INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Date: __-___-12 FN/AN Time: 2 Hrs Full Marks: 60 Deptt: Computer Sc. & Engg.
No. of Students: 111 Mid Autumn Semester Examination, 2012-13
Subject No: CS31003 Subject Name: Compiler Design
3^{rd} Year B.Tech. (H) Instruction: Attempt *all* questions

---

1. Consider the following grammar G:

   E → E + T | T
   T → id | id[] | id[X]
   X → E,E | E

   a. Eliminate left recursion in G to construct G_1 with $L(G_1)=L(G)$. [2+
   b. Perform left factoring for G_1 to construct G_2 with $L(G_2)=L(G_1)$. 4+
   c. Compute the FIRST sets for all non-terminals in G_2. 4+
   d. Compute the FOLLOW sets for all non-terminals in G_2. 6+
   e. Build an LL(1) parser for the grammar G_2. 8+
   f. Parse the string **id+id[id+id,id[]]**. Show the stack, the input, and the action taken at every stage of parsing. 10+
   g. Build the parse tree while you are parsing. Show your parse tree. 6= 40]

2. Consider the following grammar G:

   S → a A | c A b | c d | a d b
   A → d

   a. Construct canonical collection of LR(0) items and show that G is not an LR(0) grammar. [4+
   b. Compute FOLLOW sets of the non-terminals and show that G is not an SLR(1) grammar. (You need not construct the complete SLR parser table. Just highlight all the state/s with conflict and justify). 2+
   c. Construct canonical collection of LR(1) items and justify that G is an LALR(1) grammar. 5+
   d. Construct the LALR(1) parser table for G. 3
   e. Using the LALR(1) parser table, parse the following strings: (2+2)+
      i. cdb
      ii. ad
   f. From the parsing of the strings above, justify why G is LALR(1) while it is not SLR(1). 2= 20]