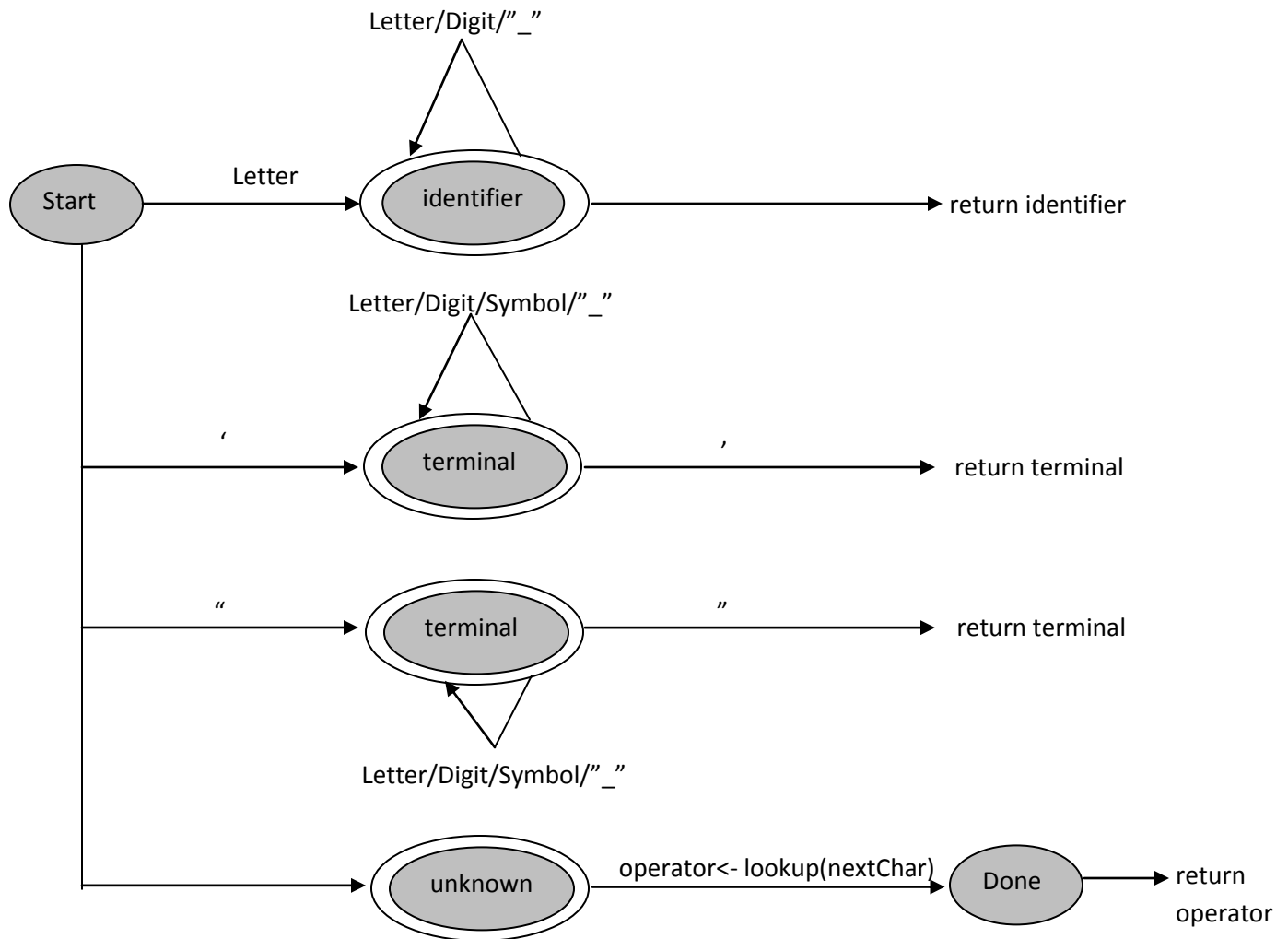


Transition State Diagram



Identifiers can start only with a letter followed by any number of letters, digits, and underscores.

Terminals can start with either a single or a double quote followed by any number of letters, digits, symbols and underscores and they must end with the same type of quote that they start with.

If a token does not start with a letter or quotes, then *lookup* function must determine if this is an operator and if it is, what its type is i.e. it can be one of the following: `[]{}() ; , = |`

Grammar

```
letter = "A" | "B" | "C" | "D" | "E" | "F" | "G"
        | "H" | "I" | "J" | "K" | "L" | "M" | "N"
        | "O" | "P" | "Q" | "R" | "S" | "T" | "U"
        | "V" | "W" | "X" | "Y" | "Z" | "a" | "b"
        | "c" | "d" | "e" | "f" | "g" | "h" | "i"
        | "j" | "k" | "l" | "m" | "n" | "o" | "p"
        | "q" | "r" | "s" | "t" | "u" | "v" | "w"
        | "x" | "y" | "z" ;

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;

symbol = "[" | "]" | "{" | "}" | "(" | ")" | "<" | ">"
        | "'" | '"' | "=" | "|" | "." | "," | ";" ;

character = letter | digit | symbol | "_" ;

identifier = letter , { letter | digit | "_" } ;

terminal = "'" , character , { character } , "'"
          | '"' , character , { character } , '"' ;

lhs = identifier ;

rhs' = rhs , | , rhs
      | rhs , , , rhs ;

rhs = identifier
      | terminal
      | [ , rhs , ]
      | { , rhs , }
      | ( , rhs , )
      | rhs' ;

rule = lhs , = , rhs , ; ;

grammar = { rule } ;
```

The grammar was modified to remove double quotes from EBNF operators for easier parsing and the rule for rhs was rewritten to remove direct left recursion.