2003 Paper 11 Question 2

Compiler Construction

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

$$
\begin{aligned}
S &::= C \ \texttt{eof} \\
C &::= I = E \mid \texttt{if}\ E\ \texttt{then}\ C \mid \texttt{if}\ E\ \texttt{then}\ C\ \texttt{else}\ C \\
E &::= I \mid N \mid E + I
\end{aligned}
$$

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]

explaining carefully whether there are any problems in the grammar $\mathcal{G}$ (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.