

Universidade Federal de Alagoas

Instituto de Computação

COMPILADORES - 2018.1

## **Especificação dos tokens: Albireo 18**

Aluno: Victor Rafael Almeida Cavalcante

Curso: Ciência da Computação

Professor: Alcino Dall' Igna Junior

Maceió-AL

Jul, 2018

# Sumário

<b>1</b>	<b>Linguagem dos analisadores</b>	<b>1</b>
<b>2</b>	<b>Categoria dos tokens</b>	<b>1</b>
<b>3</b>	<b>Tabelas</b>	<b>2</b>
3.1	Expressões regulares auxiliares . . . . .	2
3.2	Tabela de categorias simbólicas e ER dos lexemas . . . . .	2

# 1 Linguagem dos analisadores

A linguagem de programação utilizada na implementação dos analisadores Léxico e Sintático será a linguagem **Java**.

## 2 Categoria dos tokens

```
public enum TokenCategory {  
    EOF(0) ,  
    ID(1) ,  
    TYPE(2) ,  
    CONSINT(3) ,  
    CONSFLOAT(4) ,  
    CONSBOOL(5) ,  
    CONSCHAR(6) ,  
    CONSSTRING(7) ,  
    COMMASEP(8) ,  
    DOTSEP(9) ,  
    SEMICOLSEP(10) ,  
    PARENSTART(11) ,  
    PARENEND(12) ,  
    BRACKSTART(13) ,  
    BRACKEND(14) ,  
    CURLYSTART(15) ,  
    CURLYEND(16) ,  
    OPLOGICAND(17) ,  
    OPLOGICOR(18) ,  
    OPLOGICNOT(19) ,  
    OPARITADD(20) ,  
    OPARITSUB(21) ,  
    OPARITMULT(22) ,  
    OPARITDIV(23) ,  
    OPARITMOD(24) ,  
    OPNEGUN(25) ,  
    OPEQUAL(26) ,  
    OPNOTEQUAL(27) ,  
    OPGT(28) ,  
    OPLT(29) ,  
    OPGTEQ(30) ,  
    OPLTEQ(31) ,  
    OPATTRB(32) ,  
    OPCONC(33) ,  
    INSIF(34) ,  
    INSDO(35) ,  
    INELSE(36) ,  
    INSWHILE(37) ,  
}
```

```

        INSWCTRL(38),
        SWRETURN(39),
        SWEMPTY(40),
        SWREAD(41),
        SWPRINT(42),
        SWINIT(43),
        SWFUNC(44);
        SWGROUP(45);
    }

```

## 3 Tabelas

### 3.1 Expressões regulares auxiliares

```

id = '[a-zA-Z][a-zA-Z0-9_]\'
nat = '[0-9]|([1-9][0-9]*)\'
integer = '[+|-]*{nat}\'
exponential = '[+\\-]?(?:0|[1-9]\\d*)(?:\\.\\d*)?(?:[eE][+\\-]?\\d+)?\'
symbol = '!\' | '@\' | '#\' | '$\' | '%\' | '&\' | '*'\' | '(' | ')\' | ',' |
    '+' | '-' | '=' | '_' | '[' | ']' | '{' | '}' | '^' | '~' | '`' |
    ''' | ',\' | '\"' | '.' | ';' | ':' | '/' | '?' | '\\\' | '\"' | '|';

```

### 3.2 Tabela de categorias simbólicas e ER dos lexemas

**Identificador:**

EOF = 'EOF'

ID = '{id}'

**Tipos primitivos**

TYPE = 'int' | 'float' | 'char' | 'boolean' | 'charstring';

**Constantes de tipos:**

CONSINT = '{integer} | {exponential}'

CONSFLOAT = '{integer}[.]{integer}'

```
CONSBOOL = 'yes' | 'no';
CONSHAR = '('('id' | 'integer' | 'symbol'))(')';
CONSSTRING = '(")(id' | 'integer' | 'symbol')*(")'
```

#### **Delimitadores:**

```
PARENSTART = '(';
PARENEND = ')';
BRACKSTART = '[';
BRACKEND = ']';
CURLYSTART = '{';
CURLYEND = '}';
COMMASEP = ',';
DOTSEP = '.';
SEMICOLSEP = ';';
```

#### **Operadores Lógicos:**

```
OPLOGICAND = '&&';
OPLOGICOR = '||';
OPLOGICNOT = '!';
```

#### **Operadores Aritméticos:**

```
OPARITADD = '+' ;
OPARITSUB = '-' ;
OPARITMULT = '*' ;
OPARITDIV = '/' ;
OPARITMOD = '%';
```

#### **Operador Unário:**

OPNEGUN = '—';

#### **Operadores Relacionais:**

OPEQUAL = '==';

OPNOTEQUAL = '!=';

OPGT = '>';

OPLT = '<';

OPGTEQ = '>=';

OPLTEQ = '<=';

#### **Operadores de atribuição:**

OPATTRB = '=';

#### **Operador Concatenação:**

OPCONC = '+';

#### **Instruções:**

INSIF = 'if';

INSDO = 'do';

INSELSE = 'else';

INSWHILE = 'while';

INSWCTRL = 'whilecontrol';

#### **Palavras reservadas:**

SWRETURN = 'return';

SWEMPTY = 'empty';

SWREAD = 'readin';

SWPRINT = 'printout';

SWINIT = 'initializeApp';

```
SWFUNC = 'foo';  
SWGROUP = 'group';
```