UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação
Ciência da Computação

Especificação da Gramática da linguagem C+-

Compiladores

Daniel Humberto Cavalcante Vassalo
Gabriel Barbosa Pereira

2018.1

Analisador

O analisador sintático escolhido foi o ascendente SLR(1).

Especificação da Gramática SLR(1)

S = Code

```
Code  = DeclVarG Code
      | DeclFunc Code
      | DeclFuncMain

DeclVarG = DeclVar

DeclVar = 'id' RealType 'lineEnd'
      | 'id' RealType 'commaSep' DeclVar
      | 'id' RealType 'opAssign' Expression 'lineEnd'
      | 'id' RealType 'opAssign' Expression 'commaSep' DeclVar

RealType = 'colonSep' TypeAtomic
      | 'colonSep' TypeArray

TypeAtomic= 'tInt'
      | 'tFloat'
      | 'tBool'
      | 'tChar'
      | 'tString'

TypeArray = TypeAtomic 'arrayBeg' Expression 'arrayEnd'

Literal = 'consNumInt'
      | 'consNumFlo'
      | 'consBool'
      | 'consChar'
      | 'consString'
      | LiteralArray

LiteralArray = 'arraybeg' ExpressionList 'arrayEnd'

Expression = ExprLogicOr

ExprLogicOr = ExprLogicOr 'opLogicOr' ExprLogicAnd
    | ExprLogicAnd
ExprLogicAnd = ExprLogicAnd 'opLogicAnd' ExprEquals
    | ExprEquals
ExprEquals = ExprEquals 'opEquals' ExprAditiv
    | ExprEquals 'opRel' ExprAditiv
      | ExprAditiv
ExprAditiv = ExprAditiv 'opPlus' ExprMult
```

```
            | ExprAditiv 'opMinus' ExprMult
            | ExprMult
ExprMult = ExprMult 'opMult' ExprExp
            | ExprExp
ExprExp = ExprExp 'opExp' ExprUnitary
            | ExprParams
ExprParams = 'paramBeg' ExprLogicOr 'paramEnd'
            | ExprUnitary
ExprUnitary = 'opUnMinus' ExprParams
            | 'opNot' ExprParams
            | 'id'
            | 'id' MatrixIndex
            | 'id' FuncParamsReal
            | 'rwLength' 'paramBeg' ExprUnitary 'paramEnd'
            | Literal

ExpressionList = Expression
            | ExpressionList 'commaSep' Expression

MatrixIndex = MatrixIndex 'arrayBeg' Expression 'arrayEnd'
            | 'arrayBeg' Expression 'arrayEnd'

Assign = 'id' 'opAssign' Expression 'lineEnd'
            | 'id' MatrixIndex 'opAssign' Expression 'lineEnd'

LoopWhCom = 'rwWhile' ExprParams Scope

LoopForCom = 'rwFor' 'paramBeg' Assign 'lineEnd' Expression 'paramEnd'
Scope
            | 'rwFor' 'paramBeg' Assign 'lineEnd' Expression 'lineEnd' Expression
'paramEnd' Scope

CondCom = 'rwIf' ExprParams Scope
            | 'rwIf' ExprParams Scope 'rwElif' ExprParams Scope
            | 'rwIf' ExprParams  Scope 'rwElif' ExprParams Scope 'rwElse' Scope
            | 'rwIf' ExprParams Scope 'rwElse' Scope

PrintCom = 'rwPrint' 'paramBeg' 'consString' 'paramEnd' 'lineEnd'
            | 'rwPrint' 'paramBeg' 'consString' 'commaSep' ExpressionList
'paramEnd' 'lineEnd'

ReadCom = 'rwRead' 'paramBeg' 'consString' 'paramEnd' 'lineEnd'
            | 'rwRead' 'paramBeg' 'consString' 'commaSep' ExpressionList
'paramEnd' 'lineEnd'

CallFunc = 'id' FuncParamsReal 'lineEnd'

DeclFuncMain = 'main' 'paramBeg' 'paramEnd' SpecReturnType Scope

DeclFunc = 'id' FuncParamsFormal SpecReturnType Scope
```

```
SpecReturnType = 'colonSep' FormalType
      | SpecReturnTypeVoid

SpecReturnTypeVoid = 'colonSep' 'rwVoid'

FormalType = TypeAtomic
      | TypeArrayUndefined

TypeArrayUndefined = TypeAtomic 'arrayBeg' 'arrayEnd'

FuncParamsFormal = 'paramBeg' 'paramEnd'
      | 'paramBeg' FuncParamsFormalR 'paramEnd'

FuncParamsFormalR = 'id' 'colonSep' FormalType
      | 'id' 'colonSep' FormalType 'commaSep' FuncParamsFormal

FuncParamsReal = 'paramBeg' 'paramEnd'
      | 'paramBeg' ExpressionList 'paramEnd'

Scope = 'scopeBeg' 'scopeEnd'
      | 'scopeBeg' Commands 'scopeEnd'

Commands = DeclVar Commands
      | DeclVar
      | Assign Commands
      | Assign
      | LoopWhCom Commands
      | LoopWhCom
      | LoopForCom Commands
      | LoopForCom
      | CondCom Commands
      | CondCom
      | PrintCom Commands
      | PrintCom
      | ReadCom Commands
      | ReadCom
      | CallFunc Commands
      | CallFunc
      | 'rwReturn' Expression 'lineEnd'
```