18-4-16

INDIAN INSTITUTE OF ENGINEERING, SCIENCE AND TECHNOLOGY, SHIBPUR

B.E. 6th Semester(CST) Final Examination, April 2016

Compiler Design (CS 603)

Time: 3 hours **Full Marks: 70**

(Answer Question no - 1 and any five from the rest)

1. (a) How to handle the *reserved* words and *identifiers* during recognition of tokens?

   (b) Why is it necessary to generate intermediate code instead of generating target program directly?

   (c) Explain the tasks of caller and callee when procedure is called and exit.

   (d) How do register allocation and evaluation order play an important role in a target code generation?

   (e) Explain the following with examples : (i) Quadruples (ii) Triples.

   [2 × 5 = 10]

2. Consider the following Grammar

$$\begin{aligned}&E \rightarrow 5 + T|3 - T\\&T \rightarrow V|V * V|V + V\\&V \rightarrow a|b\end{aligned}$$

where E is the start symbol, set of non-terminals are {E, T, V} and set of terminals are {5, 3, a, b}

   (a) Do the left factoring for the above grammar and compute the *FIRST* and *FOLLOW* set.

   (b) Construct a Predictive parsing table and check if the grammar is LL(1).

   (c) Show the parsing for the sentence, $5 + a * b$.

   [6 + 3 + 3 = 12]

3. Consider the following grammar

$$\begin{aligned}&S \rightarrow CC\\&C \rightarrow cC|d\end{aligned}$$

where S is the start symbol, set of terminals are {c, d}

   (a) Construct the sets of LR(1) items.

   (b) Construct the LALR parsing table for this grammar.

   (c) Show the parsing actions using the parsing table generated in (b) on the input "*cdd*".

   [5 + 5 + 2 = 12]

4. (a) Give the syntax directed definition to process a sample variable declaration in C and construct dependency graph for the input *float x, y, z.*

   (b) List the fields in an activation record. Write down the purpose of each of these fields in an activation record. Explain the sequence of stack allocation process for a function call using a suitable example.

   (c) Describe the method of generating intermediate code for the flow-of-control statements.

[3 + 5 + 4 = 12]

5. (a) Write down the algorithm to find the leader in basic block. Write down the three-address code and construct the basic blocks for the following program segment.

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

   (b) Why symbol-table is needed in various phases of compilers? How hashing can be used to design symbol-table.

   (c) Explain the characteristics of peephole code optimization technique.

[5 + 4 + 3 = 12]

6. (a) Consider the following code fragment and generate the equivalent three address code, compute the basic blocks, control flow graph and eliminate global common subexpression.

```
i= m-1;    j = n;
v=a[n];
while(1)
{
    do i = i+1; while(a[i] < v);
    do j=j-1; while(a[j]>v);
    if( i >= j) break;
    x = a[i]; a[i] = a[j]; a[j] = x;
}
x = a[i]; a[i] = a[n]; a[n] = x;
```

   (b) Explain different loop optimization techniques in detail on the control flow graph generated in (a).

   (c) How is the liveness of a variable calculated?

[6 + 4 + 2 = 12]

7. (a) Explain the role of declaration statements in the intermediate code generation with example.

   (b) Discuss the issues in the designing of a code generator.

   (c) Design a code generator algorithm and explain it with an example.

[3 + 6 + 3 = 12]