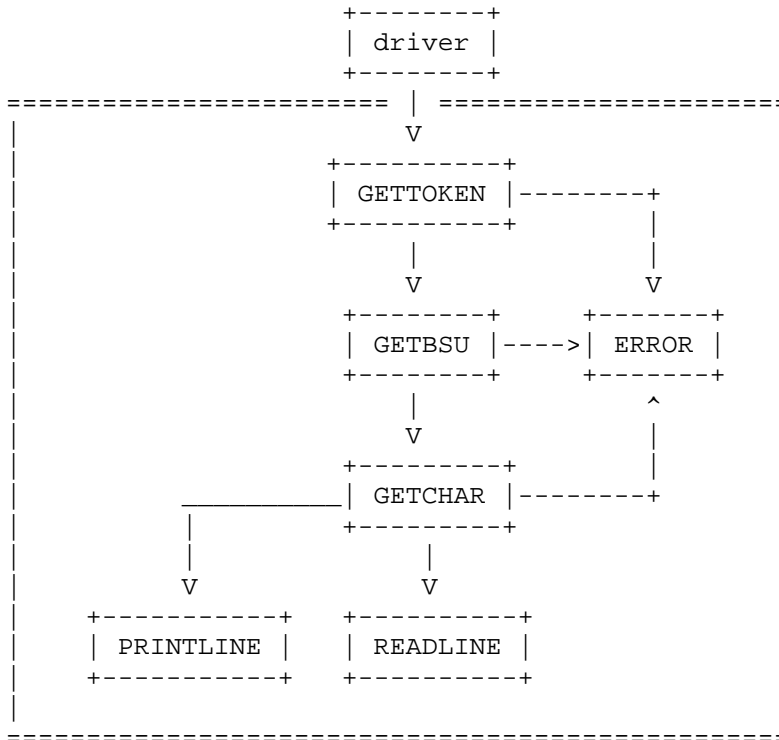


Lexical Analyzer

Using the FSM discussed in class devise a finite state automaton which will recognize the next valid PLS token from the input each time it is called. Label transitions with char1,char2,...,charn/action. Use lambda for "other" characters. Construct an corresponding Lexical Analyzer (scanner) that recognizes all PLS tokens.

Implement the automaton as a transition matrix with a procedure named GETBSU. Add another procedure, GETTOKEN, which calls GETBSU and then looks up and classifies language keywords (e.g., "and", "program", "if", etc.). Except for alphabetic tokens, the classification can be done directly by GETBSU, so GETTOKEN does not need to consider them.

It is suggested that GETTOKEN have 3 parameters: an array of characters to hold the token itself, a variable to hold the number of characters in the token, and a variable to hold the classification of the token. GETTOKEN should be called from a driver routine as a test:



The data you will use to test your lexical analyzer is in the CS4304 ftp directory. Use all files that start with "LA".