Gramática

P → LDE
LDE → LDE DE
| DE
DE → DF
| DT
T → ‘integer’
| ‘char’
| ‘boolean’
| ‘string’
| IDU
DT → ‘type’ IDD ‘=’ ‘array’ ‘[‘ NUM ‘]’ ‘of’ T
| ‘type’ IDD ‘=’ ‘struct’ ‘{‘ DC ‘}’
| ‘type’ IDD ‘=’ T
DC → DC ‘;’ LI ‘:’ ‘T’
| LI ‘:’ ‘T’
DF → ‘function’ IDD ‘(‘ LP ‘)’ ‘:’ T B
LP → LP ‘,’ IDD ‘:’ T
| IDD ‘:’ T
B → ‘{‘ LDV LS ‘}’
LDV → LDV DV
| DV
LS → LS S
| S
DV → ‘var’ LI ‘:’ T ‘;’
LI → LI ‘,’ IDD
| IDD
S → ‘if’ ‘(‘ E ‘)’ S
| ‘if’ ‘(‘ E ‘)’ S ‘else’ S
| ‘while’ ‘(‘ E ‘)’ S
| ‘do’ S while ‘(‘ E ‘)’ ‘;’
| B
| LV = E ‘;’
| ‘break’ ‘;’
| ‘continue’ ‘;’
E → E ‘&&’ L
| E ‘||’ L
| L
L → L ‘<’ R
| L ‘>’ R
| L ‘<=’ R
| L ‘>=’ R
| L ‘==’ R
| L ‘!=’ R
| R
R → R ‘+’ Y
| R ‘-‘ Y
| Y
Y → Y ‘*’ F
| Y ‘/’ F
| F
F → LV
| ‘++’ LV
| ‘--’ LV
| LV ‘++’
| LV ‘--‘
| ‘(‘ E ‘)’
| IDU ‘(‘ LE ‘)’
| ‘-‘ F
| ‘!’ F
| TRUE
| FALSE
| CHR
| STR
| NUM
LE → LE ‘,’ E
| E
LV → LV ‘.’ IDU
| LV ‘[‘ E ‘]’
| IDU
IDD → Id
IDU → Id
TRUE → ‘true’
FALSE → ‘false’
CHR → c
STR → s
NUM → n

Letra = ‘a’ ... ‘z’ + ‘A’...’Z’
Digito = ‘0’...’9’
Id = letra.(letra+digito)*
n = digito * digito
c = ‘’’ . any . ‘’’
s = ‘””’ . any * . ‘””’
onde any representa qualquer caractere lido do arquivo