USN | | | | | | | | | | |

06CS63

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

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. Explain with neat diagram, the various phases of a compiler. Mention the input and output for each phase. **(08 Marks)**

b. Define static and dynamic scoping. Explain the working and output of the following programming segment if scoping used is static and dynamic:

```
BEGIN
        Boolean b : = true
        Procedure P ;
        BEGIN
                Print (b) ;
        End ;
    BEGIN
        Boolean b : = false ;
        Call P ;
    End ;
END;
```

(04 Marks)

c. With an example, explain the use and coordination between 'LEX' and 'YACC' the compiler writing tools. **(08 Marks)**

2 Consider the grammar:

$E \rightarrow 5 + T | 3 - T$

$T \rightarrow V | V * V | V + V$

$V \rightarrow a | b$

a. What is the use of left factoring? Do the left factoring for the above grammar. **(04 Marks)**

b. Write an algorithm to obtain the FIRST and Follow table. Obtain FIRST and Follow table for the above grammar. **(08 Marks)**

c. Write an algorithm to construct the predictive parsing table. Construct predictive parsing table for the above grammar. **(08 Marks)**

3 Consider the grammar:

$S \rightarrow E\#$

$E \rightarrow E - T$

$E \rightarrow T$

$T \rightarrow F \uparrow T$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow i$

a. Write the algorithm to construct basic finite state control m/c for SLR (1) and action α goto functions entries. **(08 Marks)**

b. Construct the following for the above grammar:

i) Basic finite state control.

ii) SLR (1) parsing table containing action and goto function entries. **(12 Marks)**

4 Consider the grammar:

$G \rightarrow S$
$S \rightarrow E = E$
$S \rightarrow f$
$E \rightarrow T$
$E \rightarrow E + T$
$T \rightarrow f$
$T \rightarrow T * f$

when terminal symbols are {=,+,*,f}

a. Write an algorithm to construct finite state control for LR(1) parser. **(08 Marks)**

b. Construct LR(1) finite state control and explain the algorithm to construct parsing table containing action α goto function entries. **(12 Marks)**

PART – B

5 a. With an example, explain the concept of syntax directed definition. **(08 Marks)**

b. Write the grammar and syntax directed definitions for a simple desk calculator and show annotated parse tree for the expression (3+4)*(5+6). **(12 Marks)**

6 a. What is DAG? Construct a DAG for the following expression, $a + a * (b - c) + (b - c) * d$. **(04 Marks)**

b. With an example, explain the various formats of intermediate code. **(10 Marks)**

c. Write quadruple representation for, $a + a * (b - c) + (b - c) * d$. **(06 Marks)**

7 a. Explain the run time storage scheme for C++-language. Give the structure of activation record and explain with suitable example. **(12 Marks)**

b. Explain the design goals for garbage collectors. **(08 Marks)**

8 a. Discuss the following terms:

i) Basic blocks
ii) Next-use information
iii) Flow graph **(10 Marks)**

b. Explain the following code optimization with example:

i) Finding local common sub expression.
ii) Dead code elimination. **(10 Marks)**

* * * * *

USN | | | | | | | | | | |

06IS662

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

Time: 3 hrs. Max. Marks:100

Note: *Answer any FIVE full questions, selecting at least TWO questions from each part.*

PART – A

1 a. Explain the different phases of a compiler, with a neat diagram. (10 Marks)

b. Explain the different applications of compiler technology. (06 Marks)

c. Provide transition diagram to recognize : pipe, pet, item, petrol. 04 Marks)

2 a. Consider the following grammar:

$R \rightarrow R'|'R \mid RR \mid R* \mid (R) \mid a \mid b \mid c$

Note : Here 'I' is a 'or' symbol and not a separator between alternatives.

Check if the grammar is ambiguous or not. If it is an ambiguous grammar, construct an equivalent unambiguous grammar. (08 Marks)

b. Consider the following grammar:

$E \rightarrow E * T \mid T$

$T \rightarrow id + T \mid id$

i) Check if it is an LL(1) grammar.

ii) Show the sequence of moves made by the parser for w = id + id * id. (12 Marks)

3 a. Define : i) Handle, ii) Handle pruning. (06 Marks)

b. Explain LR parsing algorithm. (04 Marks)

c. Check if the following is SLR(1) grammar:

$S \rightarrow Aa \mid bAc \mid dc \mid bda$

$A \rightarrow d$ (10 Marks)

4 a. Construct LR(1) items for the following grammar and check if it is CLR(1) grammar:

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \in$

$B \rightarrow \in$ (07 Marks)

b. Construct LALR parsing table for the following grammar:

$S \rightarrow E$

$E \rightarrow (L) / a$

$L \rightarrow EL$ (08 Marks)

c. Write a short note on parser generators. (05 Marks)

PART – B

5 a. Define attribute grammar. Explain synthesized and inherited attributes, with an example. (06 Marks)

b. Write SDD for a simple desk calculator. Construct annotated parse tree for the expression $2 * 5 + 4n$ using SDD constructed for a simple desk calculator. (08 Marks)

c. Explain any two applications of SDT. (06 Marks)

6 a. Explain value number method algorithm for constructing the nodes of a DAG. Construct DAG for the expression:
$((x+y)-((x+y)*(x-y)))+((x+y)*(x-y))$ (08 Marks)

b. Define short circuit code. Give SDD for flow-of-control statements. (06 Marks)

c. Explain unification algorithm, with an example. (06 Marks)

7 a. Explain the heap management, in detail. (10 Marks)

b. Explain the activation trees and activation recorder. (10 Marks)

8 a. Explain the issues involved in the design of code generator. (06 Marks)

b. Construct basic blocks and flow graph for:

```
for i from 1 to 10 do
    for j from 1 to 10 do
        a [i, j ] = 0.0 ;
for i from 1 to 10 do
    a [ i, j ] = 1.0 ;
```

(08 Marks)

c. Explain the algorithm for a simple code generator. (06 Marks)

* * * * *