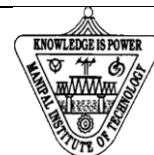


Reg. No.									
----------	--	--	--	--	--	--	--	--	--



MANIPAL INSTITUTE OF TECHNOLOGY
(Constituent Institute of MANIPAL University)
MANIPAL-576104



SIXTH SEMESTER B.E. (CS&E) DEGREE EXAMINATION

Subject: LANGUAGE PROCESSORS

Code: CSE- 302

DATE:

TIME: 3 HOUR

MAX.MARKS: 50

Instructions to the Candidates

- Answer ANY FIVE full Questions.
- Missing data can be suitably assumed
- Answer should be clear and concise in point form

- 1A** What are the parameters used in “Chomsky Hierarchy” to classify the Languages? Explain briefly about the programs related to compilers. **4M**
- 1B** Use Thompson’s Construction to convert regular expression $(aa | b)^* (a | bb)^*$ into a NFA and convert NFA into a DFA using subset construction. **6M**
- 2A** Check whether the given grammar is an ambiguous or not. Justify your answer . **4M**
- $S \rightarrow aB | bA$
 $A \rightarrow aS | bAA | a$
 $B \rightarrow bS | aBB | b$
- 2B** Construct the LL(1) parsing table for following grammar using FIRST and FOLLOW sets. **6M**
- $Statement \rightarrow Ifstmt | other$
 $Ifstmt \rightarrow if (exp) Statement Elsepart$
 $Elsepart \rightarrow else Statement | \epsilon$
 $exp \rightarrow 0 | 1$
- 3A** Write the Algorithm for LR(0) Parsing and construct the DFA of LR(0) Items for the given Grammar. **4M**
- $S \rightarrow AaAb | BbBa$
 $A \rightarrow \epsilon$
 $B \rightarrow \epsilon$
- 3B** Construct LALR(1) Parsing table for the following grammar **6M**
- $S \rightarrow CC$
 $C \rightarrow eC | d$

- 4A** Consider the following grammar, where numbers may be octal or decimal, indicated by a one character suffix **o**(for octal) or **d**(for decimal): **4M**

BasedNum \rightarrow Num Basechar
Basechar \rightarrow **o** | **d**
Num \rightarrow Num Digit | Digit
Digit \rightarrow 0|1|2|3|4|5|6|7|8|9

Write the attribute grammar for **base** and **value**.

- 4B** For the following Three-Address code : **6M**

- I. Construct the Flow graph by identifying basic blocks.
- II. Construct the DAG for the basic block where Optimisation needs to be done.
 - 1) sum = 0
 - 2) i = 0
 - 3) t1 = 4 * i
 - 4) t2 = a[t1]
 - 5) t3 = sum + t2
 - 6) sum = t3
 - 7) t4 = i + 1
 - 8) i = t4
 - 9) if i < 10 goto (3)

- 5A** Explain the statements in Assembly Language with syntax. **4M**

- 5B** Explain briefly the different *type constructors* used in type expressions **4M**

- 5C** What are the disadvantages of recursive descent parser? **2M**

- 6A** Explain how a Hash Table deals with collision, if it is chosen for Symbol table. **4M**
Show the status of Symbol Table contents after processing the declarations of the body of function **f**.

```
int i, j;  
int f( int size)  
{ char i, temp;  
.....  
{ double j;  
.....  
}  
.....  
{ char *j;  
.....  
}}  
}
```

- 6B** Define EXTRN and ENTRY statements **2M**

- 6C** Differentiate between **4M**

- I. Top down parsing and Bottom up parsing
- II. P-Code and Three-Address code

