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

06CS63

Sixth Semester B.E. Degree Examination, May/June 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 a neat diagram, the phases of a compiler. **(10 Marks)**

b. Construct a transition diagram for recognizing unsigned numbers. Sketch the program segment to implement it, showing the first two states and one final state. **(10 Marks)**

2 a. Explain the left recursion and show how it is eliminated. Describe the algorithm used for eliminating the left recursion. **(06 Marks)**

b. Eliminate left recursion from the grammar:

$S \rightarrow aB \mid aC \mid Sd \mid Se$

$B \rightarrow bBc \mid f$

$C \rightarrow g$ **(02 Marks)**

c. Given the grammar

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

i) Make necessary changes to make it suitable for LL(1) parsing.

ii) Construct FIRST and FOLLOW sets

iii) Construct the predictive parsing table

iv) Show the moves made by the predictive parser on the input (a, (a, a)) **(12 Marks)**

3 a. Obtain a set of canonical LR(0) items for the grammar: **(08 Marks)**

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow id$

$R \rightarrow L$

b. Is the grammar in Q3(a) SLR(1)? Give reasons. **(04 Marks)**

c. What is handle pruning? Explain with the help of the grammar $S \rightarrow SS + |SS*|a$ and input string aaa*a++. Give a bottom-up parse of the given input string. **(08 Marks)**

4 a. Given the grammar :

$S \rightarrow AA$

$A \rightarrow Aa \mid b$

i) Construct sets of LR(1) items

ii) Construct canonical LR(1) parsing table. **(12 Marks)**

b. Write a note on the Parser generator – Yacc. **(04 Marks)**

c. Write the Yacc specification of a simple desk calculator with the following grammar for arithmetic expressions,

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid digit$

Where, the token digit is a single digit between 0 and 9. **(04 Marks)**

PART – B

5 a. Explain the concept of syntax-directed definition. **(04 Marks)**

b. Consider the context-free grammar given below: **(08 Marks)**

$S \rightarrow EN$

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid digit$

$N \rightarrow ;$

i) Obtain the SDD for the above grammar.

ii) Construct the parse tree, syntax tree and annotated parse tree for the input string 5*6 + 7;

c. Obtain the post-fix SDT for the grammar in Q.5(b) and illustrate the corresponding parser stack implementation. **(08 Marks)**

6 a. Obtain the directed acyclic graph for the expression a + a * (b – c) + (b – c) * d. Also give the sequence of steps for constructing the same. **(06 Marks)**

b. Translate the arithmetic expression a + –(b + c) into quadruples, triples and indirect triples. **(06 Marks)**

c. Explain the syntax-directed translation of switch-statements. **(08 Marks)**

7 a. Describe the general structure of an activation record. Explain the purpose of each item in the activation record. **(06 Marks)**

b. Explain in detail, the strategy for reducing fragmentation in heap memory. **(08 Marks)**

c. Explain briefly the performance metrics to be considered while designing a garbage collector. **(06 Marks)**

8 a. Discuss the issues in the design of a code generator. **(10 Marks)**

b. What are basic blocks and how do you partition a three-address-code into basic blocks? **(05 Marks)**

c. Write the three-address code and construct the basic blocks for the following program segment.

```
Sum = 0;
for (i = 0 ; i < = 10 ; i++)
  Sum = sum + a [i]
```

(05 Marks)

* * * * *

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;
```

c. With an example, explain the use and coordination between 'LEX' and 'YACC' the compiler writing tools. **(04 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)**

* * * * *