



UNIVERSIDADE FEDERAL DE ALAGOAS – UFAL

INSTITUTO DE COMPUTAÇÃO

CIÊNCIA DA COMPUTAÇÃO

COMPILADORES

Professor Alcino Dall'Igna Júnior

AUDREY EMMELY RODRIGUES VASCONCELOS

NATÁLIA DE ASSIS SOUSA

WILLIAM PHILIPPE LIRA BAHIA

MACEIÓ, AL 17 DE DEZEMBRO DE 2021

1. EBNF

As alternativas são separadas por barras verticais: ou seja, ' $a \mid b$ ' significa " a ou b ".

Os colchetes indicam opcionalidade: ' $[a]$ ' representa um a opcional, ou seja, " $a \mid \epsilon$ " (ϵ se refere à sequência vazia).

Os colchetes indicam repetição: ' $\{a\}$ ' significa " $\epsilon \mid a \mid aa \mid aaa \mid \dots$ ".

2. Regras Léxicas

```
letter ::= a | b | ... | z | A | B | ... | Z
digit  ::= 0 | 1 | ... | 9
ch ::= 'ch' | '\n' | '\0', onde ch denota qualquer caractere
ASCII imprimível
str ::= "{ch}", onde ch denota qualquer caractere ASCII
imprimível
id ::= letter { letter | digit | _ }
```

3. Produções Gramaticais

```
prog : {func}

func : type id '(' parameters ')' '{' { type var_decl { ',',
var_decl } ';' } { statement } '}' ';'
| undefined id '(' parameters ')' '{' { type var_decl { ',',
var_decl } ';' } { statement } '}' ';'

var_decl : id [ '[' digit ']' ]

type : int
| float
| char
| string
| bool

parameters : undefined
| type id [ '[' ']' ] { ',', type id [ '[' ']' ] }

statement : if '(' expr ')' '{' statement '}' [ elseif {
statement '}' ][ else { statement '}' ] ';'
| while '(' expr ')' '{' statement '}' ';'
| do '{' statement '}' while '(' expr ')' ';'
| for '(' [ assg ] ';' [ expr ] ';' [ assg ] ')' '{'
statement '}' ';'
| return [ expr ] ';'
| assg ';'
| id '(' [expr { ',', expr } ] ')' ';'

1
```

```

| '{' { stmt } '}'
| ';'

assg : id [ '[' expr ']' ] = expr

expr : '!' expr
| '-' expr
| expr aritop expr
| expr relop expr
| expr logical_op expr
| id [ '(' [expr { ',' expr } ] ')' ] | '[' expr ']' ]
| '(' expr ')'
| digit
| ch
| str

aritop : *
| /
| +
| -

relop : ==
| !=
| <=
| <
| >=
| >

logical_op : &
| |

```

4. Associatividades e precedências de operadores

A tabela a seguir fornece as associatividades de vários operadores e suas precedências relativas. As precedências diminuem à medida que descemos na tabela.

Operador	Associatividade
!, - (unário)	direita para esquerda
*, /	esquerda para direita
+, - (binário)	esquerda para direita
<, <=, >, >=	esquerda para direita

==, !=	esquerda para direita
&	esquerda para direita
	esquerda para direita