H Roll No. .....

TCS–601

B. TECH. (CSE) (SIXTH SEMESTER)

MID SEMESTER

EXAMINATION, March, 2024

COMPILER DESIGN

Time : 1½ Hours

Maximum Marks : 50

Note : (i) Answer all the questions by choosing any *one* of the sub-questions.

(ii) Each sub-question carries 10 marks.

1. (a) Consider the given single line source code :

$$a = b + c * d;$$

Show the action taken by every phase of Compiler in order to convert the source code from high level to machine level.

(CO1)

P. T. O.

OR

(b) Consider the following grammars : (CO1)

(i) $S \rightarrow ABCD$

$A \rightarrow a/\varepsilon$

$B \rightarrow b/\varepsilon$

$C \rightarrow c$

$D \rightarrow d/\varepsilon$

(ii) $S \rightarrow aBDh$

$B \rightarrow cC$

$C \rightarrow bC/\varepsilon$

$D \rightarrow EF$

$E \rightarrow g/\varepsilon$

$F \rightarrow f/\varepsilon$

For the above grammar productions, compute the FIRST() and FOLLOW() set.

2. (a) (i) Define symbol table. Illustrate how the given single line of source code is going to store inside the symbol table ?

int arr[5] = {10, 11, 12, 13, 14};

(ii) Design a LEX program to display the integer number, floating point number and a valid identifier. (CO1)

OR

(b) Identify and remove left recursion and left factoring from the following grammar :

(i) $S \rightarrow A$

$A \rightarrow Ad/Ae/aB/ac$

$B \rightarrow bBc/f$

(ii) $X \rightarrow XSb/Sa/b$

$S \rightarrow Sb/Xa/a$

(iii) $S \rightarrow a/ab/abc/abcd$

(iv) $S \rightarrow aAd/aB$

$A \rightarrow a/ab$

$B \rightarrow ccd/ddc$

(CO1)

3. (a) Consider the following grammar :

$E \rightarrow E + T/T$

$T \rightarrow T^* F/F$

$F \rightarrow (F)/a/b/c/id$

For the above grammar construct the predictive parsing table. (CO2)

OR

(b) At lexical analysis phase, illustrate how the tokens are being generated and recognized ? Identify and count the number of tokens present in the given C code. (CO2)

```
void main()
{
  int a,b,c,d;
  b=10;
  c=20;
  d=30;
// Compute a=b+c*d
  a= b+c*d;
// printing the value of a.
  printf("%d", a);
  getch();
}
```

4. (a) What do you understand by input buffer scheme ? Illustrate the benefits of two buffer scheme over one buffer scheme. For the given source code show, how one buffer scheme works ? (CO2)

```
int a = 20, b = 30, c = 40;
```


OR

(b) What do you mean by ambiguous grammar ? To verify whether the given grammar is ambiguous or not ? (CO2)

$E \rightarrow E + E/E * E/id$

5. (a) Consider the following grammar :

$S \rightarrow AA$

$A \rightarrow aA/b$

To check whether the given grammar is SLR(1) or not. (CO2)

OR

(b) Consider the following grammar.

$S \rightarrow AaAb/BbBa$

$A \rightarrow \varepsilon$

$B \rightarrow \varepsilon$

To check whether the given grammar is CLR(1) or not. (CO2)