Compiler Construction

A programming language has commands $C$ and expressions $E$ which may involve the terminals $I$ (identifiers) and $N$ (integer constants). Its grammar $G$ (with start symbol $S$) is given by

\[
S ::= C \text{ eof} \\
C ::= I = E | \text{ if } E \text{ then } C | \text{ if } E \text{ then } C \text{ else } C \\
E ::= I | N | E + I
\]

Construct

(a) a recursive descent parser, and [8 marks]

(b) the characteristic finite state machine (CFSM) of a LR($k$), SLR($k$) or LALR($k$) parser, [12 marks]

\textit{explaining carefully} whether there are any problems in the grammar $G$ (and if so how you resolved them) and in adapting the grammar for recursive descent parsing and in adopting the CFSM so as to be suitable for LR($k$), SLR($k$) or LALR($k$) parsing. It is not necessary to consider how the corresponding parse tree is constructed.