UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação
Bacharelado em Ciência da Computação

Gramática Livre de Contexto - Linkin Park

Hellena Almeida Canuto
João Vitor Santos Tavares
Wagner Anthony de Medeiros Silva

Maceió - AL, 17/12/2021

Sumário

1.Gramática Livre de Contexto

S = DcFun S | DcId S | ε
DcFun = ‘Function’ FuncType FuncNameId ‘(’ DcConst ‘)’ DcIntern
FuncName = ‘id’ | ‘RW_MAIN’
Param = VarType ‘id’ Param | ‘,’ VarType ‘id’ Param | Es Param | ‘,’ Es Param | ε
OpClosPar = ‘(‘ Param ‘)’
FunType = ‘Int’ | ‘Float’ | ‘Char’ | ‘STR’ | ‘Bool’ | ‘Empty’ | ‘Main’
VarType = ‘Int’ | ‘Float’ | ‘Char’| ‘STR’ | ‘Bool’
InternDc = ‘Open’ Instructions ‘Close’
DcId = VarType ‘id’ ‘;’ | ‘id’ | ε
FuncParam = FuncParam ‘,’ CE | CE | ε
DcConst = DcConst ‘,’ VarType Id | VarType Id | ε
Id = ‘id’ ‘[‘ AE ‘]’ | ‘id’
IdLL = LId ‘,’ Id ‘=’ CE | LId ‘,’ Id | Id ‘=’ CE | Id
TypeVect = ‘[‘ Es ‘]’ | ε
Attr = Attr ‘,’ ‘id’ ‘=’ CE | ‘;’ | Attr ‘,’ ‘id’ ‘[‘ AE ‘]’ ‘=’ CE ‘;’ | ‘id’ ‘=’ CE | ‘id’ ‘[‘ AE ‘]’ ‘=’ CE |ε
Instructions = Comm Instructions | DcId Instructions | Id ‘(‘ FuncParam ‘)’ ‘;’ Instructions | ‘Back’ return ‘;’ | ε
Comm = ‘If’ ‘(‘ BE ‘)’ DcIntern | ‘If’ ‘(‘ BE ‘)’ DcIntern ‘Else’ DcIntern
Comm = ‘While’ ‘(‘ BE ‘)’ DcIntern | ‘For’ ‘(‘ RW_INT ‘=’ ‘id’ ‘,’ ‘id’ ‘,’ ‘id’ ‘)’ | ‘Scan’ ‘(‘ Id ‘)’ | ‘Print’ ‘(‘ Id ‘)’
Commnd = FuncCall
FuncCall = ‘id’ OpClosPar ‘;’ | ‘id’ ‘(‘ FuncParam ‘)’ ‘;’
IdFunCall = Id | ‘id’ ‘(‘ FuncParam ‘)’ | ‘id’
If = ‘If’ ‘(‘ BE ‘)’ DcIntern | ‘If’ ‘(‘ BE ‘)’ DcIntern ‘Else’
Else = ‘Else’ ‘Open’ Instructions ‘Close’ | ε
PrintParam = ‘,’ BE PrintParam | ε
ScanParam = Id | PrintParam
While = ‘While’ ‘(‘ BE ‘)’ ‘Open’ Instructions ‘Close’
For = ‘For’ ‘(‘ ‘Int ‘id’ ‘=’ AE ‘,’ AE ‘)’ ‘Open’ Instructions ‘Close’
Back = ‘Back’ CE ‘;’
Scan = ‘Scan’ ‘(‘ ‘id’ ‘)’ ‘;’
Print = ‘Print’ ‘(‘ Es ‘)’ ‘;’
CE = CE ‘OP_CONCAT’ BE BE
BE = BE ‘OP_OR’ BT | BT
BT = BT ‘OP_AND’ BF | BF
BF = BF ‘OP_REL’ AR | ‘NOT_OP’ BF | AR
AR = AR ‘OP_REL’ AE | AE
AE = AE ‘OP_SUM’ AT | AE ‘OP_SUB’ AT | AT

AT = AT ‘OP_MUL’ AF | AT ‘OP_DIV’ AF | AF
AF = ‘(‘ CE ‘)’ | ‘OP_SUB’ AF | FunCallOrId | ‘CTE_INT’ | ‘CTE_FLOAT’ | ‘DEL_BOOL’ | ‘CTE_STR’ | ‘CTE_CHAR’
OP_REL = ‘==’ | ‘!=’ | ‘OP_GREATER’ | ‘OP_LESS’ | ‘OP_EQUALG’ | ‘OP_EQUALL’

2. Gramática LL(1)

S = DclFunction S | DclId S | $
DcFunc = ‘Function’ FuncType FuncNameId ‘(‘ DcConst ‘)’ DcIntern
FuncName = ‘id’ | ‘RW_MAIN’
Param = VarType ‘id’ Param | ‘,’ VarType ‘id’ Param | Es Param | ‘,’ Es Param | $
OpClosPar = ‘(‘ Param ‘)’
FuncType = ‘Int’ | ‘Float’ | ‘Char’ | ‘STR’ | ‘Bool’ | ‘Empty’ | ‘main’
VarType = ‘Int’ | ‘Float’ | ‘Char’| ‘STR’ | ‘Bool’
DcIntern = ‘Open’ InternDcFunct LDc ‘Close’
DcId = VarType ‘id’ ‘;’ | ‘id’ | $
FuncParam = CE FuncParamLL | $
FuncParamLL = ‘,’ CE FuncParamLL | $
DcConst = VarType ‘id’ TypeVect DcConst_LL | $
DcConst_LL = ‘,’ VarType ‘id’ TypeVect DcConst_LL | $
Id = ‘id’ TypeVect
IdLL = Id Attr LL_Id
LL_Id = ‘,’ Id Attr IdLL | $
TypeVect = ‘[‘ AE ‘]’ | ε
Attr = ‘,’ Id ‘=’ CE ‘;’ Attr | $
Instructions = Comm Instructions | DcId Instructions | Id ‘(‘ FuncParam ‘)’ ‘;’ Instructions | ‘Back’ Back ‘;’ | $
Comm = ‘If’ ‘(‘ BE ‘)’ DcIntern | If ‘(‘ BE ‘)’ DcIntern ‘Else’ DcIntern
Comm = ‘While’ ‘(‘ BE ‘)’ | ‘For’ ‘(‘ ‘RW_INT’ ‘=’ ‘id’ ‘,’ ‘id’ ‘,’ ‘id’ ‘)’ | ‘Scan’ ‘(‘ Id ‘)’ | ‘Print’ ‘(‘ Id ‘)’
Comm = FuncCall
FuncCall = ‘id’ OpClosePar ‘;’
If = ‘If’ ‘(‘ BE ‘)’ DcIntern | ‘If’ ‘(‘ BE ‘)’ DcIntern ‘Else’ DcIntern
Else = ‘Else’ DcIntern | $
While = ‘While’ ‘(‘ BE ‘)’ ‘Open’ Instructions ‘Close’
For = ‘For’ ‘(‘ Int ‘id’ ‘=’ AE ‘,’ AE ‘)’ ‘Open’ Instructions ‘Close’
Back = ‘Back’ Es ‘;’
ScanParam = ‘id’ TypeVect ScanParam_LL
ScanParam_LL = ‘.’ ‘id’ VarType ScanParam_LL | $
PrintParam = ‘,’ CE PrintParam | $
CE = CE_LL
CE_LL = ‘OP_CONCAT’ BE CE_LL | $

BE = BT BE_LL
BE_LL = ‘OP_OR’ BT BE_LL | $
BT = BF BT_LL
BT_LL = ‘OP_AND’ BF BT_LL | $
BF = ‘NOT_OP’ BF | AR BF_LL
BF_LL = ‘OP_GREATER’ AR BF_LL | BF_LL = ‘OP_LESS’ AR BF_LL | ‘OP_EQUALG’ AR BF_LL | ‘OP_EQUALL’ AR BF_LL | $
AR = AE AR_LL
AR_LL = ‘OP_EQUAL’ AE AR_LL | $
AE = AT AE_LL
AE_LL = ‘OP_SUM’ AT AE_LL | ‘OP_SUB’ AT AE_LL | $
AT = Pa AT_LL
AT_LL = ‘OP_MUL’ Pa AT_LL | ‘OP_DIV’ Pa AT_LL | $
Pa = AF Pa_LL
Pa_LL = ‘OP_MOD’ AF Pa_LL | $
AF = ‘(‘ CE ‘)’ | ‘OP_SUB’ AF | FunCallOrId | ‘CTE_INT’ | ‘CTE_FLOAT’ | ‘DEL_BOOL’ | ‘CTE_STR’ | ‘CTE_CHAR’
FunCallOrId = ‘id’ FunCall_Id_LL
FunCall_Id_LL = ‘(‘ FuncParam ‘)’ | ‘[‘ AE ‘]’