Date: 26/11/2014

Tech

Morning

Student Exam No.______________

GANPAT UNIVERSITY

M. Tech. Semester III (Information Technology)

Regular Examination November – December 2014

3IT302: Compiler Design

Time: 3 Hours **Total Marks: 70**

Instruction: 1. Figures to the right indicate full marks.
2. Each section should be written in a separate answer books.
3. Be precise and to the point in your answer.

Section I

Que.-1

A Describe following terms in brief — 6
1. Compiler 2. Translator 3. Error Handler 4. Augmented Grammar

B Explain in detail the process of compilation. Illustrate the output of each phase of compilation for the input "x = (y+z)* (y+z) * 1.4" — 6

OR

Que.-1

A Why it is required to check whether the grammar is ambiguous or not? Check whether the following grammar is ambiguous or not? If ambiguous then remove ambiguity. — 6
R → R+R | RR | R* | a | b

B Do as directed: — 6
1. Identify left recursion and remove it from following grammar:
P → QP | r
Q → PQ | s
2. Perform left factoring on following grammar:
A → abcde | abcdz | abce | abe | ae
S → dfgh | dfghj | dfg | df | dS
3. Remove Null production from following grammar:
Z → TUVX | VWU
T → aT | ϵ
U → b | bU
V → cV | ϵ
W → TV
X → eX | ϵ

Que.-2

A Construct M-table & check whether following grammar is LL (1) or not? — 5
S → AcB
A → aA | a
B → bB | b

B Compute First () and Follow () set for following grammar. Test whether following grammar is LL(1) or not? — 6
S → 1AB | ϵ
A → 1AC | 0C
B → 0S
C → 1

OR

gnu.inflibnet.ac.in

Que.-2

A Compute First () and Follow () set & Select() for following grammar:

S → AACB | CbB | Ba
A → da | BC
B → g | ϵ
C → f | ϵ

B Construct predictive parsing table for following grammar and show the parsing steps for string "((ab*x%xz))". **6**

I → x | (IJ) | ab
J → *IK | Ky | ϵ
K → ϵ | %IJz | b

Que.-3

A What is left recursion & left factoring? How they are eliminated? Also write and explain rules for finding FIRST() & FOLLOW() set. **6**

B Explain operator precedence parser with example. **6**

gnu.inflibnet.ac.in

Section II

Que.-4

A “If a grammar is SLR(1) then that grammar is CLR(1)” true or false? Justify your answer with example. 6

B “If a grammar is LR(0) then that grammar is LALR(1)” true or false? Justify your answer with example. 6

OR

Que.-4

A Show that the following grammar is not LR(0) but CLR(1). 8

$W \rightarrow RiRj \mid TjTi$
$R \rightarrow \epsilon$
$T \rightarrow \epsilon$

B Discuss types of three address statements (Switch statement, for & while loop) and their implementation with examples. 4

Que.-5

A Do as directed: 5

1. Apply loop jamming on the following code fragment and explain it.

```
int i, j, a[200], b[200];
for (i = 0; i < 200; i++) {
a[i] = 134;}
for (j = 0; j < 200; j++) {
b[j] = 56;}
```

2. Apply loop-invariant code motion on the following code fragment and explain it.

```
while (j < maximum - 1)
{ j = j + (4+array[k])*pi+5; }
```

B Explain S-R conflict and R-R conflict of CLR(1) & LALR(1) with example. 6

OR

Que.-5

A Describe the different data structures for symbol table implementation and compare them. 6

B Discuss the types of errors which can be encountered by all the phases of compiler. 5

Que.-6

A Construct LALR(1) parsing table for the following grammar and parse the string “*id=*id”. 7

$S \rightarrow L{=}R \mid R$
$L \rightarrow {*}R \mid id$
$R \rightarrow L$

B Construct CLR(1) parsing table for the following grammar: 5

$S \rightarrow SB \mid Ca$
$B \rightarrow Bb \mid \epsilon$
$C \rightarrow aB \mid c$

END OF PAPER

gnu.inflibnet.ac.in