B.Sc. Engg. CSE 8th Semester (48) 31 October 2016

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION SUMMER SEMESTER, 2015-2016

DURATION: 3 Hours FULL MARKS: 150

CSE 4801: Compiler Design

Programmable calculators are not allowed. Do not write anything on the question paper.

There are 8 (eight) questions. Answer any 6 (six) of them.

Figures in the right margin indicate marks.

---

1. a) Discuss the role of an *Error Handler* during compilation. 7
   b) Why is buffering used in lexical analysis? Explain various buffering techniques. 3+5
   c) Write a *Lex* program to count similar character sequence of length two or more in a given text file. 10

2. a) Write short notes on ***front end*** and ***back end*** of a compiler. 7
   b) Classify *grammars* and define each of the class. 6
   c) Explain the position and role of a *lexical analyzer* in a multi-phase compilation model. 7
   d) Explain the function and uses of *yywrap()* in details. 5

3. a) Discuss on various error recovery strategies to recover from lexical errors. 11
   b) Write regular definitions for the following languages: 9
      i. All strings of letters that contain the five vowels in order.
      ii. All strings of 0's and 1's that do not contain the substring 011.
      iii. All strings of letters in which the letters are in ascending lexicographic order.
   c) What is left factoring? Discuss with example. 5

4. a) Consider the grammar 2+4

   $S \rightarrow (L) \mid a$

   $L \rightarrow L, S \mid S$

      i. What are the terminals, nonterminals, and start symbol?
      ii. Find the parse tree for the following sentence:

      (a,((a,a),(a,a)))

   b) Consider the grammar

   $S \rightarrow aSbS \mid bSaS \mid \varepsilon$

      i. Show that this grammar is ambiguous by constructing two different leftmost derivations for the sentence *abab*. 6
      ii. Construct the corresponding rightmost derivations for abab. 6
      iii. Construct the corresponding parse tree for abab. 3
      iv. Describe the language generated by the grammar. 4

5. a) Figure 1 represents the SLR parse table for the following grammar. Step by step show the parsing for the input: *id*(id+id)*

0. E' → E
1. E → E + T
2. E → T
3. T → T * F
4. T → F
5. F → (E)
6. F → id

State	+	*	(	)	id	$		E	T	F
0			s4		s5			1	2	3
1	s6					accept				
2	r2	s7		r2		r2				
3	r4	r4		r4		r4				
4			s4		s5			8	2	3
5	r6	r6		r6		r6				
6			s4		s5				9	3
7			s4		s5					10
8	s6			s11						
9	r1	s7		r1	r1					
10	r3	r3		r3	r3					
11	r5	r5		r5	r5					

Figure 1: SLR Parse Table

b) Discuss the impact of left recursive property of a grammar on top-down and bottom-up parsing with their remedies. 10

c) What is bootstrapping?

3

6. a) What are the two types of attributes that are associated with a grammar symbol? State their properties. 3

b) Explain the bottom-up evaluation of S-Attributed Definitions using parser stack. 8

c) Consider the following grammar 5

$E \rightarrow E + T \mid T$
$T \rightarrow T F \mid F$
$F \rightarrow F^{*} \mid (S) \mid a$

Find the set of First and Follow for each of the nonterminal.

d) Write various types of three address form of intermediate code.

9

7. a) You need to develop a calculator program to perform addition, subtraction, multiplication and division operations. Write down the programs using Lex and Yacc. 13

b) You need to develop a lexical analyzer to recognize 5 different types of life forms (domestic, fishes, birds, wild, and insects) from a given sentence. The program can update its database by adding new names for each type of life form. For example, following line of input will add 3 new names *rui, koi, and ilish* to fish category. 12

fish rui, koi, ilish

Write a Lex program to construct the lexical analyzer.

8. a) A string contains a random sequence of numbers and words. Write a *Lex* program to read the string and replace each set of consecutive numbers by their count and replace each set of consecutive words by concatenate them. 10

Sample input: IUT 50 60 70 CSE Department 10 31
Output: IUT 3 CSEDepartment 2

b) Write a *Lex* program which will take a file name as an argument and count the number of uppercase and lowercase letters, digits, words, lines and other symbols presented in the file. 15