Enrollment No._____

GANPAT UNIVERSITY

B. Tech SEMESTER-VII [CE-IT] EXAMINATION

NOV-DEC - 2014

2CE702/2IT702 : 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) List different phases of compiler? What are the advantages of breaking up the compiler functionality into two distinct stages? Explain the functions of symbol table manager and error handler. **(05)**

(B) Explain classes of grammar as per Chomsky hierarchy. **(03)**

(C) What is system software? Explain how Compiler, Assembler, Interpreter, Linker, Loader, Device driver are different then various application software? **(03)**

Q.2

(A) Consider the grammar S → aSbS | bSaS | ε. **(05)**

i. Show that this grammar is ambiguous by constructing two different leftmost derivations for the sentence ***abab***.

ii. Construct the corresponding right most derivations for ***abab***.

iii. Construct the corresponding parse tree for ***abab***.

iv. What language does this grammar generates?

(B) Explain Predictive parsing techniques with block diagram. **(04)**

(C) Differentiate DFA vs. NFA. Design a FA which accepts set of strings containing exactly five 1's at any position in every string over alphabet $\Sigma = \{0, 1\}$. **(03)**

OR

Q.2

(A) i) Remove the useless symbol from the given CFG:
S → aB | bX
A → Bad | bSX | a
B → aSB | bBX
X → SBD | aBx | ad

ii) Find and remove left recursion from the given CFG:
S → Bb | a
B → Bc | Sd | e **(05)**

gtuinfinet.ac.in

(B) Construct a predictive parsing table (M-table) for given CFG: **(05)**

S → aBDh
B → cC
C → bC | ε
D → EF
E → g | ε
F → f | ε

(C) Design a CFG for the language L(G) = $\{a^n b^{2n} \mid n \geq 0\}$ **(02)**

Q.3 Attempt Any TWO **(12)**

(A) What is recursive-decent parsing? Write a procedure code for the recursive-decent parsing of the following CFG

E → TR
R → +TR | -TR | ε
T → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

(B) Eliminate the left recursion from the following CFG

S → (L) | a
L → L,S | S

i. Construct the predictive parser for CFG generated after elimination.
ii. Show the behavior of parser on the string (a, (a, a))

(C) Show that the following CFG

S → Aa | bAc | Bc | bBa
A → d
B → d

is LR(1) but not LALR(1).

SECTION-II

Q.4

(A) Generate SLR parsing table for the following grammar and show error recovery implementation for string **3 + 8 +*+ 9** and show stack content and moves. **(05)**

X → X + X | X * X | Y
Y → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

(B) Differentiate Predictive Parser vs. Shift-Reduce Parser **(03)**

(C) What do you understand by scoping in the symbol table? Give the difference between scope-by-numbering and scope-by-location. **(03)**

Q.5 Attempt Any TWO **(12)**

(A) Construct an SLR parsing table for the following CFG

A → aAa | bAb | ba

(B) Construct an LALR(1) parsing table for the following CFG

S → Ba | bBc | dc | bda
B → d

(C) Discuss R-R and S-R conflicts with examples for SLR and LR(1) parsers. Why S-S conflict is not possible?

gtu.infinet.ac.in

Q.6 Attempt Any TWO **(12)**

(A) Consider the following code fragments. Generate the 3AC for it assuming 'a' is allocated static storage.

```
begin
    for i:= 1 to n do
        for j := 1 to n do
        c[i,j] := 0;
    for i:= 1 to n do
        for j := 1 to n do
            for k := 1 to n do
                c[i,j] := c[i,j] + a[i,k] x b[k,j]
end
```

(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) Explain following code optimization techniques in brief.

i. Loop fission
ii. Loop unrolling
iii. Loop unswitching
iv. D. Peephole optimization

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

gnu.inflibnet.ac.in