

B.Tech Third Year Mid-Semester Examination
Department: Computer Science and Engineering
Course Name: Compilers **Code: CS 346**
Full Marks-60 **Time: 2 hours**

Answer ALL the questions

Make reasonable assumption as whenever necessary. The notations carry usual meanings. You can answer the questions in any sequence. However, the answers to all the components of any particular question should appear together.

1. Is the following grammar $S \rightarrow X, X \rightarrow Yb \mid aa, Y \rightarrow a \mid bYa$ LALR? Construct all the LALR collections and draw the parsing table to justify your answer. **20**

2. Consider the CFG consisting of the following single production: $S \rightarrow S (S) S \mid \epsilon$

Describe, in words, what language this CFG recognizes. Show that this CFG is ambiguous by demonstrating a single string that can be derived from the CFG and giving two parse trees for the string. Write a non-ambiguous CFG that recognizes the same language. **10**

3. Consider the following grammar: $E \rightarrow E + T, E \rightarrow T, T \rightarrow T * F, T \rightarrow F, F \rightarrow (E), F \rightarrow id$

Assume that SLR/canonical LR parser is used for bottom-up parsing. Construct the parsing table. Show how to parse the string: $(id + id) * id$? For each step, show what is on the stack, what is left in the input buffer, and what action is taken. **20**

4. Consider the following grammar for simple LISP expressions: $List \rightarrow (Sequence), Sequence \rightarrow Sequence Cell, Sequence \rightarrow \epsilon, Cell \rightarrow List, Cell \rightarrow Atom, Atom \rightarrow a$

where 'a', (, and) are terminal symbols, all other symbols are nonterminal symbols, with 'List' being the start symbol.

The above grammar is not LL(1)-Prove or disprove your claim by proper explanation (i.e. constructing parsing table etc.). **10**