Question Paper

Exam Date & Time: 26-Apr-2018 (09:30 AM - 12:30 PM)



MANIPAL ACADEMY OF HIGHER EDUCATION

INTERNATIONAL CENTRE FOR APPLIED SCIENCES **IV SEMESTER B.S. (ENGG) END-SEMESTER THEORY EXAMINATION- APRIL 2018** DATE:26.04.2018 **TIME:9.30AM TO 12.30PM** Language Processors [CS 244]

Marks: 100

Duration: 180 mins.

Answer 5 out of 8 questions.

- 1) Briefly explain the structure of Lex program. Also, show (10)with necessary block diagram, how the Lexical Analyzer is A)
 - generated using Lex.
 - Explain the different phases of a compiler with neat block B) (6) diagram.
 - C) What are the reasons for separating the analysis phase of (4) compiling into lexical analysis and parsing? List and explain.
- 2) Construct an NFA for the regular expression abb(a|b)* and (10)convert it to DFA using subset construction method. Show A) all the intermediate steps.
 - B) Eliminate left-recursion and left-factor the following Pascal-⁽⁶⁾ like grammar. The terminals are specified in boldface. varDecl → var decList decList → decl ; decList | decl decl→idList : id idList → idList , id | id
- C) What is the drawback of having one input buffer scheme in $^{(4)}$ Lexical Analysis? How is it overcome? Explain 3) (10)
 - Consider the following grammar

A)

		$S \rightarrow uBDt$ $B \rightarrow wT$ $T \rightarrow vT \epsilon$ $D \rightarrow EF$ $E \rightarrow y \epsilon$ $F \rightarrow x \epsilon$	
		 I) Compute FIRST and FOLLOW for all non-terminals. II) Construct the LL(1) parsing table. III) Check whether the grammar is LL(1). 	
	B)	What is top-down parsing? Bring out the differences between the two types of top down parsers? Explain.	(4)
	C)	Write the recursive descent parser for the following grammar. The terminals are specified in boldface. $S \rightarrow if E \text{ then } S \mid if E \text{ then } S \text{ else } S \mid a \mid b$ $E \rightarrow x \mid y$	(6)
4)		Give the algorithm for computing Closure of Itemsets .	(4)
	A) B)	What are kernel and non-kernel items? Illustrate using the following grammar: $E \rightarrow E+T T$ $T \rightarrow T*F F$ $F \rightarrow (E) id$	(4)
	C)	Construct the Collection of sets of LR(0) items for the grammar given in Q. No. 4B. Also, draw the DFA for viable prefixes.	(12)
5)	A)	Explain with a suitable example, how a DAG is different from a syntax tree. Construct a DAG for the following sequence of expressions: t0 = a + b t1 = t0 + b d = t0 + t1	(6)
	B)	Write the algorithm for non-recursive predictive parsing. Specify the input, output and initial configuration.	(6)
	C)	What is a three address code? Represent the following expression using three address code and its Syntax tree, Quadruple and Triple representation. $a = b^*-c+b^*-c$	(8)
6)	A)	Briefly discuss the following: i) Type expression ii) Type equivalence	(4)
	B)	Explain the following design issues related to Code	(6)

Generation:

- i) Input to Code Generator
- ii) Instruction Selection
- iii) Register Allocation
- C) Discuss the two strategies used for dynamic storage (10) allocation. Explain the various fields of general Activation record.
- With the help of neat diagram, explain the concept of multi ⁽⁸⁾
 pass translation in assemblers.
 - ^{B)} What is a basic block and flow graph? Give the algorithm ⁽¹²⁾ for partitioning three address statements into basic blocks. Also discuss how flow graphs can be represented.
- With reference to run-time storage organization, explain (10)
 calling sequences and return sequences.
 - ^{B)} What is backpatching? Explain.

- (4)
- ^{C)} Give the algorithm for determining the liveness and next- ⁽⁶⁾ use information for each statement in a basic block.

-----End-----