Programação Funcional?

O que é Programação Funcional?

Eu tenho que deixar tudo de lado?

OOP?

Programação Procedural?

Logica??

Nada Disso

Motivos

Custo Por GB

Manutenbilidade

Resiliencia a Mudanças

Escalabilidade

Poder Computacional

Do que se trada Afinal?

Funções

Funções de Primeira Ordem

O que seria?

Passagem de Funções como Parametro

```
>>> def myapply(func, arg):
...     func( arg )
...
>>> myapply (lambda x: print (x),
...                  "Ola Mundo")
Ola Mundo
```

Retorno de Função

```
>>> def composition (some, func):
...     return lambda x: func(some(x))
...
```

```
>>> def mysoma (x):
...     return lambda y: x + y
...
>>> def mymulti (x):
...     return lambda y: x * y
...
```

```
>>> soma2 = mysoma ( 2 )
>>> multi3 = mymulti ( 3 )
>>> soma2mult3 = composition
(soma2, multi3)
>>> soma2mult3 ( 32 )
102
```

Closure

Constantes

Porque?

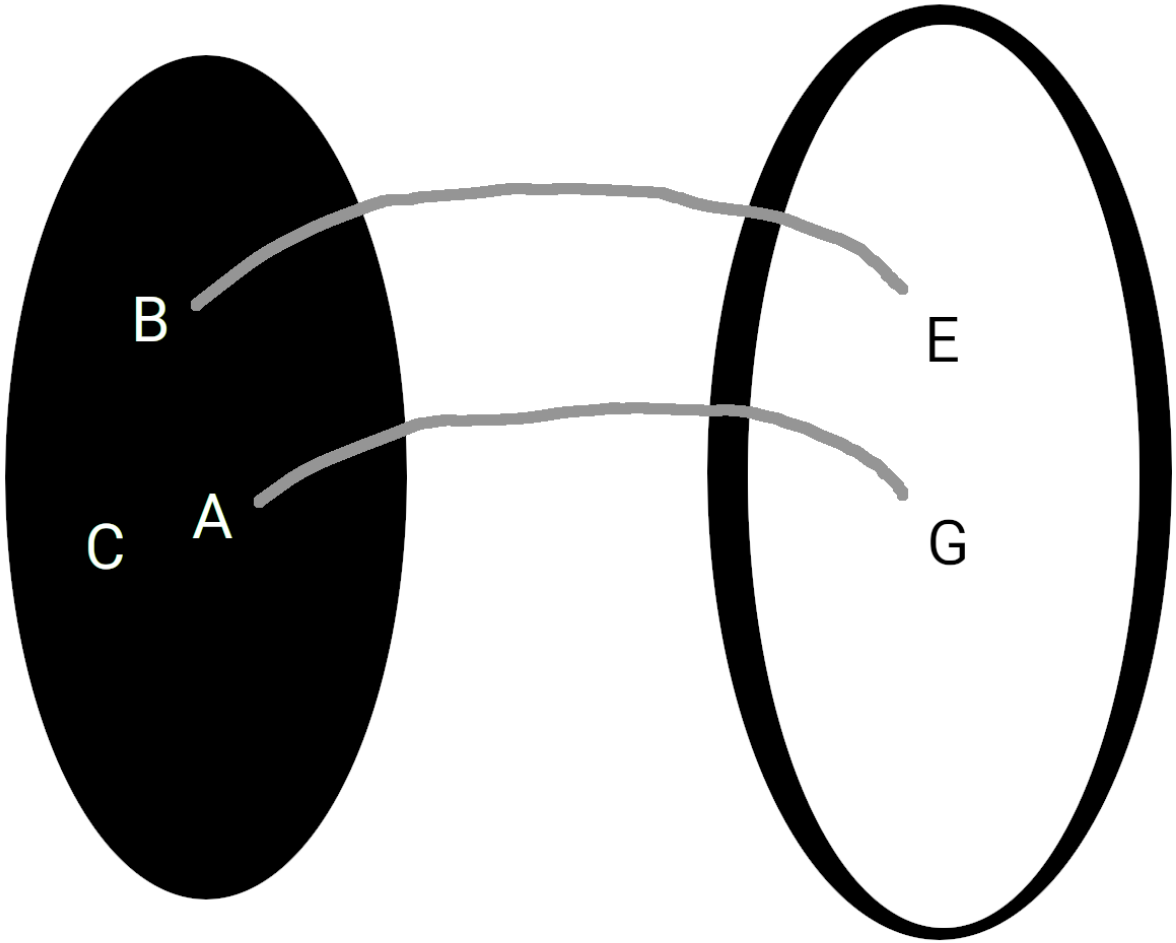
Side Effects

O que é isso ?

O que é uma função Afinal??

$$f(x) = x + 2$$

Conjuntos


B
A
C
E
G

O Que é Side Effects?

Mudança de estado

Turing Machine

Estate Machine

Exemplo

```
>>> def change (x, y):
...     soma = x + y
...     return soma
...
>>> change ( 2, 3 )
5
```

So Isso?

Acabou?

Recursão

O que é?

“Um padrão que soluciona um problema em tempo finito em termos de si mesmo”

Que Diabos é Isso.

```
>>> def fact ( num ):
...     if num = 1:
...         return num
...     return num * fact ( num - 1 )
...
>>> fact ( 5 )
120
```

Como funciona?

Stack

```
>>> fact ( 5 )
=> fact ( 5 )
=> 5 * fact ( 4 )
=> 5 * 4 * fact ( 3 )
=> 5 * 4 * 3 * fact ( 2 )
=> 5 * 4 * 3 * 2 * fact ( 1 )
=> 5 * 4 * 3 * 2 * 1
```

=> 5 * 4 * 3 * 2

=> 5 * 4 * 6

=> 5 * 24

=> 120

tem mais?

- Tail Recursion (Recursão em Calda)
- Tail Optimization
- Acumulation Passing by Style
- Continuation Passing by Style
- call/cc (Scheme)

Aplicações?

- Arvores
- map, reduce, filter
- Data Mining / Big Data
- Lambda Calculos
- Everything your Desire...

E...

- monads
- typetheory
- ...

É isso pessoal

https://github.com/haller218

/HylangPresentation