Universidade Federal de Alagoas
Instituto de Computação
Bacharelado em Ciência da Computação

Gramática - Isengard++

Márcio Henrique Vieira de Oliveira
Michael Miller Rodrigues Cardoso

Maceió - AL, 2021

Sumário

1. Gramática livre de contexto

S = DcFun S | DecId S | ε

DecFun = ‘Funct’ FunType NameFunId ‘(‘ ConstDc ‘)’ InternDc

NameFun = ‘id’ | ‘PR_MAIN’

Param = VarType ‘id’ Param | ‘,’ VarType ‘id’ Param | Es Param | ‘,’ Es Param | ε

AbFecPar = ‘(’ Param ‘)’

FunType = ‘Int’ | ‘Float’ | ‘Str’ | ‘Bool’ | ‘Void’ | ‘Char’ | ‘Main’

VarType = ‘Int’ | ‘Float’ | ‘Char’ | ‘Str’ | ‘Bool’

InternDc = ‘Begin’ Instruction ‘End’

DcId = VarType ‘id’ ‘;’ | ‘id’ | ε

ParamFun = ParamFun ‘,’ Ec | Ec | ε

ConstDc = ConstDc ‘,’ VarType Id | VarType Id | ε

Id = ‘id’ ‘[‘ Ea ‘]’ | ‘id’

IdLL = LId ‘,’ Id ‘=’ Ec | LId ‘,’ Id | Id ‘=’ Ec | Id

VetType = ‘[’ Es ‘]’ | ε

Atr = Atr ‘,’ ‘id’ ‘=’ Ec ‘;’ | Atr ‘,’ ‘id’ ‘[’ Ea ‘]’ ‘=’ Ec ‘;’ | ‘id’ ‘=’ Ec | ‘id’ ‘[‘ Ea ‘]’ ‘=’ Ec | ε

Instructions = Command Instruction | DcId Instructions | Id ‘(‘ ParamFun ‘)’ ‘;’ Instructions | ‘Return’ Return ‘;’ | ε

Command = ‘If’ ‘(’ Eb ‘)’ InternDc | If ‘(’ Eb ‘)’ InternDc ‘Else’ InternDc

Command = ‘While’ ‘(’ Eb ‘)’ | ‘For’ ‘(’ ‘PR_INT’ ‘=’ ‘id’ ‘,’ ‘id’ ‘,’ ‘id’ ‘)’ |
‘Input’ ‘(’ Id ‘)’ | ‘Output’ ‘(’ Id ‘)’

Command = FunCall

FunCall = ‘id’ AbFecPar ‘;’ | ‘id’ ‘(‘ ParamFun ‘)’ ‘;’

IdFunCall = Id | ‘id’ ‘(‘ ParamFun ‘)’ | ‘id’

If = ‘If’ ‘(’ Eb ‘)’ InternDc’ | ‘If’ ‘(’ Eb ‘)’ InternDc ‘Else’

Else = ‘Else’ ‘Begin’ Instruction ‘End’ | ε

InputParam = Id | InputParam

OutputParam = ‘,’ Eb OutputParam | ε

While = ‘While’ ‘(’ Eb ‘)’ ‘Begin’ Instruction ‘End’

For = ‘For’ ‘(‘ Int ‘id’ ‘=’ Ea ‘,’ Ea ‘)’ ‘Begin’ Instruction ‘End’

Return = ‘Return’ Ec ‘;’

Input = ‘Input’ ‘(’ ‘id’ ‘)’ ‘;’

Output = ‘Output’ ‘(’ Es ‘)’ ‘;’

Ec = Ec ‘OP_CONCAT’ Eb Eb

Eb = Eb ‘PR_OR’ Tb | Tb

Tb = Tb ‘PR_AND’ Fb | Fb

Fb = Fb ‘OP_REL’ Ra | ‘OP_NOT’ Fb | Ra

Ra = Ra ‘OP_REL’ Ea | Ea

Ea = Ea ‘OP_AD’ Ta | Ea ‘OP_SUB’ Ta | Ta

Ta = Ta ‘OP_MULT’ Fa | Ta ‘OP_DIV’ Fa | Fa

Fa = ‘(‘ Ec ‘)’ | ‘OP_SUB’ Fa | IdOrFunCall | ‘CT_INT’ | ‘CT_FLOAT’ | ‘CT_BOOL’ | ‘CT_CHAR’ | ‘CT_STR’

OP_REL = ‘==’ | ‘!=’ | ‘OP_GREATER’ | ‘OP_LESS’ | ‘OP_GREATERT’ | ‘OP_LESST’

2. Gramática LL(1)

S = DeclFunction S | DeclId S | &

DcFun = ‘Funct’ FunTipo NameFunId ‘(‘ ConstDc ‘)’ InternDc

NameFun = ‘id’ | ‘PR_MAIN’

Param = VarType ‘id’ Param | ‘,’ VarType ‘id’ Param | Es Param | ‘,’ Es Param | &

AbFecPar = ‘(’ Param ‘)’

Param = VarType ’id’ Param | ’,’ VarType ’id’ Param | Es Param | ’,’ Es Param | &

FunType = ‘Int’ | ‘Float’ | ‘Str’ | ‘Bool’ | ‘Void’ | ‘Char’ | ‘Main’

VarType = ‘Int’ | ‘Float’ | ‘Char’ | ‘Str’ | ‘Bool’

InternDc = ‘FunInternDc’ ‘Begin’ LDc ‘End’

DcId = VarType ‘id’ ‘;’ | ‘id’ | &

ParamFun = Ec ParamFunLL | &

ParamFunLL = ‘,’ Ec ParamFunLL| &

ConstDc = VarType ‘id’ VetType ConstDc_LL | &

ConstDc_LL = ‘,’ VarType ‘id’ VetType ConstDc_LL | &

Id = ‘id’ VetType

IdLL = Id Atr Id_LL

Id_LL = ‘,’ Id Atr IdLL | &

VetType = ‘[’ Ea ‘]’ | ε

Atr = ‘,’ Id ‘=’ Ec ‘;’ Atr | &

Instructions = Command Instruction | DcId Instructions | Id ‘(‘ ParamFun ‘)’ ‘;’ Instructions | ‘Return’ Return ‘;’ | &

Command = ‘If’ ‘(’ Eb ‘)’ InternDc | If ‘(’ Eb ‘)’ InternDc ‘Else’ InternDc

Command = ‘While’ ‘(’ Eb ‘)’ | ‘For’ ‘(’ ‘PR_INT’ ‘=’ ‘id’ ‘,’ ‘id’ ‘,’ ‘id’ ‘)’ | ‘Input’ ‘(’ Id ‘)’ | ‘Output’ ‘(’ Id ‘)’

Command = FunCall

FunCall = ‘id’ AbFecPar ‘;’

If = ‘If’ ‘(’ Eb ‘)’ InternDc | ‘If’ ‘(’ Eb ‘)’ InternDc ‘Else’ InternDc

Else = ‘Else’ InternDc | &

While = ‘While’ ‘(’ Eb ‘)’ ‘Begin’ Instruction ‘End’

For = ‘For’ ‘(‘ Int ‘id’ ‘=’ Ea ‘,’ Ea ‘)’ ‘Begin’ Instruction ‘End’

Return = ‘Return’ Es ‘;’

InputParam = ‘id’ VetType InputParamLL

InputParamLL = ‘,’ ‘id’ VetType InputParamLL | &

OutputParam = ‘,’ Ec OutputParam | &

Ec = Eb EcLL

EcLL = ‘OP_CONCAT’ Eb EcLL | &

Eb = Tb EbLL

EbLL = ‘PR_OR’ Tb EbLL | &

Tb = Fb TbLL

TbLL = ‘PR_AND’ Fb TbLL | &

Fb = ‘OP_NOT’ Fb | Ra FbLL

FbLL = ‘OP_GREATER’ Ra FbLL | FbLL = ‘OP_LESS’ Ra FbLL | ‘OP_GREATERT’ Ra FbLL | ‘OP_LESST’ Ra FbLL | &

Ra = Ea RaLL

RaLL = ‘OP_REL’ Ea RaLL | &

Ea = Ta EaLL

EaLL = ‘OP_AD’ Ta EaLL | ‘OP_SUB’ Ta EaLL | &

Ta = Pa TaLL

TaLL = ‘OP_MULT’ Pa TaLL | ‘OP_DIV’ Pa TaLL | &

Pa = Fa PaLL

PaLL = ‘OP_RES’ Fa PaLL | &

Fa = ‘(‘ Ec ‘)’ | ‘OP_SUB’ Fa | IdOrFunCall | ‘CT_INT’ | ‘CT_FLOAT’ | ‘CT_BOOL’ | ‘CT_CHAR’ | ‘CT_STR’

IdFunCall = ‘id’ IdFunCall_LL

IdFunCall_LL = ‘(‘ ParamFun ‘)’ | ‘[‘ Ea ‘]’