

Trabalho 1 - Compiladores 2022/1

Milton Pagliusi e Eduardo Pandini

1. Definição da Gramática (Forma de Backus-Naur)

DIGITO	[0-9]
ID	[A-Za-z_][A-Za-z0-9]*
TEXTO	[A-Za-z0-9][A-Za-z0-9]*

2. Especificação FLEX

ponto e vírgula	“,” ;
tipo simples	bool int float char double
inteiro	{DIGITO}+
valor bool	"FALSE" "TRUE"
palavra reservada	return print read
laço de repetição	while for
operador lógico condicional	if else
identificador	{ID}+
real	{DIGITO}+."{DIGITO}*
operação de adição ou or	"+" "-" or
operação de multiplicação ou and	"*" "/" and
parênteses	"(" ")"
colchetes	"[" "]"
chaves ou bloco de função	"{" "}"

outros	"! "@" #" "\$ "%" "&" ":"
comparador	"<" ">" "<=" ">=" "<>" "=="
definição de valor	"="
comentário de 1 linha	"//".
string	"\"({TEXTO} .)*\""
espaço em branco	" "
quebra de linha	"\n"
divisor	"." ","
caractere não reconhecido	.

3. Destaque para compilação/ambiente/linguagem

Compilação:

Tivemos dificuldade na ordenação dos tokens mas foi superado.

Ambiente:

Linux LUBUNTU

4. Análise do exemplo

```

1 bool teste = FALSE;
2 bool teste2 = 0;
3 int valor1 = 0;
4 int valor2 = 0;
5 float valor_r = 0.0;
6 int vetor[10]: 0 0 0 0 0 0 0 0 0 0;
7 int matriz [10];
8
9 int main () {
10     int a, g, i;
11     a = 5;
12     vetor[a] = 80;
13     print vetor[80]; // ??? o que fazer?
14
15     print "Digite um numero: \n";
16     read a;

```

```

17
18     while (i<10): {
19         i = inc(i,1);
20         a = inc(a,1);
21     }
22
23     print "Incrementado algumas vezes a fica \" a \"\n";
24
25     for (g:1,10,1):
26         print ".";
27     print "\n";
28
29     if (a==15): // Incluir operadores relacionais...
30         print "A=15\n";
31
32     if ( a==18 or a==20 ): // Incluir operadores lógicos...
33         print "A igual a 18 ou 20\n";
34
35     if (i==100):
36     {
37         print "Decorando o texto...\n";
38     }
39     else:
40         print "OK!\n"
41 }
42
43 int inc ( int x, int n ){
44     return x+n
45 }
46

```

Podemos identificar os seguintes problemas no exemplo:

- Na linha 13, o vetor de int vetor [] tem seu ponteiro apontando a posição 80, porém como o tamanho máximo do vetor é 10, o seu print voltará um lixo de memória.
- Na linha 25, a condição inicial, e a condição de parada estão erradas para o loop *for*.
- Na linha 29 e 32, os laços de *if* não tem indentação {} para definir onde a sintaxe dos laços começam ou acabam, assim como na linha 39 o laço *e/se* não tem indentação de começo.

- Na linha 40, o print também falta ; no final para printar o valor.
- Na linha 44, o return x+n falta um ; no final para retornar o valor.