

# Warlock Compiler

V6.66



Compiladores 2018.1

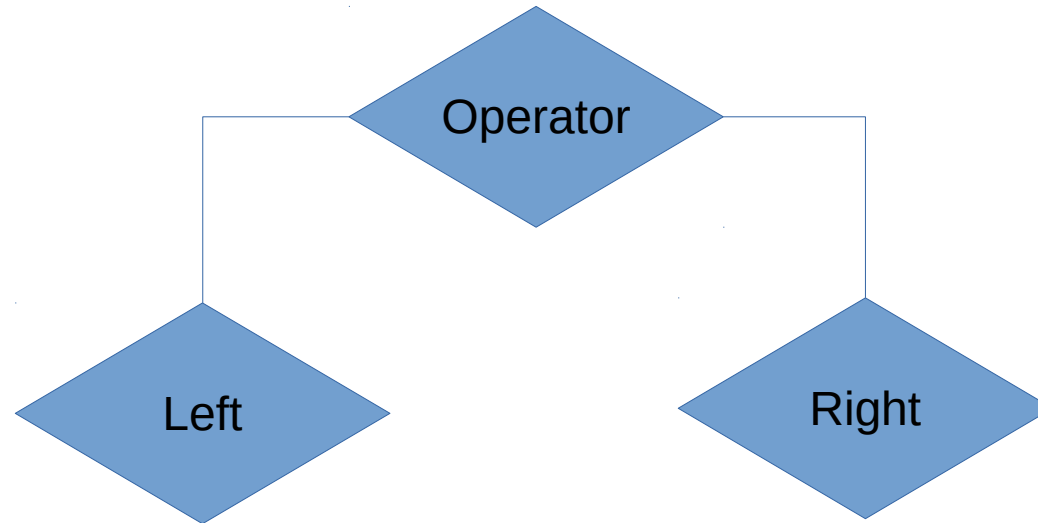
Grupo:

- Caio Saracuzza
- Eduardo Ramos

In memoriam:

- ~~Arthur Vieira~~
- ~~Beatriz Campos~~

# PEG



{A:=2}

```
{
  "left": "A",
  "operator": "ass",
  "right": "2"
}
```

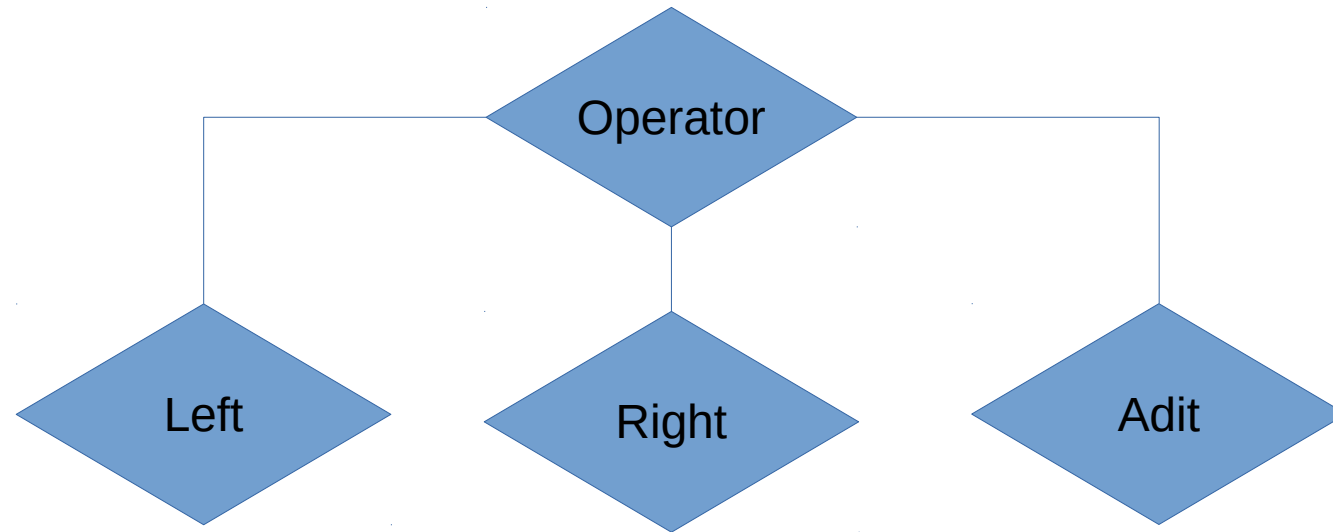
{A:=2+2}

```
{
  "left": "A",
  "operator": "ass",
  "right": {
    "left": "2",
    "operator": "add",
    "right": "2"
  }
}
```

{A:=2+2\*4+(1+1)}

```
{
  "left": "A",
  "operator": "ass",
  "right": {
    "left": {
      "left": "2",
      "operator": "add",
      "right": {
        "left": "2",
        "operator": "mul",
        "right": "4"
      }
    },
    "operator": "add",
    "right": {
      "left": "1",
      "operator": "add",
      "right": "1"
    }
  }
}
```

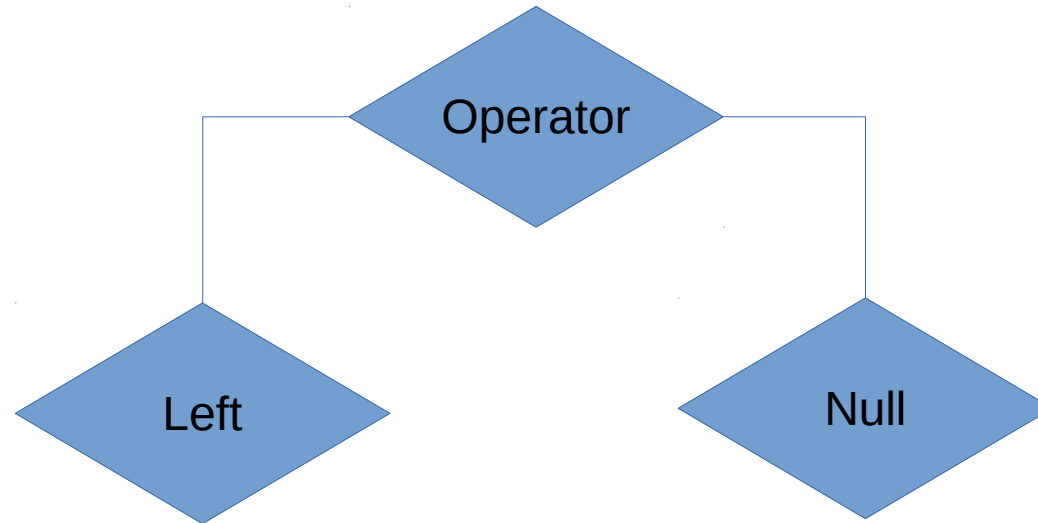
# PEG



```
{  
  "left": {  
    "left": "A",  
    "operator": "eq",  
    "right": "2"  
  },  
  "operator": "if",  
  "right": {  
    "left": "B",  
    "operator": "ass",  
    "right": "1"  
  },  
  "adit": {  
    "left": "B",  
    "operator": "ass",  
    "right": "A"  
  }  
}
```

```
{  
  if A==2  
    B:=1  
  else  
    B:=A  
}
```

# PEG



{A:=~B}

```
{
  "left": "A",
  "operator": "ass",
  "right": {
    "left": "B",
    "operator": "neg",
    "right": null
  }
}
```

# Step by Step

Expressão Aritmética

{A:= 2+2}

```
{  
  "left": "A",  
  "operator": "ass",  
  "right": {  
    "left": "2",  
    "operator": "add",  
    "right": "2"  
  }  
}
```

1º – Inicializamos o SMC colocando o retorno do parser direto na pilha de controle.

<[ ], { }, [ {left: 'A', operator: 'ass', right: [Object] } ] >

2º – Pegamos o primeiro elemento da pilha de Controle e analisamos pra saber em qual caso das relações de transição ela se encaixa e fazemos as operações necessárias para deixar da forma esperada.

$$C := I \quad \langle S, M, v := e C \rangle \Rightarrow \langle v S, M, e := C \rangle$$

<['A'], { }, [ 'ass', { left: '2', operator: 'add', right: '2' } ] >

3º – Repetimos o Passo 2 até chegar ao resultado final.

$$E \xrightarrow{+} I \quad \langle S, M, e \xrightarrow{+} e' C \rangle \Rightarrow \langle S, M, e e' \xrightarrow{+} C \rangle$$

<['A'], { }, [ 'ass', 'add', '2', '2' ] >



# Step by Step

Expressão Aritmética

{A:= 2+2}

```
{
  "left": "A",
  "operator": "ass",
  "right": {
    "left": "2",
    "operator": "add",
    "right": "2"
  }
}
```

4º – Repetimos o Passo 2 até chegar em algum operador.

$En \quad \langle S, M, n C \rangle \Rightarrow \langle n S, M, C \rangle$

<['A', '2'], { }, [ 'ass', 'add', '2' ] >

<['A', '2', '2'], { }, [ 'ass', 'add' ] >

5º – Repetimos o Passo 2.

$E \xrightarrow{+}_* E \quad \langle m' m S, M, \xrightarrow{+}_* C \rangle \Rightarrow \langle n S, M, C \rangle$

<['A', '4'], { }, [ 'ass' ] >

6º – Finalmente.

$C := E \quad \langle m v S, M, := C \rangle \Rightarrow \langle S, M[m/v], C \rangle$

<[ ], { 'A' => 4 }, [ ] >



