

# *Estruturas de Linguagem*

## *Interpretação de Programas* *(com programação funcional)*

**Francisco Sant'Anna**

`francisco@ime.uerj.br`

<http://github.com/fsantanna-uerj/EDL>

# Programa em C

- Como executar?
- Como representar?
- Como interpretar?

```
#include <stdio.h>

int main()
{
    int c = 1;

    while (c <= 10) {
        printf("%d ", c);
        c++;
    }

    return 0;
}
```

bibliotecas

funções

variáveis

loops

chamadas

tipos

expressões



# Expressões

- Combinação de constantes, variáveis, operadores, etc, que pode ser avaliada (reduzida) a um valor.
  - $1 + 10$
  - $x * y$
  - $(x > 10) \ \&\& \ (x < 100)$
- Expressões envolvendo constantes e operações aritméticas:
- Como representá-las usando Haskell?

# Expressões Aritméticas

- Expressões envolvendo números e operações aritméticas
- Como representá-las em Haskell?
  - somente números inteiros, subtração, adição
  - sem variáveis
- Como avaliá-las em Haskell?
  - `avalía :: Exp -> Int`

```
data Exp = Num Int
         | Add Exp Exp
         | Sub Exp Exp
deriving Show

e1 = Num 10
e2 = Add e1 e1
e3 = Sub (Num 100) e2

main = print e3
```

...

```
avalía :: Exp -> Int
avalía (Num v)      = v
avalía (Add e1 e2) = (avalía e1) + (avalía e2)
avalía (Sub e1 e2) = (avalía e1) - (avalía e2)

main = print (avalía e3)
```

**Teste**

**Próxima Sexta 24/05**

**Árvores e Expressões**