

**UNIVERSIDADE FEDERAL DE ALAGOAS  
INSTITUTO DE COMPUTAÇÃO  
CIÊNCIA DA COMPUTAÇÃO**



**UNIVERSIDADE FEDERAL  
DE ALAGOAS**

**Gramática - BFS**

**Bruna Leal Torres Silva  
Eirene de Oliveira Fireman  
Samuel Lucas Vieira Lins Barbosa**

**Maceió - AL  
20 de Dezembro de 2021**

# Sumário

<b>Sumário</b>	<b>1</b>
<b>Gramática livre de Contexto</b>	<b>2</b>

# 1. Gramática livre de Contexto

**S** = Function S | 'function' 'main' '(' ')' '{' FunctionBody '}'

**Function** = 'function' FTipo '\_id' '(' ParF ')' '{' FunctionBody '}'

**FTipo** = Tipo | 'void'

**Tipo** = 'int' | 'float' | 'char' | 'string' | 'bool'

**ParF** = LParF |  $\epsilon$

**LParF** = DecPar | LParF ',' Tipo PId

**DecPar** = Tipo '\_id' | 'array' Tipo '\_id' '[' Ea ']'

**PId** = 'ID' | 'ID' '[' ']' | 'ID' '[' Ea ']'

**FunctionBody** = DecVar FunctionBody |

**DecVar** = Tipo '\_id' InId | 'array' Tipo '\_id' '[' Ea ']' InArr

**InId** = '=' Ec |  $\epsilon$

**InArr** = '=' '[' LEc ']' |  $\epsilon$

**LEc** = Ec | LEc ',' Ec

===== (ANTIGO) =====

**S** = Function S | DeclD S |  $\epsilon$

**Function** = 'function' FunctionType FunctionName '(' FunctionParam ')' '{' IFunc '}' Function |  $\epsilon$

**FunctionType** = IDType | 'void'

**FunctionName** = 'ID' | PR\_MAIN

**FunctionParam** = IDType ID | IDType ID ',' MPFParam IDType ID  
FunctionParam |  $\epsilon$

**MPFParam** = IDType ID ',' |  $\epsilon$

**IDType** = 'int' | 'float' | 'char' | 'string' | 'bool'

**ArrayDeclaration** = 'array' DcID '[' Number ']' | 'array' DcID '[' Number ']'  
ArrayAtribuition

**ArrayAtribuition** = '[' Ea | Ea ',' MPArray Ea ']'

**MPArray** = Ea ',' |  $\epsilon$

**AritOperation** = (ID | FunCall | Number) AritSymbols AritOperation | (ID | FunCall |  
Number)

**IFunc** = DcID IFunc ';' | FunCall IFunc ';' | Loop IFunc | While IFunc | VarAtribuition  
IFunc ';' | ArrayDeclaration IFunc ';' | ArrayAtribuition IFunc ';' | Coments IFunc | SysIn  
IFunc ';' | SysOut IFunc ';' | StringConcat IFunc ';' | Return IFunc ';' | Condicionais  
IFunc IFunc ';' |  $\epsilon$

**StringConcat** = ID '&' CT\_SRING | ID '&' CT\_CHAR

**Condicionais** = 'if' '(' Eb ')' '{ Instrucao '}' | 'elif' '(' Eb ')' '{ Instrucao '}' | 'else' '{  
Instrucao '}'

**FuncCall** = FunctionName '(' FunctionParam ')'

**DcID** = IDType ID Atribuition | IDType ID ',' MPDcID ID |  $\epsilon$

**MPDcID** = ID ',' |  $\epsilon$

**Atribuition** = '=' Ec |  $\epsilon$

**AritSymbols** = '+' | '-' | '\*' | '/' | '%'

**VarAtribuition** = ID Atribuition;

**SysParam** = ID SysParam | ID ',' ID SysParam |  $\epsilon$

**SysIn** = 'SysIn' '(' SysParam ')'

**SysOut** = 'SysOut' '(' SysParam ')'

**Return** = 'return' ID

**Loop** = 'for' '(' ID '=' Ea ',' Ea ')' '{ Instrucao }'

**While** = 'while' '(' Eb ')' '{ Instrucao }'

**Ec** = Ec 'OPR\_CONC' Eb Eb

**Eb** = Eb 'PR\_OR' Tb | Tb

**Tb** = Tb 'PR\_AND' Fb | Fb

**Fb** = Fb 'OPR\_REL' Ra | 'OPR\_NOT' Fb | Ra

**Ra** = Ra 'OPR\_REL' Ea | Ea

**Ea** = Ea 'OPR\_ADD' Ta | Ea 'OPR\_SUB' Ta | Ta

**Ta** = Ta 'OPR\_MULT' Fa | Ta 'OPR\_DIV' Fa | Fa

**Fa** = '(' Ec ')' | 'OPR\_SUB' Fa | IdOuFunCham | 'CT\_INT' | 'CT\_FLOAT' | 'CT\_BOOL'  
| 'CT\_STRING' | 'CT\_CHAR'

**OPR\_REL** = 'OPR\_DIGUAL' | 'OPR\_DIF' | 'OPR\_MAIOR' | 'OPR\_MENOR' |  
'OP\_MAIORIG' | 'OP\_MENORIG'

## 2. Gramática LL(1)

**S** = 'function' DcFunction S

**DcFunction** = 'main' FunctionHeader | Type ID FunctionHeader

**Type** = void | float | char | string | bool

**FunctionHeader** = '(' Param ')' '{' Instructions '}'

**Param** = DcParam | DcParamArray |  $\epsilon$

**DcParam** = Type ID MultDcParam

**MultDcParam** = ',' DcParam | ',' DcParamArray |  $\epsilon$

**DcParamArray** = 'array' Type ID '[' ']' MultDcParam

**Instructions** = {DcVar|DcArr|Command|InOut|Return|ID AttrId} Instructions |  $\epsilon$

**DcVar** = Type DcVarAtr ';'

**DcVarAtr** = ID Atr DcVarAtrFat

**Atr** = '=' Ec MultAtr | '[' Ea ']' Atr |  $\epsilon$

**MultAtr** = ',' DcVarAtr | '[' Ea ']' MultAtr |  $\epsilon$

**DcArr** = 'array' Type ID '[' CT\_INT | ID ']' DcArrAtr ';'

**DcArrAtr** = '=' '{' ArrAtr '}' | epsilon

**ArrAtr** = CT | ID MultArrAtr

**MultArrAtr** = ',' ArrAtr | epsilon

**Command** = if | while | for

**Condicional** = '(' Eb ')' '{' Instructions '}' Elif Else

**If** = 'if' Condicional

**Elif** = 'elif' Condicional | epsilon

**Else** = 'else' '{' Instructions '}' | epsilon

**While** = 'while' '(' Ec ')' '{' Instructions '}'

**For** = 'for' '('('Start Stop Increment')' '{' Instructions '}'

**Start** = ID '=' Ec

**Stop** = ',' Ec

**Increment** = ',' CT\_INT | epsilon

**InOut** = SysIn | SysOut

**SysIn** = 'SysIn' '(' ParamIn ')' ';'

**ParamIn** = ID MultiParamIn | '[' Ea ']' MultiParamIn

**MultiParamIn** = ',' ParamIn | epsilon

**SysOut** = 'SysOut' '(' ParamOut ')' ';'

**ParamOut** = '\" CT\_STRING '\" MultiParamOut | ID MultiParamOut | '[' Ea ']'

MultiParamOut

**MultiParamOut** = '+' ParamOut | epsilon

**Return** = 'return' ParamReturn

**ParamReturn** = Ec ';' | ';'

**AttrId** = Attr | FunctionCall

**FunctionCall** = '(' ParamFunctionCall

**ParamFunctionCall** = ')' ';' | ParamFunction ')' ';'

**ParamFunction** = Ec MultiParamFunction

**MultiParamFunction** = ',' ParamFunction | epsilon

**IdFunCall** = FunctionCall | '[' Ea ']' IdFunCall

**Ec** = Eb EcLL

**EcLL** = 'OPR\_CONC' Eb EcLL |  $\epsilon$

**Eb** = Tb EbLL

**EbLL** = 'OPR\_OR' Tb EbLL | 'OPR\_AND' Tb EbLL |  $\epsilon$

**Tb** = Ra TbLL

**TbLL** = 'OPR\_NOT' Ra TbLL |  $\epsilon$

**Ra** = Rb RaLL

**RaLL** = **Equality** Rb RaLL |  $\epsilon$

**Rb** = Ea RbLL

**RbLL** = **Comparison** Ea RbLL |  $\epsilon$

**Ea** = Ta EaLL

**EaLL** = 'OPR\_ADD' Ta EaLL | 'OPR\_SUB' Ta EaLL |  $\epsilon$

**Ta** = Fa TaLL

**TaLL** = 'OPR\_MULT' Fa TaLL | 'OPR\_DIV' Fa TaLL | 'OPR\_MOD' Fa TaLL |  $\epsilon$

**Fa** = '(' Ec ')' | 'id' **IdFunCall** | 'CT\_INT' | 'CT\_FLOAT' | 'PR\_TRUE' | 'PR\_FALSE' |

'PR\_BOOL' | 'CT\_CHAR' | 'CT\_STR'

**Equality** = 'OPR\_DIGUAL' | 'OPR\_DIF'

**Comparison** = 'OPR\_MAIOR' | 'OPR\_MAIORIG' | 'OPR\_MENOR' |

'OPR\_MENORIG'