

S → **C** **F** 'INIT' 'DP' **B**
 # Inicializa as variáveis em **C**, declara função em **F**, programa começa em INIT
C → **D** 'PV' | **D** 'PV' **C**
 # inicialização de variáveis define cada variável de tipos **D** iguais separados por ','
D → **T** **V**
 # Definição começa por um tipo **T** e as variáveis **V**
T → 'INT' | 'REL' | 'LET' | 'TER' | 'BOL' | 'VET' 'ALIST' **T** 'FLIST'
 # Tipos são inteiros real, char, string, booleano e vetor (note que é possível ter vetor de vet)
V → **V** 'VIR' **V** | 'VAR' | 'VAR' 'IGUAL' **E**
 # podemos definir várias variáveis de mesmo tipo separados por vírgula, podem ser inicializadas com expressões
E → **E** 'OPLOG' **E**
 # operadores lógicos E '^' e OU '|' tem as menores precedências
E → 'OPN' **E**
 # operador de negação lógica '!'
E → **E** 'OPCOMP' **E**
 # Operadores de comparação '>', '>=', '<', '<='
E → **E** 'OPA' **E**
 # operadores aritméticos binários '+', '-'
E → **E** 'OPM' **E**
 # operadores multiplicativos '*' e '/'
E → 'OPA' **E**
 # operadores aritméticos unários tem maior precedência de OPM
E → 'AEXP' **E** 'FEXP'
 # expressões dentro de parênteses tem operadores com maior precedência
E → 'VAR' 'AEXP' **X** 'FEXP'
 # retorno de uma função
E → 'VAR' 'ALIST' 'CONSTI' 'FLIST' | 'VAR' 'ALIST' 'VAR' 'FLIST'
 # item de um vetor
E → 'VAR' | 'CONSTI' | 'CONSTR' | 'CONSTL' | 'CONSTP' | 'CONSTB' | **L**
 # constantes e variáveis são sempre resolvidas primeiro
L → 'ALIST' **X** 'FLIST'
X → **E** | **E** 'VIR' **X** | ϵ
 # uma lista constante é uma sequência de expressões separadas por ',' entre '[' e ']'
F → 'FUNC' 'AEXP' **A** 'FEXP' 'ABLOCO' **B** 'FBLOCO'
F → 'FUNC' 'AEXP' **A** 'FEXP' 'ABLOCO' **B** 'RET' **E** 'PV' 'FBLOCO'
 # funções podem ter retorno ou não
A → ϵ | **R** | **R** 'VIR' **R**
R → **T** 'VAR' | **T** 'VAR' 'IGUAL' **E**
 # argumentos precisam ter seus tipos especificados
B → **B** 'PV' | **B** 'PV' **B** | 'VAR' 'IGUAL' **E** | **E**
 # blocos podem ser atribuições ou expressões (já que retorno de função é expressão)
B → 'SE' **E** 'ABLOCO' **B** 'FBLOCO' | 'SE' **E** 'ABLOCO' **B** 'FBLOCO' 'SEN' 'ABLOCO' **B** 'FBLOCO'
 # define desvio
B → 'ENQ' **E** 'ABLOCO' **B** 'FBLOCO'
 # Define iteração com controle lógico
B → 'PARA' 'VAR' 'IGUAL' **E** 'VIR' **E** 'ABLOCO' **B** 'FBLOCO'
B → 'PARA' 'VAR' 'IGUAL' **E** 'VIR' **E** 'VIR' **E** 'ABLOCO' **B** 'FBLOCO'
 # define iteração com controle por controlador
B → 'TEC' **Z** | 'MON' **Z**
Z → 'VAR' | **E** | **Z** 'VIR' **Z**
 # define entrada e saída