

2002 SCHEME

USN

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|

CS664

Sixth Semester B.E. Degree Examination, December 2010 Compiler Design

Time: 3 hrs.

Max. Marks:100

Note: 1. Answer any FIVE full questions.
2. Any missing data may be suitably assumed.

- 1
 - a. Explain the different phases of a compiler, with a neat diagram. (08 Marks)
 - b. Explain the token generators and token recognizers, with a simple example. (04 Marks)
 - c. Write a Lex program to recognize the string $a^n b$. (08 Marks)

- 2
 - a. Define the following, with examples :
 - i) Ambiguous grammar
 - ii) Derivation tree. (06 Marks)
 - b. Show that the following grammar is ambiguous
 $S \rightarrow i C t S \mid i C t S c S \mid a$
 $C \rightarrow b$
 Write the unambiguous grammar for the same. (08 Marks)
 - c. Explain with an example, the recursive descent parser with backtracking. (06 Marks)

- 3
 - a. Bring out the differences between top-down and bottom-up parsing methods. (03 Marks)
 - b. Compute FIRST () and FOLLOW () symbols for the following grammar and find if the grammar is LL(1)
 $E \rightarrow TE'$
 $E' \rightarrow + TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow * FT' \mid \epsilon$
 $F \rightarrow (E) / id.$ (08 Marks)
 - c. Given the following precedence relation table, parse the string $id + id * id$

| | | | | |
|----|----|---|---|----|
| | id | + | * | \$ |
| id | | > | > | > |
| + | < | > | < | > |
| * | < | > | > | > |
| \$ | < | < | < | |

- 4
 - a. Explain with an example, the stack implementation of a shift reduce parser. (09 Marks)
 - b. Define the following, with examples. (10 Marks)
 - i) Synthesized attribute
 - ii) Inherited attribute
 - iii) Annotated parse tree
 - iv) Dependency graph. (10 Marks)

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

- 5 a. What is an activation record? Explain the purpose of different fields in an activation record. (10 Marks)
- b. Explain the following storage allocation strategies.
i) Static allocation
ii) Heap allocation. (10 Marks)
- 6 a. Define 3-address statement and list the types of 3-address statement. (08 Marks)
- b. Define the terms quadruples, triples and indirect triples. Give their representation for the assignment statement $A = B * (C + D)$ by generating an appropriate 3-address code. (12 Marks)
- 7 a. Explain the issues in the design of a code generator. (12 Marks)
- b. Generate the code for the following three address statement, using the code generation algorithm
 $t = a - b$
 $u = a - c$
 $v = t + u$
 $d = v + u$
with d live at the end. (08 Marks)
- 8 a. Optimize the following code
Product = 0
i = 1
do
 product = product + A[i] * B[i]
 i = i + 1
while (i <= 20). (12 Marks)
- b. Write an algorithm to construct a DAG from a basic block. (08 Marks)
