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 DOS TOKENS - AJA++

Sumário

1 - Linguagem da Implementação

A linguagem de programação adotada para a implementação dos analisadores léxico e sintático da linguagem AJA++ foi C++.

2 - Enumeração e categorias dos tokens

Identificator = 0, DefFunction = 1, ReservedMain = 2, TypeVoid = 3, TypeInteger = 4, TypeDouble = 5, TypeChar = 6, TypeBoolean = 7, TypeString = 8, TypeList = 9, OpenBrace = 10, CloseBrace = 11, OpenBrack = 12, CloseBrack = 13, OpenPar = 14, ClosePar = 15, EndLine = 16, ReservedIf = 17, ReservedElseIf = 18, ReservedElse = 19, ReservedFor = 20, ReservedWhile = 21, ReservedWrite = 22, ReservedRead = 23, SignalSemiColon = 24, SignalComma = 25, OperationAdd = 26, OperationSub = 27, OperationMult = 28, OperationDiv = 29, OperationInc = 30, OperationDec = 31, OperationConc = 32, OperationNot = 33, OperationXor = 34, OperationOr = 35, OperationAnd = 36, LogicAnd = 37, LogicOr = 38, LogicNot = 39, AtributionEqual = 40, RelationEqual = 41, RelationNotEqual = 42, RelationGreater = 43, RelationLower = 44, RelationGreaterEqual = 45, RelationLowerEqual = 46, ReservedReturn = 47, CharConst = 48, StringConst = 49, DoubleConst = 50, IntConst = 51, BooleanConst = 52, SignalDot = 53, ReservedAppend = 54;

3 - ERs Auxiliares

Digits = ‘[:digit:]’
Double = ‘[:digit:]’ + .
Alphanumeric = ‘[:alnum:]’
Symbol =‘[:punct:]{-}[\’]’
AlphanumericSymbol = ‘[{Alphanumeric}{Symbol}]’
String = ′\′{AlphanumericSymbol}+[:space:]+[:word:]+\′′
Bool = ‘true|false’
Id = ‘[:upper:] | [:lower:]+[:upper:] | [:lower:] | [:digit:]*’

4 - Tabela

0	Identificator	´{Id}´
1	DefFunction	'function'
2	ReservedMain	'main'
3	TypeVoid	'void'
4	TypeInteger	'itg'
5	TypeDouble	'dbl'
6	TypeChar	'chr'
7	TypeBoolean	'bool'
8	TypeString	'string'
9	TypeList	'list'
10	OpenBrace	'{'
11	CloseBrace	'}'
12	OpenBrack	'['
13	CloseBrack	']'
14	OpenPar	'('
15	ClosePar	')'
16	EndLine	'\n'
17	ReservedIf	'if'
18	ReservedElseIf	'elseif'
19	ReservedElse	'else'
20	ReservedFor	'for'
21	ReservedWhile	'while'
22	ReservedWrite	'write'
23	ReservedRead	'read'
24	SignalSemiColon	';'
25	SignalComma	','
26	OperationAdd	'+'
27	OperationSub	'-'
28	OperationMult	'*'
29	OperationDiv	'/'
30	OperationInc	'++'
31	OperationDec	'--'
32	OperationConc	'+='
33	OperationNot	'!'
34	OperationXor	'^'
35	OperationOr	'|'
36	OperationAnd	'&'
37	LogicAnd	'and'
38	LogicOr	'or'
39	LogicNot	'not'
40	AtributionEqual	'='

41	RelationEqual	'=='
42	RelationNotEqual	'!='
43	RelationGreater	'>'
44	RelationLower	'<'
45	RelationGreaterEqual	'>='
46	RelationLowerEqual	'<='
47	ReservedReturn	'return'
48	CharConst	´{Alphanumeric}´
49	StringConst	´{String}´
50	DoubleConst	´{Double}´
51	IntConst	´{Digits}´
52	BooleanConst	´{Bool}´
53	SignalDot	‘.’
54	ReservedAppend	‘append’

5 - Especificação dos tokens da linguagem

Main:
ReservedMain = 'main'

Tipos Primitivos:
TypeVoid = 'void'
TypeInteger = 'itg'
TypeDouble = 'dbl'
TypeChar = 'chr'
TypeBoolean = 'bool'
TypeString = 'string'
TypeList = 'list'

Função:
DefFunction = 'function'
ReservedReturn = 'return'

Palavras reservadas:

ReservedIf = 'if'
ReservedElseIf = 'elseif'
ReservedElse = 'else'

ReservedFor = 'for'
ReservedWhile = 'while'
ReservedAppend = ‘append’

Operadores Lógicos:

LogicAnd = 'and'
LogicOr = 'or'
LogicNot = 'not'

Operadores Aritméticos:

OperationAdd = '+'
OperationSub = '-'
OperationMult = '*'
OperationDiv = '/'
OperationInc = '++'
OperationDec = '--'
OperationConc = '+='

Operador de atribuição:

AtributionEqual = '='

Instruções de leitura e escrita:

ReservedWrite = 'write'
ReservedRead = 'read'

Símbolos:

OpenBrace = '{'
CloseBrace = '}'
OpenBrack = '['
CloseBrack = ']'
OpenPar = '('
ClosePar = ')'
EndLine = '\n'

Sinais:

```
SignalSemiColon =  ';'
SignalComma  = ','
SignalDot = ‘.’
```