

GANPAT UNIVERSITY
M. Tech SEMESTER-III [CE] EXAMINATION
NOV-DEC - 2013

3CE302 : COMPILER DESIGN

Time: 3 Hours]

[Total Marks: 70

Instructions:

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

SECTION-I

Q.1

- (A) What are the different phases of compiler? Explain each one of them. What are the advantages of breaking up the compiler functionality into two distinct stages? **(03)**
- (B) Explain classes of grammar as per Chomsky hierarchy. **(05)**
- (C) Construct DFA for the following languages: **(03)**
 $\{w \mid w \text{ has neither } 01 \text{ nor } 10 \text{ as a substring}\}$

Q.2

- (A) Give the leftmost and rightmost derivation, parse tree for **aabbaabb** string **(03)**
 $S \rightarrow aAS \mid a, A \rightarrow SbA \mid SS \mid ba$
- (B) Explain Shift-Reduce parsing techniques with block diagram **(04)**
- (C) Eliminating Use less symbols **(02)**
 $S \rightarrow aB \mid bX$
 $A \rightarrow BAd \mid bSX \mid a$
 $B \rightarrow aSB \mid bBX$
 $X \rightarrow SBD \mid aBX \mid ad$
- (C) Check using parse tree that given grammar is ambiguous or not? **(03)**
 $S \rightarrow aS \mid aSB \mid X$
 $X \rightarrow Xa \mid a$

OR

Q.2

- (A) Every unambiguous grammar is LL(1). Comment on the truth/falsehood of the statement. **(02)**
- (B) What is handle in Bottom up parser? Consider the following CFG, and show the handle of each right sentential form for the string **(a, (a, (a,a)))** **(05)**
 $S \rightarrow (L) \mid a$
 $L \rightarrow L,S \mid S$

- (C) Consider the following grammar with terminals
 $S \rightarrow aBD \mid AB \mid DAC \mid b$
 $A \rightarrow SCB \mid SABC \mid CbD \mid C \mid \epsilon$
 $B \rightarrow d \mid \epsilon$
 $C \rightarrow ADC \mid c$
 $D \rightarrow SaC \mid SC \mid fg$
 - Compute the First() and Follow() sets
 - Construct LL(1) parsing table for this grammar
 - Is this grammar LL(1)? why or why not?

(05)

Q.3 Attempt Any TWO

- (A) Explain Recursive Descent Parser with example. Also states its disadvantages.
- (B) Find First() and Follow() for the following grammar and check whether grammar is LL(1) or not
 $lexp \rightarrow atom \mid list$
 $atom \rightarrow number \mid identifier$
 $list \rightarrow (seq)$
 $seq \rightarrow lexp seq'$
 $seq' \rightarrow lexp seq' \mid \epsilon$
 Check string **(a(b(2)))(c)** is accepted or not?
- (C) Construct LALR parsing table for the following grammar
 $S \rightarrow Aa \mid bAc \mid BC \mid bBa$
 $A \rightarrow d$
 $B \rightarrow d$

(12)

SECTION-II

Q.4

- (A) Generate SLR parsing table for the following grammar and show error recovery implementation for string **id + id ++ id** and show stack content and moves.
 $A \rightarrow A + A \mid A * A \mid id$
- (B) Differentiate Predictive Parser vs Shift Reduce Parser
- (C) State the qualities of good compiler?

(05)

(03)

(03)

Q.5 Attempt Any TWO

- (A) Construct an LR(0) parsing table for the following grammar
 $S \rightarrow AA$
 $A \rightarrow aA$
 $A \rightarrow b$
- (B) Consider the following grammar
 $A \rightarrow aCDq \mid aBg \mid \epsilon$
 $D \rightarrow d \mid \epsilon$
 $B \rightarrow e \mid \epsilon$
 $C \rightarrow Ct \mid p \mid \epsilon \mid BD \mid rAb$
 Frame the transition table and Action / Goto table of the given grammar and check whether string **aedqb** is accepted or not?
- (C) Discuss R-R and S-R conflicts with examples for SLR and LR(1) parsers.

(12)

Q.6 Attempt Any TWO

(A) Consider the following code fragments. Generate the 3AC for it.

```
(i)
Count = 0;
Result = 0;
While (count++ < 20)
{
    increment = s+count;
    result+ = increment;
}
```

```
(ii)
Sum = 0;
For (I=1;I<=0;I++)
    Sun = sum + a[I] + b[I];
```

(B) What is locality of reference? Optimize the following code fragment

```
P = 5;
Q = 10;
For (j=0;j<10;++j)
    For (i=0;i<10;++i)
    {
        Y[j] = x[j] - x[Q];
        Z[i] = x[i] - x[P]
    }
```

(C) State and compare following set of cod optimization techniques

- A. Loop fussion vs. Loop fission
- B. Loop splitting vs. Loop unwinding

----- END OF PAPER -----

gnu.inflibnet.ac.in