28/11/17

INDIAN INSTITUTE OF ENGINEERING, SCIENCE AND TECHNOLOGY, SHIBPUR
B.Tech 7^{th} Semester Examinations, 2017
Under 5-year Dual-Degree (B. Tech-M. Tech) Programme
Compiler Design (CS 701)

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

(Answer question no - 1 and any five from the remaining)

1. (a) What is intermediate code? Why is it necessary for Compiler Design?
   (b) What are the characteristics of peephole optimization?
   (c) What are the rules to determine the leaders of basic blocks?
   (d) What is register descriptor and address descriptor?
   (e) List the advantages and disadvantages of an operator precedence parser.

   $[2+2+2+2+2=10]$

2. Consider the following grammar G for Boolean expressions:

$$B \rightarrow B \ or \ T \mid T$$
$$T \rightarrow T \ and \ F \mid F$$
$$F \rightarrow not \ B \mid (B) \mid true \mid false$$

   where B is the start symbol, set of non-terminals are $\{B, T, F\}$ and set of terminals are $\{$*and, or, not, (,), true, false* $\}$.

   (a) Compute $FIRST$ and $FOLLOW$ for each nonterminal in G.
   (b) Construct a predictive parsing table for G.
   (c) Show how your predictive parser processes the input string:

   true and not false or true

   Draw the parse tree traced out by your parser.

   $[6+3+3=12]$

3. (a) Describe the relationship between a production and an item in an LR(0) grammar.
   (b) Consider the following grammar and construct the set of LR(0) items. Then fill out an SLR parse table for this grammar and indicate whether the grammar is ambiguous.

$$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$$
$$A \rightarrow d$$
$$B \rightarrow d$$

   Here S is the start symbol.
   (c) Explain *Phrase Level Error Recovery* strategy in LALR parsing method using suitable example.

   $[2+6+4=12]$

4. (a) Give the *Syntax Directed Definition* to process sample statements like declarations in a procedure in C.
   (b) Explain how the types and relative addresses of declared names are computed and how scope information is dealt with.

(c) How backpatching can be used to generate intermediate code for Boolean expressions and flow of control statements.

$[3 + 6 + 3 = 12]$

5. (a) Translate the expression $-(a + b) * (c + d) + (a + b + c)$ into: (i) quadruples, (ii) triples and (iii) indirect triples.

(b) List the fields in an activation record. Write down the purpose of each of these fields in an activation record.

(c) Explain the sequence of stack allocation process for a function call using a suitable example.

$[3 + 5 + 4 = 12]$

6. (a) What are the main operations for a symbol table? Discuss the data structures associated with a symbol table maintained as a list of hash tables and how the operations of a symbol table are implemented in that case. Give an example of what your symbol table would look like for a sample program.

(b) How would you type check a for statement from C++?

$[9 + 3 = 12]$

7. (a) Consider the following sequence of three-address code:

$$
\begin{array}{ll}
 & x = 0 \\
 & i = 0 \\
L: & t_1 = i * 4 \\
 & t_2 = a[t_1] \\
 & t_3 = i * 4 \\
 & t_4 = b[t_3] \\
 & t_5 = t_2 * t_4 \\
 & x = x + t_5 \\
 & i = i + 1 \\
 & if\ i < n\ goto\ L
\end{array}
$$

i. Draw a flow graph for this three-address code.

ii. Optimize this code by eliminating common subexpressions, performing reduction in strength on induction variables, and eliminating all the induction variables that you can. State what transformations you are using at each optimization step.

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

$[(3 + 6) + 3 = 12]$