MCA/IVS/403/12

MCA 4th Semester Examination, 2012

COMPILER CONSTRUCTION

PAPER— 403

Full Marks : 100

Time : 3 hours

Answer **Q. No. 1** and any **four** from the rest

The figures in the right-hand margin indicate marks

Candidates are required to give their answers in their own words as far as practicable

Illustrate the answers wherever necessary

1. Answer any *seven* questions : 2 × 7

(*a*) What is the difference between a compiler and an interpreter ?

(*b*) What do you mean by token ?

(*c*) What are the disadvantage of top down parsing.

(*d*) What is Basic Block ?

(*e*) What is flow graph ?

(*f*) Define Annotated parse tree ?

(*g*) Why LR parsing is attractive ?

(*h*) "Every SLR grammar is unambiguous, but every unambiguous grammar is not a SLR grammar."— Explain.

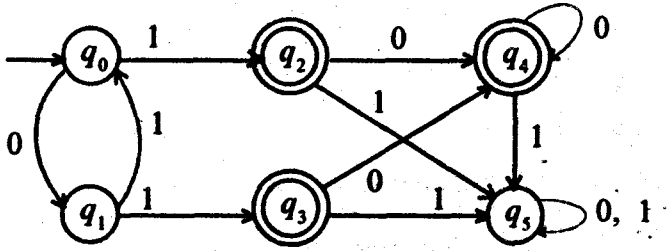
(*i*) What is DAG ?

(*j*) What is cross-compiler ?

(*k*) In which phase of compilation process type errors are defected ?

2. (*a*) Convert the regular expression (a / b)* aba directly to DFA using nullable, first pos, last pos and follow pos functions.

(*b*) Construct the minimum-state DFA. 9 + 5



3. Compute the specified first and follow sets for the following grammar, then construct LL(1) parse table for the grammar. Use the parse table to parse the input string albbs. This string may not be a legal string in the Language.

$$S \rightarrow aS \mid Ab$$
$$A \rightarrow CDE \mid \epsilon$$
$$C \rightarrow CS \mid \epsilon$$
$$D \rightarrow dS \mid \epsilon$$
$$E \rightarrow eS \mid$$

10 + 4

4. Given a grammar with the following rules :

$$S \rightarrow A\#$$
$$A \rightarrow bB$$
$$B \rightarrow cC$$
$$B \rightarrow cCe$$
$$C \rightarrow dA$$
$$A \rightarrow a$$

(*a*) generate the set of LR(1) items.

(*b*) Is the grammar LR(1) ? If not, why ? 14

5. (*a*) Construct an LALR(1) parsing table for the following grammar :

$$D \rightarrow L : T$$
$$L \rightarrow L_1, \text{ id} \mid \text{id}$$
$$T \rightarrow \text{integer}.$$

(*b*) Define symbol table. 10 + 4

6. (*a*) Describe three address code for intermediate code generation phase.

(*b*) Generate three address code for the following program fragment :

```
While (A < C and B > D) do
if A = 1 then C = C + 1
 else
  while A <= D do
       A = A + 3.
```

(*c*) Show quadruple and triple representation of $x = (a + b) * - c \mid d$ 3 + 6 + 5

7. (*a*) Consider three-address code below :

(1) f = 1;
(2) i = 2;
(3) if i <= x goto (8)
(4) f = f * i
(5) t1 = i + 1
(6) i = t1
(7) go to (3)
(8) go to calling program.

Partition these into Basic Blocks also show the corresponding flow graph of Basic Block.

(*b*) Consider the following code and show the DAG. Also show the final code after common sub-expressions elimination.

(1) S1 := 4 * I
(2) S2 :=adds (A) – 4
(3) S3 := S2 [S1]
(4) S4 := 4 * I
(5) S5 :=addr (B) – 4
(6) S6 :=S5 [S4]

(7) S7 :=S3 * S6

(8) S8 :=PROD + S7

(9) PROD :=S8

(10) S9 := I + 1

(11) I = S9

(12) if I <= 20 go to (1). 7 + 7

8. Write short note (any *two*) : 7 × 2

(*i*) Syntax directed defination and translation

(*ii*) NFA to DFA conversion

(*iii*) Lex (A lexical Analyzer Generator).

[*Internal Assessment* : 30 Marks]