rightarrow dA$

$A \rightarrow a$

   i) Generate the sets of LR(1) items.

   ii) Is the grammar LR(1)? if not why?

OR

Create a set LR(0) items and transitions for the following grammar. Show that the grammar is not LR(0). Give the SLR(1) parse table for the grammar.

Terminals: {num, id [,], $}

Non terminals: {E', E, V}

Rules: E' → E$

E → V

E → num

V → id

V → v [E]

Start symbol E'

Unit-IV

4. a) What are the major issues in code generation for expressions?

   b) Explain triples and quadruples for representing intermediate code for expression

   c) Explain calling convention in a static memory allocation.

   d) Explain various parameter parsing mechanisms compare and contrast in terms of execution efficiency and power to produce side effects.

OR

Consider the case control structure of 'C' comment on how the code differs from nested if statements. Design code generation model for the case statement.