CS319 Compiler Design

Time: 1 hr. MM. 20

1. (a) What is left recursion elimination and left factoring? What is their significance in context of predictive parser? 3

(b) Consider the following grammar. 3

$A \rightarrow BA'$

$A' \rightarrow +BA' \mid \epsilon$

$B \rightarrow CB'$

$B' \rightarrow *CB' \mid \epsilon$

$C \rightarrow (A) \mid id$

Find the FIRST and FOLLOW set of all non terminals.

2. Consider the following grammar. 7

$S \rightarrow iSeS$

$S \rightarrow iS$

$S \rightarrow a$

Construct the sets of LR(0) items and SLR parsing table, then parse the string ***iiaea***. Any shift/reduce conflict will be resolved in favor of shift.

3. (a) Consider the following grammar. 7

$S \rightarrow L=R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow id$

$R \rightarrow L$

Construct the sets of LR(1) items and Canonical LR parsing table, then parse the string ***id=id***.

OR

(b) Consider the following operator grammar.

$E \rightarrow E+E \mid E*E \mid E\hat{}E \mid (E) \mid id$

Where, ^ (exponentiation operator) is of highest precedence and right associative. Construct the precedence functions, and parse the string ***id*(id^id)+id*** using operator precedence parser.

Roll No. 16419 CM PG

Department of Computer Science, Banaras Hindu University, Varanasi – 221005.

MCA-IV / M.Sc-II Semester - CS319 – Compiler Design – Mid-term Examination–2019-20

Date: 05-03-2020 Duration: 60 minutes Total Marks: 20 Min. 10

Answer **all** the questions

1. Define the terms *lexeme, token* and *pattern*. (3 marks)
2. Consider the context free grammar: (3 marks)

$$S \longrightarrow SS+ \mid SS* \mid a$$

and the string $aa+a*$.

(a) Give leftmost derivation for the string. (b) Give rightmost derivation for the string.
(c) Give a parse tree for the string.

3. Write the unambiguous context free grammar for *if-then-else* statement. Compute the Follow sets of the grammar. (4 marks)
4. Construct the collection of sets of LR(0) items for the following grammar. (5 marks)

$$S' \longrightarrow S$$
$$S \longrightarrow aAS \mid c$$
$$A \longrightarrow ba \mid SB$$
$$B \longrightarrow bA \mid S$$

5. Construct the SLR parse table for the grammar given in question number 4 and the collection of sets of LR(0) items constructed for the grammar. (5 marks)

MS/SemIII/412

Roll No. 16419CMP029

M.Sc. III[rd]-Semester Examination 2017-18

Computer Science

Paper code: CS-319: Compiler Design

Time: Three hours **Max marks: 70**

Attempt five questions including question no. 1 which is compulsory

1. Write short notes on the following: 3.5*4=14
   (a) Synthesized and inherited attributes
   (b) Top down vs. Bottom up parsing
   (c) Symbol table
   (d) Strength reduction and Constant folding

2. (a) Define Basic blocks and flow graphs. 7

   (b) The following intermediate code turns a 10 × 10 matrix **a** into an identity matrix, build the flow graph for this code. 7

```
1)  i=1
2)  j=1
3)  t1=10*i
4)  t2=t1+j
5)  t3=8*t2
6)  t4=t3-88
7)  a[t4]=0.0
8)  j=j+1
9)  if j <= 10 goto (3)
10) i=i+1
11) if i <= 10 goto (2)
12) i=1
13) t5=i-1
14) t6=88*t5
15) a[t6]=1.0
16) i=i+1
17) if i<=10 goto (13)
```

14

3. Show that the following grammar

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa, \qquad A \rightarrow d, \qquad B \rightarrow d$

is LR(1) but not LALR(1). Also parse an input string using parsing table.

4. (a) Define peephole optimization and list the characteristics of peephole optimization. 7

(b) Define Activation tree & Activation record. 7

5. (a) Explain the various phases of a Compiler by an example. 7

(b) Consider the following grammar 3.5

$A \rightarrow ABd \mid Aa \mid a, \qquad B \rightarrow Be \mid b$

remove left recursion.

(c) Define Ambiguity and show that the following grammar is ambiguous 3.5

$E \rightarrow E + E \mid E * E \mid id$

6. (a) Define Intermediate codes and its benefits? Write Quadruples, triples and indirect triples for the following expression 7

-(x+y) * (z+c)-(x+y+z)

(b) Construct three address code for the following 7

```
if ( (a<b) && (c < d) || (a < d) )
z=x+y+z;
else
z=z+1;
```

REFNO005966

Roll No. 17410CMP021

M.Sc. Semester-II / M.C.A. Semester-IV Examination 2017-18

Subject: Computer Science / Computer Applications

Paper: CS319 / MCA401 – Compiler Design

Time: Three Hours | Full Marks: 70

Note: Attempt any five questions, including Question No. 1 which is compulsory. All the symbols and expressions have their standard meaning. Figures on the right hand side margin indicate marks.

1. (a) Explain the various phases of a compiler with a neat diagram. What are the advantages of this phase architecture of compiler? Take an appropriate assignment statement and show its translation produced by each phase. 3+2+3

   (b) What is M*N vs. M+N problem and what is the possible solution? 3

   (c) What is bootstrapping in compiler design? 3

2. (a) Distinguish the terms token, lexeme, and pattern with suitable examples. 4

   (b) How does a lexical analyzer recognize tokens? Construct a transition diagram for identifiers and unsigned numbers. 3+3

   (c) What is symbol table? How does lexical analyzer interact with symbol table? 4

3. (a) What is dangling else ambiguity? How to resolve it? Explain with example. 4

   (b) Consider the following ambiguous grammar.

   $E \rightarrow E + E \mid E / E \mid (E) \mid id$

   Write the equivalent unambiguous grammar and construct the predictive parsing table for this unambiguous grammar, and then show the moves made by predictive parser on the input id + id / id. 1+4+2

   (c) Compare the top-down and bottom-up parsers. 3

4. (a) How to construct precedence functions from precedence matrix? Explain with example. What is the positive and negative consequence of using precedence function over precedence relation? 5+2

   (b) Consider the following grammar.

   $S \rightarrow L=R \mid R$

   $L \rightarrow *R \mid id$

   $R \rightarrow L$

   Construct the collection of sets of LALR (1) items. How many states will be there in SLR, Canonical LR, and LALR parsing table for this grammar? 6+1

5. (a) What is attribute grammar? What is difference between SDD and SDT? Explain inherited and synthesized attributes with suitable examples. 1+2+3

(b) Construct a DAG and syntax tree for the following statement, and write three address codes from both. 2+2

a = (a + b) * (c + d) + (a - b) * (c + d)

(c) Assume that a[ ][ ] is a 2×3 array of integers, and c, i, j, and x are integers; and the width of an integer is 4. Translate the following statement into three-address code. 4

```
if ( x < 10 || x > 20 && x != y )
        x = c + a[i][j];
```

6. (a) Explain the optimization of basic blocks. 5

(b) What is the peephole optimization? 5

(b) Write the machine instructions for each of the following three-address instructions. 4

(i) a[j] = c (ii) if x < y goto L

---x---x---

L1: if (x > 20) then goto
L2: goto
L3: if (x != 5) then goto L7
goto L5
L4
L5: if (x < 10) then goto L7
L6
L7: x = c + a[i][j]
L8: L8

Roll Number: 063

MCA SEMESTER IV/M.Sc. SEMESTER II - EXAMINATION 2018-19
Computer Applications/Computer Science

Paper No.: CS319-Compiler Design

Full Marks: 70

Time: 3 hours

Note: Question 1 is compulsory, attempt any four questions from the remaining five questions.

1. a) Specify the functionality of linker, loader, and compiler. 2
   b) What is an ambiguous grammar? Give an example. 2
   c) Write the regular expression that derive all the strings of a's and b's where the length is at most 3. 2
   d) What is a handle in bottom up parsing? 2
   e) Define Syntax Directed Translation. 2
   f) Compute the FIRST Set of the following grammar. 2

$$E \rightarrow T\,E' \quad E' \rightarrow +T\,E' \mid \in \quad T \rightarrow F\,T' \quad T' \rightarrow * F\,T' \mid \in \quad F \rightarrow (E) \mid id$$

   g) What is symbol table? 2

2. a) Discuss the phases of a compiler indicating the inputs and outputs of each phase in translating the following statement: 8

"amount = principle + rate * 36.0"

   b) What is LL(k) grammar? Check whether the following grammar is LL(1) or not? 6

$$E \rightarrow E+T, \quad T \rightarrow T*F/F, \quad F \rightarrow id/(E)$$

3. a) Define a Parser. What is the role of grammar in Parser construction? Construct the predictive parsing table for the grammar G : 10

$$E \rightarrow E+T \mid T, \quad E \rightarrow T*F \mid F, \quad F \rightarrow (E) \mid id$$

   b) Remove the left recursion from the following grammar. $S \rightarrow Aa \mid b \quad A \rightarrow Ac \mid Sd \mid \in$ 4

4. a) Discuss the working principle of LR parser. Also construct the LR parse table for the following grammar. 8

$$E \rightarrow E+T \mid T, \quad T \rightarrow T*F \mid F, \quad F \rightarrow (E) \mid id$$

   b) Consider the grammar with the following translation rule and E is a start symbol. 6

E →	E1 # T	{E.Value = E1.Value * T.Value}
	T	{E.Value = T.Value}
T →	T1 & F	{T.Value = T1.Value + F.Value}
	F	{T.Value = F.Value}
F →	num	{F.Value = num.Value}

Compute E. Value for the root of the parse tree for the following expression :

2 # 3 & 5 # 6 & 4

5. a) What is L-attributed syntax directed definition? Give an example. 4

P.T.O.

b) Convert the following expression into three address code. $-a+b/c\uparrow d\uparrow e*f/g$ 4

c) Construct LL(1) parse table for the following expression grammar. Remove left recursion before constructing parse table if exists. 6

bexpr →bexpr or bterm | bterm
bterm →bterm and bfactor | bfactor
bfactor →not bfactor | (bexpr) | true | false

Q.5 a) What do you mean by attributed grammars? Discuss the translation scheme for converting an infix expression to its equivalent postfix form. 8

b) Explain the following code optimization techniques with examples. 6
i) Constant propagation ii) Strength reduction ii) Code motion