CS-362: Lab Problems-4

1) Design a suitable data structure for storing Context Free Grammars. Use this data structure in the following problems.

2) Consider the following grammar:

   G[S]: S → *a*S*b*S | *b*S*a*S |

   Show that the grammar is ambiguous by constructing two different leftmost derivations for *abab*. Write a recursive descent parser with backtracking for this grammar.

3) Consider the following grammar for logical expressions:

   G[E]: E → E **or** T | T
   T → T **and** F | F
   F → **not** F | (E) | **true** | **false**

   Here **or**, **and**, and **not** are logical operators; and **true** and **false** are logical constants. Make this grammar suitable for top-down parsing by removing left recursion from this grammar. Then write a recursive descent parser with backtracking for this grammar.

4) The grammar

   G[S]: S → *a*S*a* | *aa*

   generates all even length strings of *a*'s except for the empty string. Construct a recursive descent parser with backtracking for the grammar that tries the alternate *a*S*a* before *aa*. Show parsing of various strings of a's and determine whether the parser succeeds on 2, 4, 6, and 8 *a*'s.

5) Try constructing a recursive descent parser with exhaustive backtracking for the grammar of problem 5 and show parsing of above strings. Determine if it is able to accept string of 6 *a*'s.