NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI
Department Of Computer Science And Engineering

Final Assessment

Compiler Design

Marks: 80 Course Code: CSPC62 Time: 3hrs

Instructions to the Students: Answer all questions.

1) a. Describe the stages of a language processing system with the help of a flow diagram. **[5]**

b. Define tokens, patterns and lexemes. Show the difference with the help of an example. **[3+2]**

c. What is a regular definition in compiler design? Give an example of a regular definition for representing unsigned integers or floating point numbers. Draw a transition diagram for this regular definition. **[2+3+3]**

2) a. Consider the following grammar:

$$\begin{array}{lcl}
stmt & \rightarrow & \textbf{if}\ e\ \textbf{then}\ stmt\ stmtTail \\
 & | & \textbf{while}\ e\ \textbf{do}\ stmt \\
 & | & \textbf{begin}\ list\ \textbf{end} \\
 & | & s \\
stmtTail & \rightarrow & \textbf{else}\ stmt \\
 & | & \epsilon \\
list & \rightarrow & stmt\ listTail \\
listTail & \rightarrow & ;\ list \\
 & \rightarrow & \epsilon
\end{array}$$

Take e and s to be the terminals corresponding to "conditional expressions" and "other statements" respectively. (i) Resolve the conflict regarding the expansion of the optional "else" (*stmtTail*) by preferring to consume an else from the input whenever you see one. (ii) Then build a predictive parser for this grammar. (iii)Show error-correction of the idea of synchronizing symbols. Show the behaviour of the parser for the following input:

if e then s; if e then s end **[5+3+3]**

b. Consider the context-free grammar: **S -> S S + | S S * | a** and the string **aa + a*** .
(i) Generate a parse tree for the string. (ii) Design a shift-reduce parser and show the parsing table for the string. (iii) Is this grammar SLR(1) or LALR(1)? Justify. **[2+5+2]**

3) a. Consider the following statement **if (x<100 && x!=y || y>200)**

i) Write syntax-directed definition for implementing the three-address code for the control-flow. b) Generate the code. **[4+4]**

b. Consider the expression: **-((a+b)*(c-d)) + (a+b+c)**

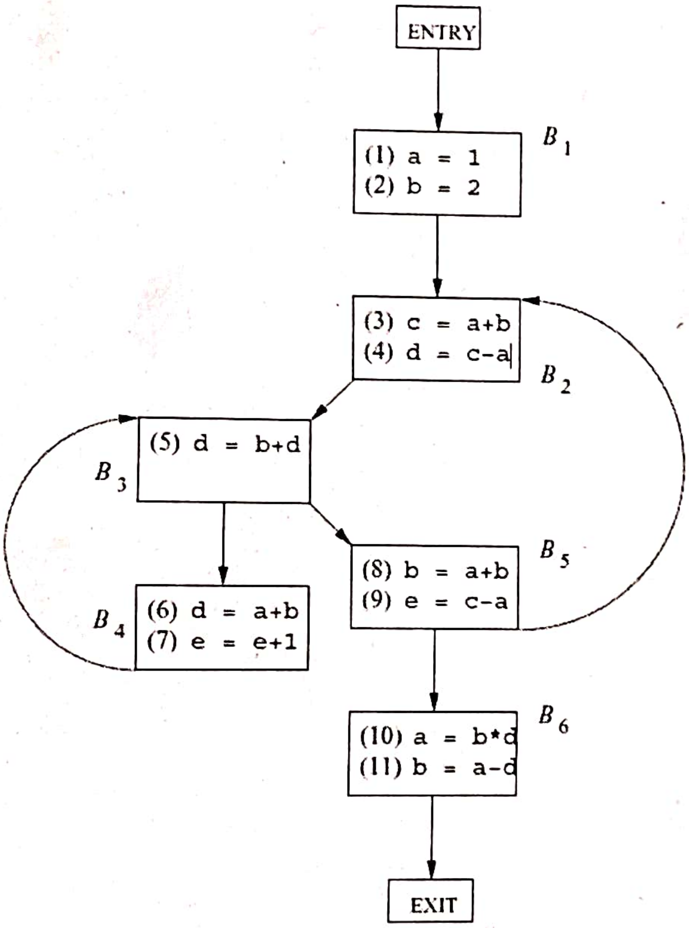
(i) Write quadruples for it. (ii) Draw a DAG for the basic block(s). (iii) Apply optimization on the basic block(s) and find the optimized block. **[3+3+3]**

4) a. Explain address and register descriptors. [3]

b. What is the role of ***getReg*** in code generation. [3]

c. Assume a simple code generator that uses four operations LD, ST, ADD, SUB, MUL. Now translate the basic block generated in the above question **3b** into the actual code. Show the status of the address and register descriptors. Explain the role of the ***getReg*** in this translation. Here, **a,b,c,d** are variables live on exit. [6]

5) Consider the flow-graph. Compute the live variables for each block. Show steps. [5]



```
int f(int n) {
    int t, s;
    if (n < 2) return 1;
    s = f(n-1);
    t = f(n-2);
    return s+t;
}
```

6) a. Discuss the different strategies of run-time storage allocation with pros and cons of each. [4]

b. Draw the activation tree of the function **f()** with initial call **f(5).** [4]