Nirma University

Institute of Technology

Semester End Examination (IR/RPR), December 2018

B.Tech. in Computer Engineering, Semester – VII

IT794 – COMPILER CONSTRUCTION

Roll / Exam No. _____ Supervisor's Initial with Date _____

Time: 3 Hours Max Marks :100

Instructions: 1 Attempt all questions of Section I and II separately in same Answerbook.
2. Figure to right indicate full marks
3. Draw neat sketches wherever necessary.
4. Assume suitable data wherever necessary and mention the same.

SECTION – I

Q-1. Do as directed. **[18]**

A) Compare single pass compiler with multi-pass compiler. (5)

B) Explain role of each phases used in compiler (5)

C) What is difference between Look-ahead and follow set? Demonstrate it using suitable example. (4)

D) Define terms: Lexeme, Token. Which variables from YACC specify lexeme and its corresponding value. (4)

Q-2. Answer following Questions. **[16]**

A) Is following grammar LL(1), LR(1) or both? (8)

$E \rightarrow A \mid B$
$A \rightarrow a \mid c$
$B \rightarrow b \mid c$

OR

A) Write the actions of LR parser to parse string '*aa1bbb*', for the grammar and parse table shown below: (8)

Grammar:

1) $S \rightarrow A$ 2) $S \rightarrow B$ 3) $A \rightarrow aAb$ 4) $A \rightarrow 0$ 5) $B \rightarrow aBbb$ 6) $B \rightarrow 1$

Parse Table:

	ACTION					GOTO		
State	a	b	0	1	$	S	A	B
0	S1		S2	S3		11	4	5
1	S1		S2	S3			6	7
2		r4	r4		r4			
3	r6	r6	r6	r6	r6			
4	r1	r1	r1		r1			
5	r2	r2	r2	r2	r2			
6		S8						
7		S9						
8		r3						
9		S10						
10		r5			r5			
11					acc			

B) Eliminate left recursion from grammar described in following syntax directed definition and modify semantic rules accordingly. (8)

Grammar	Semantic rule
$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
$E \rightarrow E_1 - T$	$E.val = E_1.val - T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T_1 * NUM$	$T.val = T_1.val * NUM.val$
$T \rightarrow T_1 / NUM$	$T.val = T_1.val / NUM.val$
$T \rightarrow NUM$	$T.val = NUM.val$

Q-3. Do as directed. **[16]**

A) Which of these grammar is ambiguous? Prove using constructing parse tree for an input string '*A B C * D * E F*'. (6)

Grammar 1:	Grammar 2:
$E \rightarrow T \mid E\,T$	$E \rightarrow T \mid T\,E$
$T \rightarrow ID \mid ID * T$	$T \rightarrow ID \mid T * T$
$ID \rightarrow A \mid B \mid C \mid D \mid E \mid F$	$ID \rightarrow A \mid B \mid C \mid D \mid E \mid F$

B) Trace panic mode error recovery for an input string '*id id * (id – id* ' using following parse table. (6)

	Id	+	*	(	)	$
E	$E \rightarrow T\,E_R$			$E \rightarrow T\,E_R$	*synch*	*synch*
E_R		$E_R \rightarrow +T\,E_R$			$E_R \rightarrow \varepsilon$	$E_R \rightarrow \varepsilon$
T	$T \rightarrow F\,T_R$	*synch*		$T \rightarrow F\,T_R$	*synch*	*synch*
T_R		$T_R \rightarrow \varepsilon$	$T_R \rightarrow * F\,T_R$		$T_R \rightarrow \varepsilon$	$T_R \rightarrow \varepsilon$
F	$F \rightarrow$ id	*synch*	*synch*	$F \rightarrow (\,E\,)$	*synch*	*synch*

C) Give example of shift-reduce conflict and reduce-reduce conflict. (4)

OR

C) Explain why the following grammar is not LL(k) for any k: (4)

$S \rightarrow A \mid B$
$A \rightarrow aaA \mid aa$
$B \rightarrow aaB \mid a$

SECTION – II

Q-4. Answer following Questions. **[18]**

A) Does it make difference to put code optimization phase module before code generation phase or after code generation phase? Explain using proper example. (5)

B) Explain following static checking performed by semantic analyzer phase using suitable example. (5)
i) Type checks ii) Flow-of-control-checks

C) Explain any two local code optimization techniques. (4)

D) "Every S-attributed syntax directed definition is L-attributed syntax directed definition". Write your opinion about this statement with proper justification. (4)

Q-5. Do as directed. **[16]**

A) Write semantic rules for generating intermediate code for the following two constructs: (8)

i) repeat-until construct:
S -> repeat S1 until E

ii) if construct:
S -> if E then S1 else S2

B) Suppose we have the following C declarations: (8)

```
struct { int a , b ; } CELL;
CELL foo[100] , *PCELL;
PCELL bar( int x , CELL y) {...}
```

Draw graphical presentation of type expressions for the types of foo and bar.

OR

B) Describe contents of 'Symbol Table' used in compiler. Draw hierarchy of symbol table for below given code fragment. (8)

```
int n, a;
void main()
{
  int a;
  float b;
  {
     char b;}}
```

Q-6. Do as directed. **[16]**

A) Generate three address intermediate code for 'a:=*b * (-c)+ b * (-c)*' and represent it in triples, indirect triples and quadruples formats. (6)

B) Allocate registers to each variables in following code fragment using *getReg* algorithm (6)

```
t := a-b
u := a-c
v := t+u
d := v+ u
```

C) What is difference between syntax directed definition and translation scheme? Explain it for translating infix binary expression to postfix expression (4)

OR

OR

C) Draw the Control Flow Graph for the code given below: (4)

```
        receive m
        f0 <- 0
        f1 <- 1
        if m <= 1 goto L3
        i <- 2
L1:     if i<= m goto L2
        return f2
L2:     f2 <- f0 + f1
        f0 <- f1
        f1 <- f2
        i <- i+1
L3:     return m
```