unknow → (any character not belonging to the language)

unknow
... → return < UNKNOW, unknow character >

ignored
ignored → (ascii control characters | space)

ignored
... → other → restart the process

letter, underscore or digit
letter
... → ID → other → return <ID, *(lexeme in table entry) >

return < BAD_NUM_1, lexeme >   return < BAD_NUM_3, lexeme >

dot (.)
...
other   other

digit   digit   digit   digit
... digit → NUM → dot (.) → NUM_1 → digit → NUM_2 → E or e → NUM_3 → + or - → NUM_4 → digit → NUM_5 → other →
E or e
return < NUM, *(lexeme in table entry) <

other   other   other

digit   return < BAD_NUM_4, lexeme >

letter (except E or e)
letter

START

letter (except E or e)   return < BAD_NUM_K, lexeme >

end-of-file
... → return < ENDFILE, "ENDFILE" >

Special Symbols Transition Diagrams

<
... → LT → = → return < LTEQ, - >
other → return < LT, - >

>
... → GT → = → return < GTEQ, - >
other → return < GT, - >

+
... → return < PLUS, - >

linefeed
EOF → return < ENDFILE, "ENDFILE" >   EOF → return < ERROR, "missing */ >
-
-
... → MINUS → WEIRD COMM_   COMM_2   COMM_3_   other
other → START   linefeed
linefeed   other
other → return < MINUS, - >   return < ERROR, lexeme >
*
*
COMM   COMM_1   / → START
*
other

=
... → DIV → = → return < NEQ, - >
other → return < DIV, - >

*
... → return < MULT, - >

=
... → return < EQ, - >

:
ASSIGN → = → return< ASSIGN, - >
other → return < UNKNOWN, lexeme >

;
... → return < SEMI, - >

,
→ return < COMMA, - >

(
→ return < LPAREN, - >

)
→ return < RPAREN, - >

[
→ return < LSQR, - >

{
→ return < LCRLY, - >

]
→ return < RSQR, - >

}
→ return <RCRLY, - >