

Laboratório 06:

Interpretador para a Linguagem COMP-ITA 2017

Alunos: Gabriel Brito e Lucas França de Oliveira

Professor: Fábio C. Mokarzel

Disciplina: CES-41

20 de Maio de 2017

**Introdução**

A atividade realizada consistiu em implementar o interpretador de código intermediário da linguagem COMP-ITA 2017*.* A fim de demonstrar o seu bom funcionamento, será fornecido como entrada vários programas escritos na referida linguagem e com as respectivas saídas impressas.

**Desenvolvimento**

Para a realização da atividade, foram adaptados os arquivos da atividade de laboratório 5: “lab5.l” e “lab5.y”.

**Analisadores**

Os analisadores léxico e sintático produzidos durante a realização da atividade encontram em anexo, a fim de não poluir o texto do relatório. Os arquivos são “lab6.l” (léxico) e “lab6.y” (sintático).

**Testes**

A fim de demonstrar o bom funcionamento do gerador de código intermediário, foram realizados vários testes com diferentes programas escrito na linguagem COMP-ITA 2017. Para cada teste, será exibido no relatório apenas os trechos escritos na tela através do comando “write”. Os testes encontram-se a seguir:

**1. Programa “teste” do arquivo “lab6\_teste\_oper-arit”, para verificar o bom funcionamento das operações da linguagem com escalares**

Entrada:

program teste;

var

int a, b, c, i, j; float k; logic b1;

int A[5][4];

{

write ("Valor 1: ", 14, "; Valor 2: ", 15.2, "; Valor 3: ", true);

write ("Valor de 3+4: ", 3+4);

write ("Valor de 3.4+6: ", 3.4+6);

write ("Valor de 3-4: ", 3-4);

write ("Valor de 3.4-6: ", 3.4-6);

write ("Valor de char 7 menos char 0: ", '7'-'0');

write ("Valor de 3\*24: ", 3\*24);

write ("Valor de 3.4\*6: ", 3.4\*6);

write ("Valor de 35/7: ", 35/7);

write ("Valor de 25.5/5: ", 25.5/5);

write ("Valor de 49%6: ", 49%6);

write ("Valor de -3: ", ~3);

write ("Valor de -49/7.3: ", ~49/7.3);

write ("Resultado de -4.5<65: ", ~4.5<65);

write ("Resultado de -4.5>65: ", ~4.5>65);

write ("Resultado de -4.5<=65: ", ~4.5<=65);

write ("Resultado de -4.5<=-4.5: ", ~4.5<=~4.5);

write ("Resultado de -4.5>=65: ", ~4.5>=65);

write ("Resultado de -4.5>=-4.5: ", ~4.5>=~4.5);

write ("Resultado de -4.5==65: ", ~4.5==65);

write ("Resultado de -4.5==-4.5: ", ~4.5==~4.5);

write ("Resultado de TRUE==FALSE: ", true==false);

write ("Resultado de -4.5!=65: ", ~4.5!=65);

write ("Resultado de -4.5!=-4.5: ", ~4.5!=~4.5);

write ("Resultado de TRUE!=FALSE: ", true!=false);

write ("Resultado de TRUE||FALSE: ", true||false);

write ("Resultado de TRUE&&FALSE: ", true&&false);

write ("Resultado de !TRUE: ", !true);

write ("Resultado de !FALSE: ", !false);

}

Saída:

Escrevendo:

Valor 1: 14; Valor 2: 15.2; Valor 3: TRUE

Escrevendo:

Valor de 3+4: 7

Escrevendo:

Valor de 3.4+6: 9.4

Escrevendo:

Valor de 3-4: -1

Escrevendo:

Valor de 3.4-6: -2.6

Escrevendo:

Valor de char 7 menos char 0: -156

Escrevendo:

Valor de 3\*24: 72

Escrevendo:

Valor de 3.4\*6: 20.4

Escrevendo:

Valor de 35/7: 5

Escrevendo:

Valor de 25.5/5: 5.1

Escrevendo:

Valor de 49%6: 1

Escrevendo:

Valor de -3: -3

Escrevendo:

Valor de -49/7.3: -6.71233

Escrevendo:

Resultado de -4.5<65: TRUE

Escrevendo:

Resultado de -4.5>65: FALSE

Escrevendo:

Resultado de -4.5<=65: TRUE

Escrevendo:

Resultado de -4.5<=-4.5: TRUE

Escrevendo:

Resultado de -4.5>=65: FALSE

Escrevendo:

Resultado de -4.5>=-4.5: TRUE

Escrevendo:

Resultado de -4.5==65: FALSE

Escrevendo:

Resultado de -4.5==-4.5: TRUE

Escrevendo:

Resultado de TRUE==FALSE: FALSE

Escrevendo:

Resultado de -4.5!=65: TRUE

Escrevendo:

Resultado de -4.5!=-4.5: FALSE

Escrevendo:

Resultado de TRUE!=FALSE: TRUE

Escrevendo:

Resultado de TRUE||FALSE: TRUE

Escrevendo:

Resultado de TRUE&&FALSE: FALSE

Escrevendo:

Resultado de !TRUE: FALSE

Escrevendo:

Resultado de !FALSE: TRUE

**2. Programa “teste” do arquivo “lab6\_teste\_oper-arit”, para verificar o bom funcionamento das operações da linguagem com escalares**

Entrada:

program teste;

var

int a, b, c, i, j; float k, A[5][4][6]; logic b1, B[1][2];

{

A[1][1][1] = 1.0;

write("A[1][1][1] = ",A[1][1][1]);

A[2][2][2] = 2;

write("A[2][2][2] = ",A[2][2][2]);

A[3][3][3] = A[1][1][1] + A[2][2][2];

write("A[3][3][3] = ",A[3][3][3]);

A[3][3][3] = A[1][1][1] - A[2][2][2];

write("A[3][3][3] = ",A[3][3][3]);

A[3][3][3] = A[1][1][1] \* A[2][2][2];

write("A[3][3][3] = ",A[3][3][3]);

A[3][3][3] = A[1][1][1] / A[2][2][2];

write("A[3][3][3] = ",A[3][3][3]);

B[0][1] = 1 < 2;

write("B[0][1] = ",B[0][1]);

}

Saída:

Escrevendo:

A[1][1][1] = 1

Escrevendo:

A[2][2][2] = 2

Escrevendo:

A[1][1][1] = 1

Escrevendo:

A[3][3][3] = 3

Escrevendo:

A[3][3][3] = -1

Escrevendo:

A[3][3][3] = 2

Escrevendo:

A[3][3][3] = 0.5

Escrevendo:

B[0][1] = TRUE

**3. Programa “teste” do arquivo “indexTest”, carregando o arquivo “entrada 2017” com a string “1 2 3 4 5 6 7 8 9”.**

Entrada:

program teste;

var

int i, j, k, A[10][10];

float B[2][3][2], c;

{

i = 7; j = 5;

k = A[i-3][j+2] + 5;

write("k = ",k);

A[10-i][9-j] = i + j \* k;

write("A[10-i][9-j] = ", A[10-i][9-j]);

read(i, j, A[2][3], k);

write("i = ",i);

write("j = ",j);

write("k = ",k);

write("A[10][10] = ",A[10][10]);

read(A[1][2]);

write("A[1][2] = ",A[1][2]);

c = 5.5;

B[1][2][1] = c\*1.2;

write("B[1][2][1] = ",B[1][2][1]);

for(i=0;i<2;i = i+1){

for(j=0;j<3;j = j+1){

for(k = 0; k <2;k=k+1){

B[i][j][k] = i\*j\*k/2.0;

}

}

}

for(i=0;i<2;i = i+1){

for(j=0;j<3;j = j+1){

for(k = 0; k <2;k=k+1){

write("B[", i, "][", j, "][", k, "] = ",B[i][j][k]);

}

}

}

}

Saída:

Escrevendo:

k = 5

Escrevendo:

A[10-i][9-j] = 32

Lendo:

Lendo:

Lendo:

Escrevendo:

i = 1

Escrevendo:

j = 2

Escrevendo:

k = 4

Escrevendo:

A[10][10] = 0

Lendo:

Escrevendo:

A[1][2] = 5

Escrevendo:

B[1][2][1] = 6.6

Escrevendo:

B[0][0][0] = 0

Escrevendo:

B[0][0][1] = 0

Escrevendo:

B[0][1][0] = 0

Escrevendo:

B[0][1][1] = 0

Escrevendo:

B[0][2][0] = 0

Escrevendo:

B[0][2][1] = 0

Escrevendo:

B[1][0][0] = 0

Escrevendo:

B[1][0][1] = 0

Escrevendo:

B[1][1][0] = 0

Escrevendo:

B[1][1][1] = 0.5

Escrevendo:

B[1][2][0] = 0

Escrevendo:

B[1][2][1] = 1

**4. Programa “teste” do arquivo “pret012017”:**

program teste;

var

int a, b, c, i, j; float k; logic b1;

int A[5][4];

{

write ("Valor 1: ", 14, "; Valor 2: ", 15.2, "; Valor 3: ", true);

write ("Valor de 3+4: ", 3+4);

a = 1;

b = 2;

i = 3;

j = 4;

c = a + b + i + j + 20;

k = 12.6;

b1 = true;

write ("c = ", c, "; k = ", k, "; b1 = ", b1, ";");

i = 4; j = 5; b1 = true;

if (b1) {i = j + 6;} else {i = i + 15;}

write ("i = ", i);

i = 4; j = 5; b1 = true;

while (b1) {i = i + j; b1 = false;}

write ("i = ", i);

i = 3;

k = 20.3;

write ("i = ", i, "; k = ", k);

while (i < k) {

i = i + 4;

write ("i = ", i, "; k = ", k);

}

write ("Valor de i+k: ", i+k);

i = 0;

repeat{

i = i+1;

}while(i < 10);

write("Valor de i: ",i);

}

Saída:

Escrevendo:

Valor 1: 14; Valor 2: 15.2; Valor 3: TRUE

Escrevendo:

Valor de 3+4: 7

Escrevendo:

c = 30; k = 12.6; b1 = TRUE;

Escrevendo:

i = 11

Escrevendo:

i = 9

Escrevendo:

i = 3; k = 20.3

Escrevendo:

i = 7; k = 20.3

Escrevendo:

i = 11; k = 20.3

Escrevendo:

i = 15; k = 20.3

Escrevendo:

i = 19; k = 20.3

Escrevendo:

i = 23; k = 20.3

Escrevendo:

Valor de i+k: 43.3

Escrevendo:

Valor de i: 10

**5. Programa “teste” do arquivo “pret022017”, utilizando como entrada o arquivo “entrada2017” fornecido durante o semestre (string “ 30 20 3.14 1W”).**

Entrada:

program teste;

var

int i, j, k; float x, y; logic a, b; char m, n;

{

read (i, j, x, b);

write ("i: ", i, "; j: ", j, "; x: ", x, "; b: ", b);

k = i + j;

y = x + 3.4;

a = i < j;

write ("k: ", k, "; y: ", y, "; a: ", a);

read (m);

n = m + ' ';

write ("m: ", m, "; n: ", n);

}

Saída:

Lendo:

Escrevendo:

i: 30; j: 20; x: 3.14; b: TRUE

Escrevendo:

k: 50; y: 6.54; a: FALSE

Lendo:

Escrevendo:

m: W; n: #

**6. Programa “primos” para calcular os n primeiros primeiros números primos. Carregando a entrada com o número n = 30, temos:**

Entrada:

program primos;

var

logic achou; int n, div, resto, cont, num;

{

read (n);

write (n, " primeiros numeros naturais primos:\n\n");

if (n > 0) {

num = 1; cont = 0;

while (cont < n) {

num = num + 1; div = 2; achou = false;

while (achou == false && div \* div <= num) {

resto = num % div;

if (resto == 0) achou = true;

else div = div + 1;

}

if (achou == false) {

write (num); cont = cont + 1;

}

}

}

}

Saída:

Lendo:

Escrevendo:

30 primeiros numeros naturais primos:\n\n

Escrevendo:

2

Escrevendo:

3

Escrevendo:

5

Escrevendo:

7

Escrevendo:

11

Escrevendo:

13

Escrevendo:

17

Escrevendo:

19

Escrevendo:

23

Escrevendo:

29

Escrevendo:

31

Escrevendo:

37

Escrevendo:

41

Escrevendo:

43

Escrevendo:

47

Escrevendo:

53

Escrevendo:

59

Escrevendo:

61

Escrevendo:

67

Escrevendo:

71

Escrevendo:

73

Escrevendo:

79

Escrevendo:

83

Escrevendo:

89

Escrevendo:

97

Escrevendo:

101

Escrevendo:

103

Escrevendo:

107

Escrevendo:

109

Escrevendo:

113

**Conclusão**

Após bastante trabalho, o interpretador para a linguagem COMP-ITA 2017 foi concluído e funciona bem, inclusive rodando programas fornecidos pelo professor, como o programa para calcular os “n” primeiros números primos.