

FSheila - Primeira Entrega

Erick Grilo Max Fratane Vítor Lourenço

Universidade Federal Fluminense

28 de Abril de 2018

Desafios e Dificuldades

- Gordon D. Plotkin: *falou, falou e falou e nada disse*
- SMC: Trabalhar com operadores derivados de *Cmd*
- BPLC: Montar a priorização da árvore de parsing

Desafios e Dificuldades

Entrada: `3 < 3 and 3 == 3`

Parser: `And (Leb (Number 3,Number 3),
Eq (Number 3,Number 3))`

C: `["Eq"; "Leb"; "And"]`

S: `[Number 3; Number 3; Number 3; Number 3]`

Type Cmd: `Sheila of string; XSheila of string`

Stack: `System.Collections.Generic`

- Definição de tipos
- Regras para os comandos
 - Expressões aritméticas e booleanas;
 - Estruturas de desvio e iteração

```
member this.generalRule = this.assignRule |-  
this.loopRule |- this.seqRule |- this.ifRule |-  
this.calcOp |- this.boolOp
```

Parser - Expressões aritméticas e booleanas

- Aritmética

$3 * 3 + 6 / 2$

Add (Multiply (Number 3,Number 3),
Divide (Number 6,Number 2))

- Booleana

$3 < 3$ and true and false

And (Leb (Number 3,Number 3),And (Boolean true,Boolean
false))

Parser - Repetição

```
while  $\sim(x == 0)$ {  
     $y := y * x$ ;  
     $x := x - 1$ ;  
}
```

\Rightarrow

```
Loop  
  (Neg (Eq (Id "x", Number 0)),  
   Seq  
     (Assign ("y", Multiply (Id "y", Id "x")),  
      Assign ("x", Subtract (Id "x", Number 1))))
```

Parser - Desvio

```
if 7 <= 9 {  
    sheila := 3;  
    sheila2 := 4 + 5  
else sheila := 4}
```

```
(Leq (Number 7, Number 9),  
  Seq (Assign ("sheila", Number 3), Assign ("sheila",  
    Add(Number 4, Number 5))),  
  Assign ("sheila", Number 4))}
```

- Parsering no modelo da BPLC

tufak := $6 / 2 * 3 + (-4) * 1 - 5$

```
Assign ("tufak", Subtract (Add (Multiply (Divide (Number
6,Number 2),Number 3),Multiply (Number -4,Number 1)),Number
5))
```


- Operadores aritméticos
- Operadores booleanos
- Operadores atribuição

Obrigado!

FSheila - Primeira Entrega

Erick Grilo Max Fratane Vítor Lourenço

Universidade Federal Fluminense

28 de Abril de 2018