

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 45895

5 Year M.Sc. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2014.

Sixth Semester

Software Engineering

XCS 362/10677 SWE 42 — COMPILER DESIGN

(Common to 5 Year M.Sc. Information Technology and M.Sc. Computer Technology)

(Regulation 2003/2010)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What are the three general approaches to the implementation of a Lexical Analyzer?
2. Give a regular expression for recognizing the set of all identifiers.
3. What is meant by parser generator?
4. What is syntax error? Give example.
5. What are the advantages of symbol table?
6. Define the term : dangling reference.
7. What is backpatching?
8. Draw the syntax tree for : “if $a > b + c$ then $max = a$; else $max = b + c$ ”.
9. What is peephole optimization?
10. What is known as dead code elimination?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Construct an NFA using Thompson's construction rule for the following regular expression. $(a / b)^* abb (a / b)^*$ (10)
- (ii) Explain the role of lexical analyzer. (6)

Or

- (b) Construct a minimal state DFA for the regular expression $(\epsilon / a^* / b^*)^*$ and $(a / b)^*$. Show that they are equal or not. (16)
12. (a) (i) Construct the predictive parser for the following grammar : (12)
- $$S \rightarrow a \mid \uparrow \mid (T)$$
- $$T \rightarrow t, S \mid S.$$
- (ii) Show the behaviour of the parser in the sentence (a, (a, a)). (4)

Or

- (b) Eliminate the left recursion from the grammar :
- $$S \rightarrow (L) \mid a$$
- $$S \rightarrow L, S \mid s$$
- Construct a predictive parser for the grammar.
13. (a) Explain briefly about the following storage allocation strategies :
- (i) Static allocation
- (ii) Stack allocation
- (iii) Heap allocation.

Or

- (b) (i) Discuss about the symbol table mechanism : Linear list, Hash table. (8)
- (ii) Explain about the issues of source languages and its activations at run time. (8)
14. (a) (i) Write the translation scheme for translating assignment statements having an array references to three address code. (8)
- (ii) Translate the following assignment statement into three address code using the translation scheme.
- $$A[i, j] : B[i, j] + C[A[k, l]] + D[i, j] \quad (8)$$

Or

- (b) Translate the executable statements of the following program :

```
main( )
{
    int i;
    int a[10];
    i = 1;
    while (i <= 10)
    {
        a[i] = 0;
        i = i+ 1; }
    }
```

Convert into :

- (i) Syntax tree
 - (ii) Postfix notation
 - (iii) Three address code.
15. (a) (i) Explain briefly about the generic issues in the design of code generator. (8)
- (ii) Explain about the principle sources of optimization in detail. (8)

Or

- (b) (i) Consider the following fragment of source code : (8)

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

Convert the source code into three address statement forms of a block.

- (ii) Discuss about the two classes of transformation applied on the basic blocks. (8)