UNIVERSIDADE FEDERAL DE ALAGOAS

INSTITUTO DE COMPUTAÇÃO - IC

CIÊNCIA DA COMPUTAÇÃO

JOÃO VICTOR DE ALARCÃO AYALLA ALCÂNTARA

ASCANIO SAVIO DE ARAUJO NEVES

JACKSON BARBOSA DA SILVA

COMPILADORES

ESPECIFICAÇÃO DA LINGUAGEM AJA++

Sumário

1 - Gramática

S = MainFunction | Function S | Declaration S | ε

MainFunction = ´function´ ´void´ ´main´ ´{´ ´}´ ´[´ CommandsBlock ´]´

Function = ´function´ FunctionType ´Identificator´ ´{´ ParametersList ´}´ ´[´ CommandsBlock ´]´

FunctionType = ´void´ | ´itg´ | ´dbl´ | ´chr´ | ´bool´ | 'str´

ParametersList = Declaration ´,´ ParametersList | Declaration

CommandsBlock = Command CommandsBlock | Command

Command = FunctionCall | AttributionExpression | ConcatenationExpression | Declaration | Condition | Loop | Output | Input | AppendList | Return

FunctionCall = ´Identificator´ ´{´ IdList ´}´

IdList = ´Identificator´ ´,´ IdList | ´Identificator´

AttributionExpression = ´Identificator´ ´=´ AhritimeticExpression | VariableDeclaration ´=´ AhritimeticExpression

AhritimeticExpression = AhritimeticExpression BitwiseOperation AddtiveExpression | AddtiveExpression

AddtiveExpression = AddtiveExpression AddtiveOperation MultiplicativeExpression | MultiplicativeExpression

MultiplicativeExpression = MultiplicativeExpression MultiplicativeOperation AhritimeticValue | AhritimeticValue

AhritimeticValue = AhritimeticFactor IncrementOperation | AhritimeticFactor

AhritimeticFactor = Value | FunctionCall

BitwiseOperation = ´OperationXor´ | ´OperationOr´ | ´OperationAnd´

AddtiveOperation = ´OperationAdd´ | ´OperationSub´

MultiplicativeOperation = ´OperationMult´ | ´OperationDiv´

IncrementOperation = ´OperationInc´ | ´OperationDec´

ConcatenationExpression = ConcatenationExpression ´OperationConc´
BooleanExpression | BooleanExpression

Declaration = VariableDeclaration | ListDeclaration
ListDeclaration = ´list´ ´(´ Type ´)´ ´Identificator´

VariableDeclaration = Type ´Identificator´

Type = ´itg´ | ´dbl´ | ´chr´ | ´bool´ | 'str´

Condition = IfCommand ElseCondition | IfCommand
ElseCondition = ElseIfCommand ElseCondition | ElseIfCommand | ElseCommand

IfCommand = ´if´ ´{´ BooleanExpression ´}´ ´[´ CommandsBlock ´]´

ElseIfCommand = ´elseif´ ´{´ BooleanExpression ´}´ ´[´ CommandsBlock ´]´

ElseCommand = ´else´ ´[´ CommandsBlock ´]´

Loop = LogicForStatement | CounterForStatement | WhileStatement

LogicForStatement = ´for´ ´{´ AttributionExpression ´;´ BooleanExpression ´;´
AhritimeticExpression ´}´ ´[´ CommandsBlock ´]´

CounterForStatement = ´for´ ´{´ ´Identificator´ ´,´ IntValue ´,´ IntValue ´,´ IntValue
´}´ ´[´ CommandsBlock ´]´ | ´for´ ´{´ ´Identificator´ ´,´ IntValue ´,´ IntValue ´}´ ´[´
CommandsBlock ´]´

WhileStatement = ´while´ ´{´ BooleanExpression ´}´ ´[´ CommandsBlock ´]´

BooleanExpression = BooleanExpression 'LogicOr' BooleanTerm | BooleanTerm

BooleanTerm = BooleanTerm 'LogicAnd' BooleanFactor | BooleanFactor

BooleanFactor = 'LogicNot' BooleanRelation | BooleanRelation

BooleanRelation = AhritimeticExpression LogicalRelation AhritimeticExpression |
AhritimeticExpression

LogicalRelation = `RelationEqual´ | ´RelationNotEqual´ | ´RelationGreater´ |
´RelationLower´ | ´RelationGreaterEqual´ | ´RelationLowerEqual´

Output = ´write´ ´{´ OutputValues ´}´

OutputValues = Value ´,´ OutputValues | Value

Input = ´read´ ´{´ Identificators ´}´

Identificators = ´Identificator´ ´,´ Identificators | ´Identificator´

AppendList = ´Identificator´ ´.´ ´append´ ´{´ Value ´}´

Return = 'return' Value

Value = ´Identificator´ | ´CharConst´ | ´StringConst´ | ´DoubleConst´ | ´IntConst´ |
´BooleanConst´

IntValue = ´Identificator´ | ´IntConst´