2002 Paper 4 Question 2

Compiler Construction

The specification for a pocket-calculator-style programming language is as follows:

- Valid inputs consist either of an Expression followed by the $\boxed{\texttt{enter}}$ button or of an Expression followed by $\boxed{\texttt{store}}$ Identifier $\boxed{\texttt{enter}}$;

- Expressions consist of Numbers and Identifiers connected with the binary operators $\boxed{+}$, $\boxed{\times}$ and $\boxed{\uparrow}$ (in increasing binding power), with the unary operators $\boxed{-}$ and $\boxed{\texttt{abs}}$, and possibly grouped with parentheses. Unary operators bind more strongly than $\boxed{+}$ but weaker than $\boxed{\times}$ so that $-a + b$ means $(-a) + b$ but $-a \times b$ means $-(a \times b)$;

- Numbers consist of a sequence of at least one digit, possibly interspersed with exactly one decimal point, and possibly followed by an exponential marker "`e`" followed by a signed integer, e.g. `6.023e+22`. Identifiers are sequences of lower-case letters.

(*a*) Give a Context-Free Grammar (Type 2 in the Chomsky Hierarchy) for the set of valid input sequences using names beginning with an upper-case letter for non-terminals. It should be complete in that you should go as far as to define e.g.

```
Letter ::= a | b | ... | z
```

[10 marks]

(*b*) Indicate, giving brief reasoning, which non-terminals are appropriate to be processed using lexical analysis and which using syntax analysis proper. [5 marks]

(*c*) Give yacc or CUP input describing those elements deemed in part (*b*) to be suitable for syntax analysis. You need not give "semantic actions". [5 marks]