B.Tech. SEM -VII (Computer) 2014 Course (CBCS) : WINTER - 2018

SUBJECT : COMPILER DESIGN

Date: Friday
Day: 30/11/2018

Time: 02.30 PM TO 05.30 PM
Max. Marks: 60

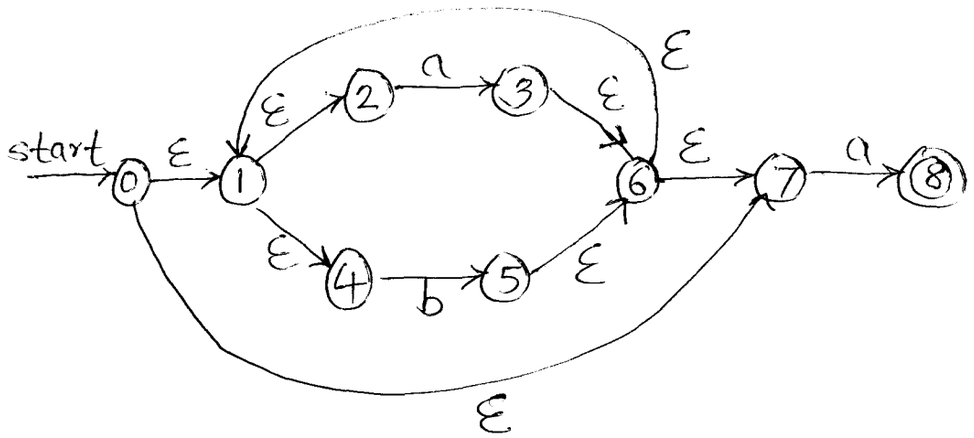
W-2018-2539

N.B.:

1) All questions are **COMPULSORY**.
2) Figures to the right indicate **FULL** marks.
3) Draw neat and labeled diagram **WHEREVER** necessary.
4) Assume suitable data, wherever necessary.

Q.1 Define Lexemes, token and pattern with example. **(10)**

Consider the NFA with ε-moves shown in the figure below:
Use subset construction algorithm to convert NFA to DFA.



OR

Explain Boot strapping and Incremental compiler. **(10)**

Construct NFA with ε for $RE\ (a/b)^*.abb$ and then convert into a DFA using subset construction algorithm.

Q.2 Explain augmented grammar with example. **(10)**

Construct the nonempty sets of $LR(1)$ items for the following grammar:

$S \rightarrow A$

$A \rightarrow AB \mid \varepsilon$

$B \rightarrow a\ B \mid b$

OR

Compare TOP down parsing and BOTTOM up parsing. **(10)**

Test whether the grammar is LL(1) or not and construct a Predictive Parsing table for it

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \varepsilon$

$B \rightarrow \varepsilon$

P.T.O.

Q.3 Explain S – Attribute and L – Attribute definition. **(10)**

Write a SDT scheme for:

i) if E then S

ii) if E then S else S

iii) do S while E

where E can be arithmetic or condition expression, S is simple assignment statement.

OR

Consider the following assignment statement **(10)**

$$X = (M + N) * (Q + R)$$

Draw the annotated parse tree for the generation of three address code.
Construct DAG for the expression:

$$(((a + a) + (a + a)) + ((a + a) + (a + a)))$$

Q.4 **a)** Explain strength reduction in brief. How strength reduction is applied to induction variable? **(05)**

b) Explain Data flow analysis framework. **(05)**

OR

a) Explain Depth-first Ordering in Iterative algorithms. **(05)**

b) Explain the term 'expression aliases'. How do you deal with it? **(05)**

Q.5 **a)** Consider the following source code: **(06)**

```
begin
    prod:=0;
    i:=1;
    do begin
        prod:= prod +a[i] * b[i];
        i:=i+1;
       end
    while i<=20
end
```

Represent three-address statements performing computation on target machine.

b) Explain in detail tree-translation scheme. **(04)**

OR

a) Explain in detail DAG representation of Basic Blocks. **(06)**

b) Explain the principles of Dynamic Programming. **(04)**

Q.6 **a)** Explain in detail GCC architecture with neat diagram. **(05)**

b) Explain different levels of optimizations of GCC. **(05)**

OR

a) Discuss any five command line options of GCC. **(05)**

b) Explain in detail preprocessing unit of GCC. **(05)**

* * * * *