

Elective - I : Compiler Design

P. Pages : 3

Time : Three Hours



TKN/KS/16/7590

Max. Marks : 80

- Notes :
1. All questions carry marks as indicated.
 2. Solve Question 1 OR Questions No. 2.
 3. Solve Question 3 OR Questions No. 4.
 4. Solve Question 5 OR Questions No. 6.
 5. Solve Question 7 OR Questions No. 8.
 6. Solve Question 9 OR Questions No. 10.
 7. Solve Question 11 OR Questions No. 12.
 8. Due credit will be given to neatness.
 9. Illustrate your answers whenever necessary with the help of neat sketches.

1. a) Explain with neat diagram various stages or phases of compiler. Also, explain analysis and synthesis tasks performed by compiler. **8**
- b) Explain the following with appropriate example :- **6**
 - i) Bootstrapping
 - ii) Cross compiler.

OR
2. a) Explain the term token. Find the tokens and count the number of tokens used or generated by the following program fragment :- **6**
scanf (" % d % d % f ", & marks, & total, & percentage);
- b) Explain the following compiler writing tools :- **8**
 - i) FLEX tool
 - ii) YACC tool.
3. a) Construct a LL(1) parsing table and test whether the given grammar is LL(1) or not. **7**
 $A \rightarrow NiNs \mid a$
 $N \rightarrow \epsilon \mid b \mid G$
 $G \rightarrow \epsilon \mid j$
- b) Find the FIRST and FOLLOW sets for the grammar G_1 , where **7**
 $G_1 = \{P \rightarrow QNQ \quad Q \rightarrow tQ \mid \epsilon \quad N \rightarrow bN \mid \epsilon\}$ Construct parsing table for CLR (1) and test whether the grammar is CLR (1) or not.

OR
4. a) With neat diagram, explain the working of LR parsers. And differentiate between , LR(0), LALR, CLR(1) parsers. **6**
- b) What do you mean by viable prefix? Find the viable prefixes for the string (d(d,d)) for the grammar given below :- **8**
 $R \rightarrow d \mid \wedge \mid (D)$
 $D \rightarrow D, R \mid R$
 Show the parsing actions generated by shift-reduce parser for the string $W = (d(d,d))$.
5. a) What do you mean by semantic action or semantic rules used for SDT? Explain with suitable example. Write SDT for evaluation of Boolean expression. **6**

- b) For the given program fragment, obtain TAC and draw the control flow graph for the same. 7
- ```

begin
 add := 0; j := 1;
 do
 begin
 add := add + R[j] * T [j];
 j := j+1;
 end
 while j <= 20
end

```

**OR**

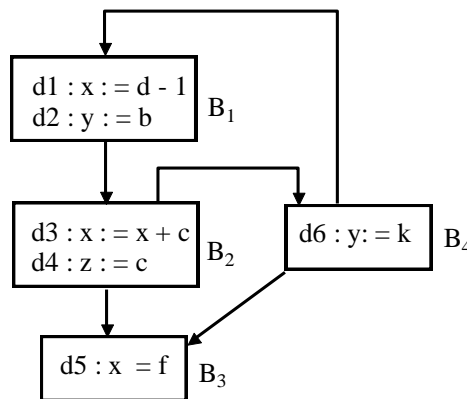
6. a) State the difference between syntax tree and annotated parse tree. Generate Annotated parse tree and syntax tree for the given expression :- 6
- $s + (r * (r-d)) + ((r-d) * d)$

- b) Write the translation scheme for the given array reference and write TAC for the given multidimensional array, arrays are statically allocated and size of B is  $10 * 10 * 20$ , size of C is 20, size of D is  $20 * 10$  and BPW is 8. 7
- $B[i,j,k] := C [i] + D [i,j]$

7. a) Write four properties for good error diagnosis. 4
- b) Explain data structures used for symbol table organization. 5
- c) Explain heap-allocation storage strategy. 4

**OR**

8. a) Explain various error recovery techniques used by compiler. 7
- b) Explain the syntactic errors and semantic errors with one example of each. 6
9. a) For the given graph, compute IN and OUT set and U-d chaining information. 8



- b) Write the steps or algorithm for partitioning a sequence of three-address statements into basic blocks. 5

**OR**

10. a) What do you mean by reducible flow graph? Is it necessary to detect loops in a reducible flow graph? Justify your answer. 5
- b) Explain, with suitable example, loop invariant statements or computation, elimination from the code. Explain its importance in loop optimization. 8

11. a) Write about the following peephole optimizations :- 8  
i) Redundant loads and stores            ii) Unreachable code.  
iii) Use of machine idioms                iv) Reduction in strength.  
Write one example of each.

- b) Explain the working or implementation of labeling algorithm for the statement given below :- 5  
 $S := n - t * (r/f)$

**OR**

12. a) Explain the evaluation of number of registers to be allocated for the expression given below. And generate the code using code generation procedure. 8  
 $S := - (z+y) + x + (x *(z+y))$   
 $\quad + ((z-y) * t)$

- b) Explain the DAG. What are advantages of using DAG as compared to syntax tree. 5

\*\*\*\*\*

