281

B.Sc.Engg (CSE) 8th Semester **13 April 2022**

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION **SUMMER SEMESTER, 2020-2021**

DURATION: 3 HOURS **FULL MARKS: 150**

CSE 4801: Compiler Design

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

Answer all **6 (six)** questions. Marks of each question and corresponding CO and PO are written in the right margin with brackets.

1. Consider the context-free grammar shown below and answer the following questions:

$$
\begin{aligned}
&G \rightarrow L \\
&L \rightarrow E\,;\,L \\
&L \rightarrow E \\
&E \rightarrow E + T \\
&E \rightarrow T \\
&T \rightarrow \textbf{id} \\
&T \rightarrow \textbf{id}\;(\;) \\
&T \rightarrow \textbf{id}\;(\;L\;)
\end{aligned}
$$

a) Find the set of FIRST(x) and FOLLOW(x), where x is a non-terminal. 4 (CO2, PO1)

b) Generate canonical LR(0) collection of items for the grammar. 8 (CO2, PO1)

c) Generate the SLR parse table 10 (CO2, PO1)

d) Is the grammar SLR(1)? Justify your answer. 3 (CO2, PO2)

2. a) List the contents of an activation record (for a procedure call) along with brief description. 5 (CO4, PO1)

b) Design syntax-directed definitions to generate intermediate codes for the following statements: 20 (CO5, PO3)

$S \rightarrow$ **if** E **then** $S1$
$S \rightarrow$ **if** E **then** $S1$ **else** $S2$
$S \rightarrow$ **do** $S1$ **while** E
$S \rightarrow$ **while** E **do** $S1$

3. a) Draw the block diagram of a language processing system and briefly discuss each of its components. 10 (CO1, PO1)

b) As a member of a compiler construction team you are asked to implement a *symbol table* along with *symbol table manager*. Discuss the implementation strategy you would follow to complete the task with fast access time and efficient memory uses. 10 (CO5, PO3)

c) Discuss the transformation of a grammar which are needed to apply top-down parsing. 5 (CO2, PO1)

282

4. a) Consider the context-free grammar shown below and respective parse table shown in Table 1: 13 (CO2, PO2)

$G \rightarrow L$
$L \rightarrow L\ P$
$L \rightarrow P$
$P \rightarrow (\ P\ )$
$P \rightarrow (\ )$

Table 1: Parsetable

state	action			goto	
	(	)	$	L	P
0	s3			1	2
1	s3		accpt		4
2	r3		r3		
3	s6	s7			5
4	r2		r2		
5		s8			
6	s6	s10			9
7	r5		r5		
8	r4		r4		
9		s11			
10		r5			
11		r4			

Show in full detail, the steps that an LR(1) parser would follow to parse the string (( )) ( ) using the above grammar. For each step of the parsing, show the contents of the stack, present input symbol and the action taken.

b) A compiler designer writes following grammar to support *if-then-else* statement: 12 (CO2, PO2)

stmt → *if expr then stmt*
  | *if expr then stmt else stmt*
  | *other*

Then he realizes that the grammar is ambiguous. So he rewrites the grammar as follows to remedy the dangling-else ambiguity:

stmt → *if expr then stmt*
  | *matched_stmt*
matched_stmt -> *if expr then matched_stmt else stmt*
  | *other*

Show that the grammar is still ambiguous.

5. a) Write a *Lex* program which can recognize presence of an even number of alphabetic strings followed by an odd number of integers in a text file. Text file name will be supplied as an argument to the program. The Lex program will report start and end position of such sequence(s) present in the provided text file. 10 (CO1, PO3)

b) In C language, variables can be declare as per following format- 10 (CO2, PO3)
data_type $var_1, var_2, var_3, \ldots\ldots, var_n$;

Common data type keywords in C are *int*, *char* and *float*.
Design a grammar to recognize multiline of variable declarations as per C syntax.

6. a) Write a program using *Lex* and *Yacc* that can convert a prefix expression into postfix expression. 10 (CO1, PO3)

b) A compiler is needed to provide recursive call for functions. The compiler designer chose *static allocation* strategy for run-time memory allocation for functions. Explain why the selected run-time memory allocation strategy will fail to support the required recursive function call. 10 (CO4, PO2)

c) Design a tree traversal algorithm to evaluate L-Attributed definitions. Write down the pseudocode to implement the algorithm. 10 (CO3, PO3)