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 the usual meanings. You can answer the question in any sequence. However the answers of all the components of any particular question should appear together.

1. Consider the following grammar for Boolean expressions:

bexpr → *bexpr* **or** *bterm* | *bterm*

bterm → *bterm* **and** *bfactor* | *bfactor*

bfactor → **not** *bfactor* | (*bexpr*) | **true** | **false**

(a) Eliminate left recursion from the grammar.

(b) Computer FIRST and FOLLOW for all nonterminals of the resulting grammar.

(c) Using the rules, construct a predictive parsing table for the grammar. **(20)**

2. Consider the CFG as follows: S → R, R → RR, R → R | R, R → R*, R → ε, R → a, R → (R) [*Note that | is a terminal symbol in the grammar*].

LR(1) is a much stronger parsing algorithm than SLR(1). Would using an LR(1) parser instead of the SLR(1) parser resolve the ambiguities? Why or why not? Prove by generating all the canonical LR(1) item sets, and constructing the corresponding parsing table(s). **(20)**

3. (a). Draw a deterministic finite automata that accepts the set of all strings over **(a + b)*** that contain either **aba** or **bb** (or both) as substrings. **(20)**

(b). Consider the following grammar fragment:

stmt → **if** *expr* **then** *stmt*

| **if** *expr* **then** *stmt* **else** *stmt*

| **other**

where **other** stands for other statements in the language.

Show that the grammar is ambiguous.

(79)