UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
CIÊNCIA DA COMPUTAÇÃO

Gramática - BFS

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

Maceió - AL
20 de Dezembro de 2021

Sumário

1. Gramática livre de Contexto

S = Function S | DecID S | ε

Function = ‘function’ FunctionType FunctionName ‘(‘ FunctionParam ‘)’ ‘{‘ IFunc ‘}’ Function | ε

FunctionType = IDType | ‘void’

FunctionName = ‘ID’ | PR_MAIN

FunctionParam = IDType ID | IDType ID ‘,’ MPFParam IDType ID | ε

MPFParam = IDType ID ‘,’ | ε

IDType = ‘int’ | ‘float’ | ‘char’ | ‘string’ | ‘bool’

ID = ‘_’A
A = {a, …, z}| {A, …, Z} + B
B = {a, …, z}B | {A, …, Z}B | {0, …, 9}B | B | ε

IFunc = DcID IFunc ‘;’ | FuncCall IFunc ‘;’ | Loop IFunc | While IFunc | VarAtribuition IFunc;’ | ArrayDeclaration IFunc ‘;’ | ArrayAtribuition IFunc ‘;’ | Coments IFunc | SysIn IFunc ‘;’ | SysOut IFunc ‘;’ | StringConcat IFunc ‘;’ | Return IFunc ‘;’ | Condicionais IFunc| IFunc ‘;’ | ε

DcID = IDType ID Atribuition | IDType ID ‘,’ MPDcID ID | ε

MPDcID = ID ‘,’ | ε

StringConcat = ID ‘&’ CT_SRING | ID ‘&’ CT_CHAR

Condicionais = ‘if’ ‘(‘ Eb ‘)’ ‘{‘ Instrucao ‘}’ | ‘elif’ ‘(‘ Eb ‘)’ ‘{‘ Instrucao ‘}’ | ‘else’ ‘{‘ Instrucao ‘}’

FuncCall = FunctionName ‘(‘ FunctionParam ‘)’

Atribuition = ‘ = ‘ ID | ‘ = ‘ FuncCall | ‘ = ‘ SysIn ‘| ‘ = ‘ AritOperation | ‘ = ‘ ‘’’ Char ‘’’ | ‘ = ‘ ““‘ String ““‘ | ‘ = ‘ Number | ‘ = ‘ StringConcat | ε

ArrayDeclaration = ‘array’ DcID ‘[‘ Number ‘]’ | ‘array’ DcID ‘[‘ Number ‘]’ ArrayAtribuition

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

AritSymbols = ‘+’ | ‘-’ | ‘*’ | ‘/’ | ‘%’

VarAtribuition = ID Atribuition;

String = {a…z} String | {A…Z} String | {0…9} String | ‘[:punct:] {-} [\’]’ String | ε

Char = {a…z}| {A…Z} | ε

SysParam = ID SysParam | ID ‘,’ ID SysParam | ε

SysIn = ‘SysIn’ ‘(‘ SysParam ‘)’

SysOut = ‘SysOut’ ‘(‘ SysParam ‘)’

ArrayAtribuition = ‘[‘ Ea | Ea ‘,’ MPArray Ea ‘]’

MPArray = Ea ‘,’ | ε

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 S | DecID S | ε

Function = ‘function’ FunctionType FunctionName ‘(‘ FunctionParam ‘)’ ‘{‘ IFunc ‘}’ Function | ε

FunctionType = IDType | ‘void’

FunctionName = ‘ID’ | PR_MAIN

FunctionParam = IDType ID | IDType ID ‘,’ MPFParam IDType ID | ε

MPFParam = IDType ID ‘,’ | ε

IDType = ‘int’ | ‘float’ | ‘char’ | ‘string’ | ‘bool’

ID = ‘_’A
A = {a, …, z}| {A, …, Z} + B
B = {a, …, z}B | {A, …, Z}B | {0, …, 9}B | B | ε

IFunc = DcID IFunc | FuncCall IFunc | Loop IFunc | While IFunc | VarAtribuition IFunc| ArrayDeclaration IFunc | ArrayAtribuition IFunc| Coments IFunc | SysIn IFunc | SysOut IFunc | StringConcat IFunc | Return IFunc | Condicionais I ‘;’ IFunc | ε

DcID = IDType ID Atribuition | IDType ID ‘,’ MPDcID ID | ε

MPDcID = ID ‘,’ | ε

StringConcat = ID ‘&’ CT_SRING | ID ‘&’ CT_CHAR

Condicionais = ‘if’ ‘(‘ Eb ‘)’ ‘{‘ Instrucao ‘}’ | ‘elif’ ‘(‘ Eb ‘)’ ‘{‘ Instrucao ‘}’ | ‘else’ ‘{‘ Instrucao ‘}’

FuncCall = FunctionName ‘(‘ ID FuncCall | ‘,’ ID FuncCall | Es FuncCall | ‘,’ Es FuncCall | FuncCall ‘)’ | ε

Atribuition = ‘ = ‘ ID | ‘ = ‘ FuncCall | ‘ = ‘ SysIn ‘| ‘ = ‘ AritOperation | ‘ = ‘ ‘’’ Char ‘’’ | ‘ = ‘ ‘“‘ String ‘“‘ | ‘ = ‘ Number ‘ = ‘ StringConcat | ε

ArrayDeclaration = ‘array’ DcID ‘[‘ Number ‘]’ | ‘array’ DcID ‘[‘ Number ‘]’ ArrayAtribuition

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

AritSymbols = ‘+’ | ‘-’ | ‘*’ | ‘/’ | ‘%’

VarAtribuition = ID Atribuition

String = {a…z} String | {A…Z} String | {0…9} String | ‘[:punct:] {-} [\’]’ String | ε

Char = {a…z}| {A…Z} | ε

SysParam = ID SysParam | ID ‘,’ ID SysParam | ε

SysIn = ‘SysIn’ ‘(‘ SysParam ‘)’

SysOut = ‘SysOut’ ‘(‘ SysParam ‘)’

ArrayAtribuition = ‘[‘ Ea | Ea ‘,’ MPArray Ea ‘]’

MPArray = Ea ‘,’ | ε

Return = ‘return’ ID

Loop = ‘for’ ‘(‘ ID ‘=’ Ea ‘,’ Ea ‘)’ ‘{‘ Instrucao‘}’

While = ‘while’ ‘(‘ Eb ‘)’ ‘{‘ Instrucao ‘}’

Ec = Eb EcLL

EcLL = ‘OPR_CONC’ Eb EcLL | ε

Eb = Tb EbLL

EbLL = ‘PR_OR’ Tb EbLL | ε

Tb = Fb TbLL

TbLL = ‘PR_AND’ Fb TbLL | ε

Fb = ‘OPR_NOT’ Fb | Ra FbLL

FbLL = ‘OPR_MAIOR’ Ra FbLL | FbLL = ‘OPR_MENOR’ Ra FbLL | ‘OPR_MAIORIG’ Ra FbLL | ‘OPR_MENORIG’ Ra FbLL | ε

Ra = Ea RaLL

RaLL = ‘OPR_REL’ Ea RaLL | ε

Ea = Ta EaLL

EaLL = ‘OPR_ADD’ Ta EaLL | ‘OPR_SUB’ Ta EaLL | ε

Ta = Pa TaLL

TaLL = ‘OPR_MULT’ Pa TaLL | ‘OPR_DIV’ Pa TaLL | ε

Pa = Fa PaLL

PaLL = ‘OPR_MOD’ Fa PaLL | ε

Fa = ‘(‘ Ec ‘)’ | ‘OPR_SUB’ Fa | IdOuFunCham | ‘CT_INT’ | ‘CT_FLT’ | ‘CT_BOOL’ | ‘CT_STRING’ | ‘CT_CHAR’

OPR_REL = ‘OPR_DIGUAL’ | ‘OPR_DIF’ | ‘OPR_MAIOR’ | ‘OPR_MENOR’ | ‘OP_MAIORIG’ | ‘OP_MENORIG’