

Code No: V3223

R07

Set No: 1

III B.Tech. II Semester Supplementary Examinations, April/May - 2013

COMPILER DESIGN

(Computer Science & Engineering)

Time: 3 Hours**Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

1. a) Write a LEX input file that will produce a program that Capitalizes all comments in C?
b) Draw a DFA that accepts the four reserved words **case**, **char**, **const** and **continue** from the C language? [10M+6M]
2. a) Compare and contrast context free grammars and regular expressions. [6M]
b) Given the following grammar $A \rightarrow (A) A | \epsilon$
(i) Construct FIRST and FOLLOW sets for the non terminal A?
(ii) Show this grammar is LL(1)?
(iii) Construct predictive parse table for the above grammar? [10M]
3. a) What are the common conflicts that can be encountered in shift – reduce Parsers [6M]
b) What is YACC? Explain how YACC parser resolves ambiguity with suitable example [10M]
4. a) What is a syntax tree? Write syntax directed definition for constructing a syntax tree for the expression **id+id*num**. The grammar is given below.
 $E \rightarrow E + T | E - T | T$ $T \rightarrow (E) | id | num$ [12M]
b) Describe the role of intermediate code generator in compilation Process? [4M]
5. a) Explain how symbol table is used in compilation and how to implement a symbol table.
b) Explain heap storage allocation scheme with suitable example. [8M+8M]
6. a) What are some of the errors that a compiler should detect? What course of action should the compiler take when it detects an error? [6M+10 M]
b) What is intermediate code? Explain different intermediate forms and their implementations?
7. a) Specify the necessary and sufficient conditions for performing. [2M+2M+2M+2M]
(i) constant propagation (ii) dead code elimination
(iii) Code motion (iv) induction variable elimination
b) Explain live variable analysis with a suitable flow graph. [8M]
8. a) What is peephole optimization? Explain different peephole optimization techniques?
b) Explain the various issues in the design of code generators. [8M+8M]

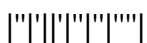


R07

Set No: 2

Code No: V3223

7. a) What are *basic blocks* and *flow graphs*? Identify the basic blocks and draw the flow graph for the following three-address code segment-
- (i) $PROD = 0$
 - (ii) $I = 1$
 - (iii) $T_1 = 04 * I$
 - (iv) $T_2 = \text{addr}(A) - 4$
 - (v) $T_3 = T_2[T_1]$
 - (vi) $T_4 = \text{addr}(B) - 4$
 - (vii) $T_5 = T_4[T_1]$
 - (viii) $T_6 = T_3 * T_5$
 - (ix) $PROD = PROD + T_6$
 - (x) $I = I + 1$
 - (xi) if $I \leq 20$ goto (3)
- [8M]
- b) What is a *Directed Acyclic Graph* and what are its uses? Explain with suitable example? [8M]
8. a) What do the *register descriptors* contain? How are these useful in code generation?
- b) Give an example to show how reordering of intermediate code statements can improve code generation. [8M+8M]



R07**Set No: 3****Code No: V3223**

7. a) What are *basic blocks* and *flow graphs*? Identify the basic blocks and draw the flow graph for the following three-address code segment
- (i) $PROD = 0$
 - (ii) $I = 1$
 - (iii) $T_1 = 04 * I$
 - (iv) $T_2 = \text{addr}(A) - 4$
 - (v) $T_3 = T_2[T_1]$
 - (vi) $T_4 = \text{addr}(B) - 4$
 - (vii) $T_5 = T_4[T_1]$
 - (viii) $T_6 = T_3 * T_5$
 - (ix) $PROD = PROD + T_6$
 - (x) $I = I + 1$
 - (xi) if $I \leq 20$ goto (3) [8M]
- b) What is a *Directed Acyclic Graph* and what are its uses? [8M]
8. a) What do the *register descriptor* contain? How are these useful in code generation? [8M]
- b) Give an example to show how reordering of intermediate code statements can improve code generation. [8M]



R07**Set No: 4****Code No: V3223**

III B.Tech. II Semester Supplementary Examinations, April/May - 2013

COMPILER DESIGN

(Computer Science & Engineering)

Time: 3 Hours**Max Marks: 80**

Answer any FIVE Questions

All Questions carry equal marks

1. a) Explain the role of lexical analyzer? And also differentiate pattern, lexeme and token?
b) Discuss how compiler design process depends on syntax, semantics and pragmatics of a programming language. [8M+8M]
2. a) What is ambiguity? Explain how to resolve ambiguity from context free grammars with suitable examples? [6M]
b) What is left recursion? Remove left recursion for the following grammar? [6M]
 $E \rightarrow E+E \mid E-E \mid E * E \mid E \setminus E \mid (E) \mid id$
c) Discuss the problems associated with top -down parsers. [4M]
3. a) What is handle pruning? Explain how to identify handles in bottom-up parsing. [6M]
b) What is YACC? Explain how YACC parser resolves ambiguity with suitable example. [10M]
4. a) What is Syntax Directed Translation (SDT)? [8M]
b) Explain the significance of intermediate code in compiler design process and Convert the following statement into TAC? $(a + b) * c + (d * a) - b$ [8M]
5. a) Discuss the role of symbol table in compilers and explain its implementation in Hash tables?
b) Compare and contrast DAG and syntax trees? [10M+6M]
6. a) What are the principle sources of code optimization? Explain the significance of Data flow analysis in code optimization with suitable examples? [10M]
b) Explain block partition algorithm? [6M]
7. a) Why is *code optimization* done by compilers but not by assemblers? [6M]
b) What are some of the optimizations that can be performed by a compiler?
Which kind of optimization is more effective inside loops - space optimization? Or time optimization? Why [10M]
8. Write in detail about :
a) Issues in the design of a code generator [8M]
b) Code generation algorithm [8M]

