

## PARSER PARA RECONHECER EXPRESSÕES COM ELEMENTOS LÓGICOS, RELACIONAIS, NUMÉRICOS E STRINGS

Considerando o Parser produzido na aula de 10/02/2022, realize adaptações para que obtenham-se retornos booleanos nas expressões que seguem abaixo, exemplificadas no condicional “if”. Assumir os operadores Relacionais com maior precedência em relação ao operador AND/∧ booleano, e este com maior precedência em relação ao operador OR/∨. DICA: “Explodir” a void termo() em termo\_alfa() e termo\_beta(). Dessa maneira, obtêm-se o gerenciamento adequado de precedências.

### **Entrada 01 a ser reconhecida:**

```
$BEGIN_PROGRAM
BOOL Valor1
BOOL Valor2
INT X
INT Y
INT Z
FLOAT W
STRING Nome

X=-5
Y=10
Z=15
Valor1=false
Valor2=true
W=21.2

if (Valor1 ∧∧ Valor2 ∨∨ X<Z ) then
{
  Z=X
}
else
{
  W=50.12
}

$END_PROGRAM
```

### **Retorno da expressão no condicional “if”:**

Comando IF

Return TRUE

**Entrada 02 a ser reconhecida:**

```
$BEGIN_PROGRAM
```

```
BOOL Valor1
```

```
BOOL Valor2
```

```
INT X
```

```
INT Y
```

```
INT Z
```

```
FLOAT W
```

```
STRING Nome
```

```
X=-5
```

```
Y=10
```

```
Z=15
```

```
Valor1=false
```

```
Valor2=true
```

```
W=21.2
```

```
if (Valor1 && Valor2 || X<Z && Y>=Z ) then
```

```
{
```

```
  Z=X
```

```
}
```

```
else
```

```
{
```

```
  W=50.12
```

```
}
```

```
$END_PROGRAM
```

**Retorno da expressão no condicional “if”:**

Comando ELSE

Return FALSE

**Entrada 03 a ser reconhecida:**

```
$BEGIN_PROGRAM
```

```
BOOL Valor1
```

```
BOOL Valor2
```

```
INT X
```

```
INT Y
```

```
INT Z
```

```
FLOAT W
```

```
STRING Nome
```

```
X=-5
```

```
Y=10
```

```
Z=15
```

```
Valor1=false
```

```
Valor2=true
```

```
W=21.2
```

```

if (Valor1 && Valor2 || X<Z && Y<=Z && W<21.3) then
{
  Z=X
}
else
{
  W=50.12
}
$END_PROGRAM

```

#### **Retorno da expressão no condicional “if”:**

Comando IF

Return TRUE

#### **Entrada 04 a ser reconhecida:**

```

$BEGIN_PROGRAM
BOOL Valor1
BOOL Valor2
INT X
INT Y
INT Z
FLOAT W
STRING Nome

```

```

X=-5
Y=10
Z=-15
Valor1=false
Valor2=true
W=21.23

```

```

if (Valor1 || X>Z && Y<=Z && Valor2 && W<70.0) then
{
  Z=X
}
else
{
  W=50.12
}
$END_PROGRAM

```

#### **Retorno da expressão no condicional “if”:**

Comando ELSE

Return FALSE

**Entrada 05 a ser reconhecida:**

```
$BEGIN_PROGRAM  
BOOL Valor1  
BOOL Valor2  
INT X  
INT Y  
INT Z  
FLOAT W  
STRING Nome
```

```
X=-5  
Y=10  
Z=-15  
Valor1=false  
Valor2=false  
W=21.23  
Nome='Jose'
```

```
if (Valor1 && Valor2 || X>Z && Y>=Z && ~Valor2 && W<70.0 && Nome!='Maria') then  
{  
  Z=X  
}  
else  
{  
  W=50.12  
}  
$END_PROGRAM
```

**Retorno da expressão no condicional “if”:**

Comando IF

Return TRUE

**Entrada 01 a ser rejeitada:**

```
$BEGIN_PROGRAM  
BOOL Valor1  
BOOL Valor2  
INT X  
INT Y  
INT Z  
FLOAT W  
STRING Nome
```

```
X=-5  
Y=10  
Z=-15  
Valor1=false
```

```
Valor2=false
W=21.22
```

```
if (Valor1 && Valor2 || > X>Z && Y>Z ) then
{
  Z=X
}
else
{
  W=50.12
}
$END_PROGRAM
```

#### **Retorno da expressão no condicional “if”:**

Syntactic Error: Encountered " <RELACIONAIS> "> "" at line 18, column 24.

Was expecting one of:

```
"~" ...
<SENTENCA_BOOLEANA> ...
<IDENTIFICADOR> ...
<STRING> ...
<NUM> ...
<DIGITO> ...
```

#### **Entrada 02 a ser rejeitada:**

```
$BEGIN_PROGRAM
BOOL Valor1
BOOL Valor2
INT X
INT Y
INT Z
FLOAT W
STRING Nome
```

```
X=-5
Y=10
Z=-15
Valor1=false
Valor2=false
W=21.22
```

```
if (Valor1 && Valor2 || X>Z && Y>=Valor1 ) then
{
  Z=X
}
else
{
  W=50.12
}
```

```
}  
$END_PROGRAM
```

**Retorno da expressão no condicional “if”:**

Semantic error: For input string: "false" / Equivalente

**Entrada 03 a ser rejeitada:**

```
$BEGIN_PROGRAM  
BOOL Valor1  
BOOL Valor2  
INT X  
INT Y  
INT Z  
FLOAT W  
STRING Nome
```

```
X=-5  
Y=10  
Z=-15  
Valor1=false  
Valor2=false  
W=21.22  
Nome='Jose'
```

```
if (Valor1 && Valor2 || X>Z && Y==Nome ) then  
{  
  Z=X  
}  
else  
{  
  W=50.12  
}  
$END_PROGRAM
```

**Retorno da expressão no condicional “if”:**

Semantic error: For input string: "Jose" / Equivalente

#### Entrada 04 a ser rejeitada:

```
$BEGIN_PROGRAM
BOOL Valor1
BOOL Valor2
INT X
INT Y
INT Z
FLOAT W
STRING Nome

X=-5
Y=10
Z=-15
Valor1=false
Valor2=false
W=21.22
Nome='Jose'

if (Valor1 && Valor2 || X>Z && Y>Z && Nome==Ana ) then
{
    Z=X
}
else
{
    W=50.12
}
$END_PROGRAM
```

#### Retorno da expressão no condicional “if”:

Semantic error: null, java.lang.NullPointerException at Tabelas.pesquisa\_valor(Tabelas.java:36)

<Ana deveria estar entre aspas, no entanto, a string está sendo assumida como sendo um identificador e não foi encontrado seu conteúdo na tabela de símbolos>.

**Observações Importantes:** As comparações na Expressao() podem ser realizadas somente entre tipos idênticos ou equivalentes de comparação, exemplo: INT com FLOAT pode ser comparável, no entanto, não devem ser plausíveis comparações entre STRING e INT/FLOAT ou BOOL e INT/FLOAT/STRING.

Na manipulação de Strings pode assumir apenas os operadores == e !=.

Quanto a negação “~” de elementos booleanos recomenda-se programar como um Token isolado e especificar sua sintaxe na void fator(), como sendo um identificador que pode ter ou não uma negação, e este identificador, caso tenha a negação, ser uma variável booleana: (<NEGACAO>)? <IDENTIFICADOR>.

No momento de testes garantir que ao alterar os elementos da expressão o retorno do condicional “if” também é alterado de forma coerente, ou seja, se de fato as expressões estão sendo avaliadas de acordo com as relações lógicas/relacionais/numéricas/strings existentes na maioria das linguagens de programação.