`The Grammar Executing Machine

```
$es \leftarrow newstack();\ mode \leftarrow$ SEARCH;                 `Initialization
$ps \leftarrow newstack();\ p0 \leftarrow ps;\ *ps \leftarrow$ " = G'E';";
$+*ps;\ p \leftarrow grammar;\ +p;\ **ps \leftarrow *p;\ p \leftarrow *ps;\ ++p;\ *p \leftarrow$ EoS;
$p \leftarrow grammar;\ i \leftarrow input;\ o \leftarrow$ "";             `‘;’ must begin *grammar*
do $mode =$ PARSE $\Rightarrow$                               `Executing a Rule.
   if $*p \in$ Letter $\Rightarrow$                             `recursive call
      $mode \leftarrow$ SEARCH;
      $+ps;\ *ps \leftarrow p;\ p \leftarrow grammar$               ` *descent*
   ▯ $*p =$ ''' $\Rightarrow$                                 `input
      $+p$;                                          `skip ‘'’
      if $*p = *i \ \Rightarrow\ +i;\ ++p$                          ` *shift*
      ▯ $true \Rightarrow mode \leftarrow$ BACK; $--p$                  `no match
      fi
   ▯ $*p =$ '"' $\Rightarrow$                                 `output
      $+p;\ +o;\ *o \leftarrow *p;\ ++p$
   ▯ $*p =$ ';' $\Rightarrow$                                 `return
      $-p;\ +es;\ *es \leftarrow p$;                             `record parse
      if $ps < p0 \ \Rightarrow mode \leftarrow$ QUIT;                  `stack empty: done
      ▯ $true \Rightarrow p \leftarrow *ps;\ -ps;\ +p$                  ` *ascent*
      fi
   fi
▯ $mode =$ BACK $\Rightarrow$                                `Backtracking.
   if $*p \in$ Letter $\Rightarrow\ +ps;\ *ps \leftarrow p;\ p \leftarrow *es;\ -es$
   ▯ $*p =$ ''' $\Rightarrow\ -i;\ ---p$                          `unshift input terminal
   ▯ $*p =$ '"' $\Rightarrow\ -o;\ ---p$                          `unshift output terminal
   ▯ $*p =$ '=' $\Rightarrow\ mode \leftarrow$ SEARCH; $+p$
   fi
▯ $mode =$ SEARCH $\Rightarrow$                              `Searching for a Rule.
   if $*p \in$ Letter $\Rightarrow +p$                          `skip nonterminal
   ▯ $*p =$ ''' $\Rightarrow +++p$                             `skip input terminal
   ▯ $*p =$ '"' $\Rightarrow +++p$                             `skip output terminal
   ▯ $*p =$ ';' $\Rightarrow$                                 `end of rule
      $+p$;                                          `skip ‘;’
      if $*p = **ps \ \Rightarrow\ mode \leftarrow$ PARSE                `match, *descent*
      ▯ $true \Rightarrow ++p$                              `no match
      fi
   ▯ $*p =$ EoS $\Rightarrow$                                 `search failed
      if $ps \neq p0 \ \Rightarrow$                               `continue or abort
      fi;
      $p \leftarrow *ps;\ -ps;\ mode \leftarrow$ BACK; $-p$               `backtrack
   fi
od
```