Code No: 07A60502 **R07** **Set No. 2**

III B.Tech II Semester Examinations,APRIL 2011
COMPILER DESIGN
Computer Science And Engineering

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. (a) What is the use of Symbol table in compilation process? List out various attributes stored in the symbol table.

   (b) Explain different schemes of storing name attribute in symbol table. [8+8]

2. Explain how redundant sub expression elimination and dead code elimination techniques are applied across different blocks with examples. [16]

3. (a) What is the string generated by the grammar A → (A)A

   (b) Explain the basic method of LL(1) parsing and hence explain how very simple grammar generates strings of balanced parentheses. [8+8]

4. (a) Let A be a 10 * 20 array with low1 = low2 = 1. Therefore n1=10 and n2=20. Take w to be 4. Give the annotated parse tree for the assignment x : = A [y , z].

   (b) Give the semantic rules for declarations in a procedure ? [6+10]

5. Explain the following terms.

   (a) Register descriptor.

   (b) Address descriptor.

   (c) Instruction costs

   (d) Flow graphs. [16]

6. Explain the input buffer scheme for scanning the source program. How the use of sentinels can improve its performance? Describe in detail. [16]

7. Write importance of loop optimization technique. With example explain loop unrolling and frequency reduction . [16]

8. (a) Explain LALR parsing, justify how it is efficient over SLR parsing

   (b) What is phrase level error recovery? [14+2]

⋆ ⋆ ⋆ ⋆ ⋆

Code No: 07A60502 **R07** **Set No. 4**

III B.Tech II Semester Examinations,APRIL 2011
COMPILER DESIGN
Computer Science And Engineering

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. Explain about global data flow analysis? List the data flow equations for reaching definitions for structured programs [16]

2. Explain how storage allocation is done for arrays, strings and records? [16]

3. (a) Construct SLR parsing table for the following grammar.
   S− > AS|b
   A− > SA|a
   (b) What are the actions of shift reduce parse. [10+6]

4. (a) Give a, c are 2 dimensional real arrays and b is a 2 dimensional integer array and i, j are integer variables, write the 3-address code for the following program fragment
   for $(\mathrm{i} = 0, \mathrm{j} = 0; \mathrm{i} < 10\mathrm{AND}\mathrm{j} < 10; \mathrm{i} + +, \mathrm{j} + +)$

   {
   C[i][j]=a[i][j]+b[i][j];
   }

   (b) What is a symbol table? Describe any two methods of implementing a symbol table. [12+4]

5. (a) Consider the following fragment of C code:
   float i, j;
   i = i * 70 + j + 2;
   Write the output at all phases of the compiler for the above 'C' code.
   (b) Write short notes on: input buffering. [8+8]

6. (a) Explain the reasons for separating lexical analysis phase from syntax analysis.
   (b) Eliminate ambiguities from the following grammar.
   S− > iEtSeS|iEtS|a
   E− > b|c|d. [8+8]

7. What is a DAG? Explain role of DAG in optimization with example. [16]

8. What is local and global optimization? Explain with example any three local optimization techniques. [16]

⋆ ⋆ ⋆ ⋆ ⋆

Code No: 07A60502 **R07** **Set No. 1**

III B.Tech II Semester Examinations,APRIL 2011
COMPILER DESIGN
Computer Science And Engineering

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. (a) Explain SDD for Boolean expressions with and without back patching
   (b) Write the SDD for "Do - While" statement and explain? [8+8]

2. (a) Consider the grammar given below.
   $E \rightarrow E + E \,|E - E|\, E^*E|E/E|a|b$
   obtain left most and right most derivation for the string $a+b\ ^*\ a+b$.
   (b) Explain back tracking with example. [8+8]

3. (a) Write about the issues in the design of code generator.
   (b) Write about target code forms. Explain how the instruction forms effect the computation time. [8+8]

4. (a) What is local and global optimization?
   (b) Consider the following part of code.

```
int main()
{
int n,k=0;
scanf("%d",&n);
for(i=2; i<n;i++)
{
if( (n % i) == 0) break;
}
k=1;
if( i==n)
printf("number is prime");
else
printf("number is not prime");
}
```

   i. Identify the basic blocks in the given program.
   ii. Draw the domination tree for the program [6+5+5]

5. Explain shift-reduce parsing with stack implementation. [16]

6. List the various data structures that can be used to organize a symbol table? Compare the performance. [16]

7. Explain the formulation of data flow equations for reaching definition in structured programs. Describe the procedure to compute in and out values. [16]

8. Explain with one example how LEX program perform lexical analysis for the following patterns in 'C': identifier, comments, numerical constants, arithmetic operators. [16]

⋆⋆⋆⋆⋆

Code No: 07A60502 **R07** **Set No. 3**

III B.Tech II Semester Examinations,APRIL 2011
COMPILER DESIGN
Computer Science And Engineering

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. Explain with example the various techniques in loop optimization. [16]

2. (a) What is yaac?
   (b) Explain error recovery in yaac?
   (c) Explain yacc grammar rules? [2+10+4]

3. (a) Construct predictive parse table for the following grammar.
   $E \rightarrow E + T|T$
   $T \rightarrow T\_F|F$
   $F \rightarrow F\_|a|b$
   (b) What are the limitations of recursive descent parser. [8+8]

4. (a) Describe the steps involved for creating a lexical analyzer with Lex.
   (b) Explain the boot strapping process. What is the advantage of using this process? [8+8]

5. (a) Differentiate between L-attributed and S-attributed grammars.
   (b) Define S-attributed and L-attributed definition. [10+6]

6. (a) Explain the importance of each attribute stored in symbol table?
   (b) Compare the performance of different symbol table organization. [10+6]

7. Write the iterative algorithm for reaching definition. Compute in and out for the following figure 1. [16]

8. Generate the code for the following C statements using its equivalent three address code.
   (a) a = b + c
   (b) x = a/(b+c) d*(e+f)
   (c) *A=p
   (d) A=B+C. [16]

⋆ ⋆ ⋆ ⋆ ⋆

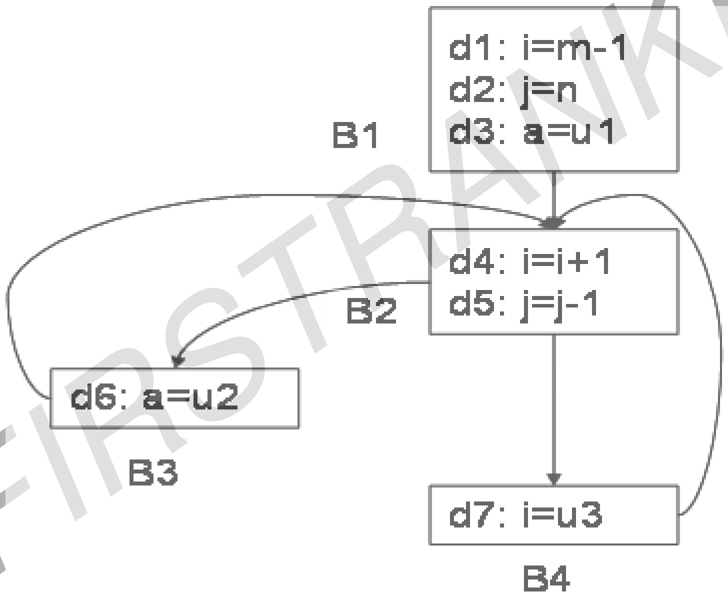

Figure 1: