

# Representação Interação-Transformação para Regressão Simbólica

---

Prof. Fabricio Olivetti de França

Federal University of ABC

Center for Mathematics, Computation and Cognition (CMCC)

Heuristics, Analysis and Learning Laboratory (HAL)

12 de Julho de 2018



1. Introdução
2. Transformação-Interação
3. Lab Assistant
4. Experimentos
5. Conclusão

# Introdução

---

**Análise de Regressão** estuda a relação entre **variáveis dependentes** ( $y$ ) e uma ou mais **variáveis independentes** ( $x$ )

O objetivo é encontrar uma função  $\hat{f}(\mathbf{x})$  tal que:

$$\begin{array}{ll} \text{minimize} & \|\epsilon\|^2 \\ \hat{f}(\mathbf{x}) & \\ \text{subject to} & \hat{f}(\mathbf{x}) = f(\mathbf{x}) + \epsilon. \end{array}$$

A maioria dos algoritmos usam uma forma fixa  $\hat{f}(\mathbf{x}, \mathbf{w})$  e ajustam apenas os parâmetros livres  $\mathbf{w}$ .

Assume uma relação na forma de combinação linear das variáveis:

$$\hat{f}(\mathbf{x}, \mathbf{w}) = \mathbf{w} \cdot \mathbf{x}.$$

- É fácil de entender o impacto da mudança de valor de uma variável: uma mudança no valor de  $x_1$  altera a saída do sistema em  $w_1$ .
- Limitado a relações lineares.

# Perceptron de Múltiplas Camadas

O Perceptron de múltiplas camadas, em sua forma com apenas uma camada escondida, ajusta a função:

$$\hat{f}(\mathbf{x}, \mathbf{w}) = \mathbf{w}_2 \cdot g(\mathbf{w}_1 \cdot \mathbf{x}).$$

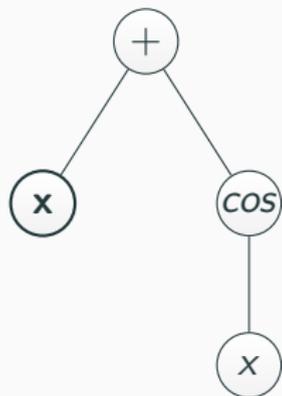
sendo  $g$  uma *função de ativação*.

- É um **aproximador universal**
- Apesar de ser uma forma fechada, é possível *evoluir* sua topologia podendo ser considerado um misto de forma fechada e livre.
- Qual o significado de  $\tanh(\tanh(\tanh(\tanh(\tanh(\dots)))$ ?

**Regressão Simbólica** busca pela forma da função e os valores ótimos dos parâmetros livres ao mesmo tempo que minimize o erro de aproximação.

Como objetivo secundário, busca também pela função mais simples possível.

- Algoritmos Evolutivos: Genetic Programming, Gene Expression, etc.
- Exploram todo o espaço de busca de funções.
- Árvores de Expressão, representação linear, gramática, etc.



```
p = arvoresAleatorias()
```

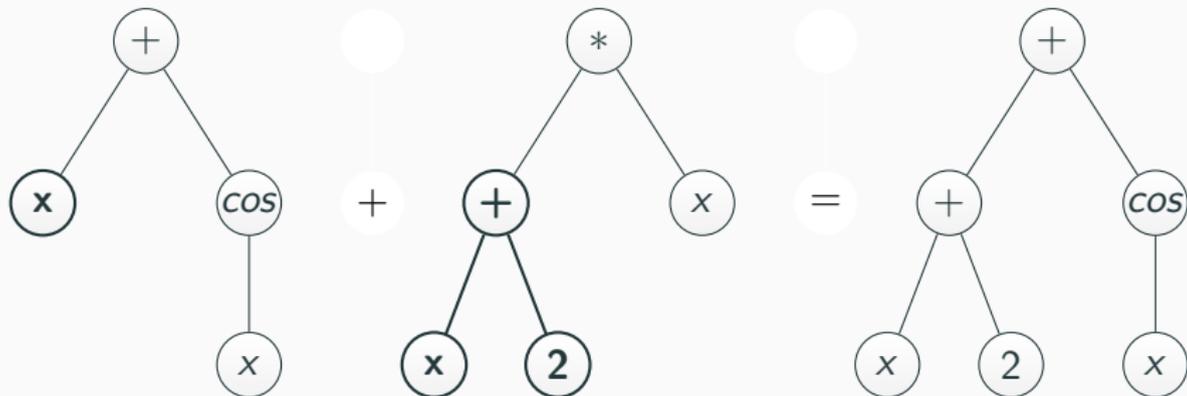
```
PG p | convergiu p = p  
    | otherwise   = do
```

```
    p' <- crossover(p)
```

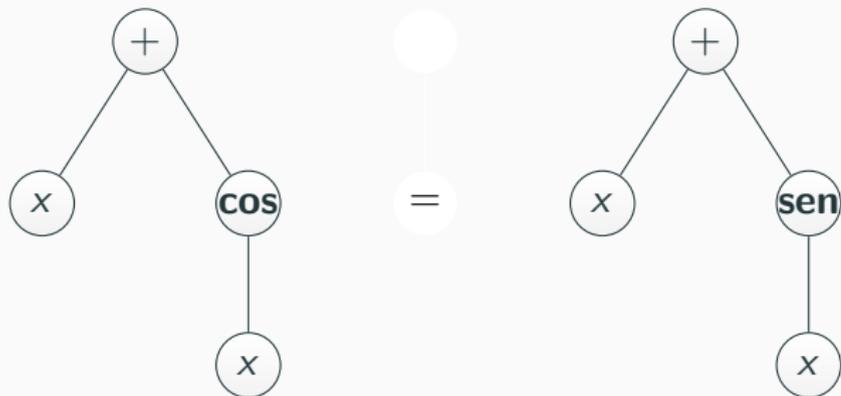
```
    p'' <- mutacao(p')
```

```
    PG (selecao(p''))
```

# Crossover



# Mutação



Problemas:

- Espaço de busca enorme e não suave.
- Uma pequena mudança na árvore de expressão (genótipo) pode levar a uma mudança enorme no comportamento da função de regressão (fenótipo).
- Muitos ótimos locais e globais (expressões equivalentes)

Exemplo:

$$f(x) = \frac{x^3}{6} + \frac{x^5}{120} + \frac{x^7}{5040}$$

$$f(x) = \frac{16x(\pi - x)}{5\pi^2 - 4x(\pi - x)}$$

$$\mathbf{f(x) = \sin(x)}.$$

Soluções:

- Introduzir uma medida de complexidade no objetivo (penalização ou multi-objetivo).
- Espaço de busca restrito.

# Transformação-Interação

---

Restringe a forma da função como uma **combinação linear** da aplicação de diferentes **funções de transformação** em **interações** das variáveis originais.

Essencialmente esse padrão de função:

$$\hat{f}(x) = \sum_i w_i \cdot t_i(p_i(x))$$

$$p(x) = \prod_{i=1}^d x_i^{k_i}$$

$$t_i = \{id, \sin, \cos, \tan, \sqrt{\cdot}, \log, \dots\}$$

# Transformação-Interação

Descrito como um tipo de dado algébrico:

```
IT x      = 0
           | Weight * (Term x) + (IT x)
```

```
Term x    = Trans (Inter x)
```

```
Trans     = a -> a
```

```
Inter x:xs = 1
            | x^s * Inter xs
```

Essencialmente uma lista ligada de termos...

Expressões válidas:

- $5.1 \cdot x_1 + 0.2 \cdot x_2$
- $3.5 \sin(x_1^2 \cdot x_2) + 5 \log(x_2^3/x_1)$

Expressões inválidas:

- $\tanh(\tanh(\tanh(w \cdot x)))$
- $\sin(x_1^2 + x_2)/x_3$

A complexidade pode ser ainda mais controlada incluindo restrições de número máximo de termos e valor máximo dos expoentes das interações.

Pode ser generalizada para qualquer outra tarefa de Programação Genética (expressões booleanas, árvores de decisão, síntese de programas).

Heurística simples para encontrar uma expressão IT:

```
symtree x leaves | stop      = best leaves  
                | otherwise = symtree x leaves'
```

**where**

```
leaves' = [expand leaf | leaf <- leaves]
```

```
symtree x [linearRegression x]
```

```
expand leaf = expand' leaf terms
  where terms = interaction leaf U transformation leaf

expand' leaf terms = node : expand' leaf leftover
  where (node, leftover) = greedySearch leaf terms
```

Supondo uma base de dados com 2 variáveis:  $x = \{x_1, x_2\}$ , iniciamos com uma Regressão Linear:

$$it = w_1 \cdot id(x_1^1 \cdot x_2^0) + w_2 \cdot id(x_1^0 \cdot x_2^1)$$

Criamos novos termos através das interações positivas e negativas entre todos os pares de termos:

$$t_1 = id(x_1^1 \cdot x_2^1)$$

$$t_2 = id(x_1^1 \cdot x_2^{-1})$$

$$t_3 = id(x_1^{-1} \cdot x_2^1)$$

E criamos mais termos com a substituição da função de transformação de cada termo:

$$t_4 = \sqrt{x_1^1 \cdot x_2^0}$$

$$t_5 = \sin(x_1^1 \cdot x_2^0)$$

$$t_6 = \sqrt{x_1^0 \cdot x_2^1}$$

$$t_7 = \sin(x_1^0 \cdot x_2^1)$$

Finalmente, aplicamos uma heurística para criar diversas expressões IT como a combinação dos novos termos:

$$it1 = w_1 \cdot id(x_1^1 \cdot x_2^0) + w_2 \cdot id(x_1^0 \cdot x_2^1) + w_3 \cdot \sqrt{x_1^0 \cdot x_2^1}$$

$$it2 = w_1 \cdot \sin(x_1^0 \cdot x_2^1) + w_2 \cdot \sqrt{x_1^1 \cdot x_2^0}$$

...

Em (de França, 2018)<sup>1</sup> esse algoritmo se mostrou melhor do que diversas abordagens na maioria das bases de *benchmark*.

---

<sup>1</sup>de França, Fabricio Olivetti. "A Greedy Search Tree Heuristic for Symbolic Regression." Information Sciences (2018).

| Function                                | expressible |
|---|-------------|
| $F_1 = x^3 + x^2 + 5 * x$               | Y           |
| $F_2 = x^4 + x^3 + x^2 + x$             | Y           |
| $F_3 = x^5 + x^4 + x^3 + x^2 + x$       | Y           |
| $F_4 = x^6 + x^5 + x^4 + x^3 + x^2 + x$ | Y           |
| $F_5 = \sin(x^2)\cos(x) - 1$            | N           |
| $F_6 = \sin(x) + \sin(x + x^2)$         | N           |
| $F_7 = \log(x + 1) + \log(x^2 + 1)$     | Y           |
| $F_8 = 5 * \sqrt{\ x\ }$                | Y           |

| Function  | expressible |
|---|-------------|
| $F_9 = \sin(x) + \sin(y^2)$                     | Y           |
| $F_{10} = 6\sin(x)\cos(y)$                      | N           |
| $F_{11} = 2 - 2.1\cos(9.8x)\sin(1.3w)$          | N           |
| $F_{12} = \frac{e^{-(x-1)^2}}{1.2+(y-2.5)^2}$   | N           |
| $F_{13} = \frac{10}{5+\sum_{i=1..5} (x_i-3)^2}$ | N           |
| $F_{14} = x_1x_2x_3x_4x_5$                      | Y           |
| $F_{15} = \frac{x^6}{x^3+x^2+1}$                | N           |
| $F_{16} = \frac{x}{1-\log(x^2+x+1)}$            | N           |
| $F_{17} = 100 + \log(x^2) + 5\sqrt{( x )}$      | Y           |

Algoritmos para comparação:

- Regressão Linear (LR)
- Regressão Polinomial (PF)
- Gradient Tree Boosting (GB)
- *neat*-GP
- Evolutionary Feature Synthesis

# Resultados

| algorithm | LR      | PF     | GB    | <i>neat</i> -GP | EFS   | SymTree     |
|-----------|---------|--------|-------|-----------------|-------|-------------|
| F1        | 17.48   | 0.00   | 0.49  | 108.91          | 6.43  | <i>0.00</i> |
| F2        | 131.99  | 0.00   | 1.83  | 284.70          | 6.62  | <i>0.00</i> |
| F3        | 483.36  | 0.00   | 8.52  | 1025.46         | 32.50 | <i>0.00</i> |
| F4        | 2823.94 | 247.72 | 58.39 | 813.25          | 47.55 | <b>0.00</b> |
| F5        | 0.32    | 0.30   | 0.03  | 0.65            | 0.25  | 0.23        |
| F6        | 0.79    | 0.60   | 0.07  | 1.95            | 0.57  | 0.58        |
| F7        | 0.02    | 0.00   | 0.00  | 0.58            | 0.01  | <i>0.00</i> |
| F8        | 2.22    | 0.56   | 0.03  | 0.95            | 0.14  | <b>0.00</b> |
| F9        | 0.78    | 0.58   | 0.18  | 4.45            | 0.24  | <b>0.00</b> |

## Resultados

| algorithm | LR    | PF   | GB    | <i>neat</i> -GP | EFS   | SymTree     |
|-----------|-------|------|-------|-----------------|-------|-------------|
| F10       | 2.26  | 2.05 | 0.95  | 8.74            | 1.88  | <b>0.72</b> |
| F11       | 0.81  | 0.83 | 0.86  | 2.91            | 0.85  | 0.82        |
| F12       | 0.06  | 0.06 | 0.01  | 2.00            | 0.04  | 0.06        |
| F13       | 0.06  | 0.04 | 0.03  | 1.41            | 0.05  | <i>0.03</i> |
| F14       | 61.26 | 0.00 | 71.87 | 3.62            | 90.50 | <i>0.00</i> |
| F15       | 22.41 | 8.63 | 7.00  | 188.11          | 12.57 | <i>8.88</i> |
| F16       | 5.21  | 5.82 | 5.50  | 6.26            | 6.71  | <i>6.31</i> |
| F17       | 3.88  | 1.34 | 0.11  | 3.78            | 0.26  | <b>0.00</b> |

## **Lab Assistant**

---

**Objetivo:** prova de conceito do algoritmo SymTree como uma ferramenta prática de análise de regressão.

Ferramenta web *client-side* desenvolvida com HTML + JavaScript.

SymTree no seu Browser! Alimente com dados, receba uma expressão matemática em troca!

<https://galdeia.github.io/>

## Data input

Data can be typed manually, or you can upload a local `csv` file. Optionally, the first line may contain the name of the variables. In [example input data](#) you can find some examples.;

### Manual input

```
m E
50 4.4937759e+18
170 1.527883806e+19
70 6.29128626e+18
190 1.707634842e+19
10 8.9875518e+17
90 8.08879662e+18
80 7.19004144e+18
40 3.59502072e+18
120 1.078506216e+19
```

### Upload local file

Choose File No file chosen

Use first line as variable names?

Use typed values

Use local file

I've chosen an example

Success! Data loaded.

Figura 1: Main Interface

Vertical pressure variation:

$$\Delta P = \rho \cdot g \cdot \Delta h$$

| $\rho$ | $\Delta h$ | $\Delta P$ |
|--------|------------|------------|
| 95     | 40         | 37266.6    |
| 40     | 35         | 13729.8    |
| 90     | 50         | 44131.5    |
| 85     | 15         | 12503.925  |
| 30     | 85         | 25007.85   |
| 15     | 65         | 9561.825   |
| 60     | 25         | 14710.5    |
| 55     | 60         | 32363.1    |
| 0      | 45         | 0.0        |
| 20     | 80         | 15691.2    |
| 5      | 75         | 3677.625   |
| 80     | 55         | 43150.8    |
| 65     | 30         | 19123.65   |
| 10     | 10         | 980.7      |
| 25     | 90         | 22065.75   |
| 50     | 20         | 9807.0     |
| 45     | 70         | 30892.05   |
| 70     | 95         | 65216.55   |

Mass-energy equivalence:

$$E = m \cdot c^2$$

| $m$ | $E$             |
|-----|-----------------|
| 50  | 4.4937759e+18   |
| 170 | 1.527883806e+19 |
| 70  | 6.29128626e+18  |
| 190 | 1.707634842e+19 |
| 10  | 8.9875518e+17   |
| 90  | 8.08879662e+18  |
| 80  | 7.19004144e+18  |
| 40  | 3.59502072e+18  |
| 120 | 1.078506216e+19 |
| 130 | 1.168381734e+19 |
| 180 | 1.617759324e+19 |
| 110 | 9.88630698e+18  |
| 30  | 2.69626554e+18  |
| 160 | 1.438008288e+19 |
| 20  | 1.79751036e+18  |
| 140 | 1.258257252e+19 |
| 150 | 1.34813277e+19  |
| 100 | 8.9875518e+18   |

Moment of inertia in a rectangle:

$$I_x = (1/12) \cdot b \cdot h^3$$

| $b$ | $h$ | $I_x$         |
|-----|-----|---------------|
| 90  | 30  | 202500.0      |
| 10  | 75  | 351562.5      |
| 170 | 85  | 8700104.16667 |
| 150 | 10  | 12500.0       |
| 70  | 25  | 91145.8333333 |
| 60  | 80  | 2560000.0     |
| 120 | 95  | 8573750.0     |
| 40  | 65  | 915416.666667 |
| 30  | 15  | 8437.5        |
| 50  | 45  | 379687.5      |
| 20  | 40  | 106666.666667 |
| 180 | 90  | 10935000.0    |
| 100 | 70  | 2858333.33333 |
| 160 | 55  | 2218333.33333 |
| 80  | 60  | 1440000.0     |
| 110 | 50  | 1145833.33333 |
| 190 | 35  | 678854.166667 |
| 140 | 20  | 93333.3333333 |

Figura 2: Main Interface

## Data input

Data can be typed manually, or you can upload a local `csv` file. Optionally, the first line may contain the name of the variables. In [example input data](#) you can find some examples.;

### Manual input

```
m E
50 4.4937759e+18
170 1.527883806e+19
70 6.29128626e+18
190 1.707634842e+19
10 8.9875518e+17
90 8.08879662e+18
80 7.19004144e+18
40 3.59502072e+18
120 1.078506216e+19
```

### Upload local file

Choose File No file chosen

Use first line as variable names?

Use typed values

Use local file

I've chosen an example

Success! Data loaded.

Figura 3: Main Interface

## Results

SymTree (Symbolic Tree)

IT-LS (Interaction Transformation Local Search)

IT-ES (Interaction Transformation Evolution Strategies)

Results:

Algorithm: SymTree

Expression: 89875518000000000.00\*x0

Score: 1

Time (in ms): 132.5000000698492

**Figura 4:** Main Interface

Check the behavior of the terms (or combinations) in relation to the target variable or in relation to the input variables.

T:

$\cos(x_0)$

$x_1^2$

X:

$x_0$

$x_1$

Plot T x Y graphic

Plot T x X graphic

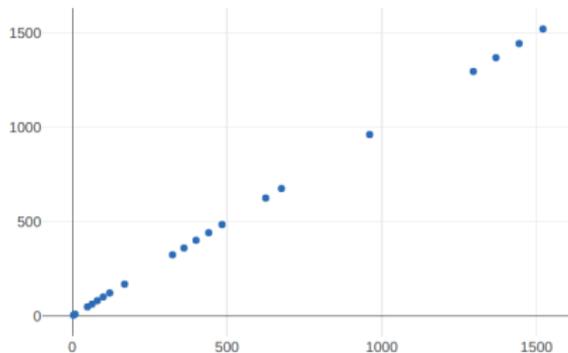


Figura 5: Main Interface

Check the behavior of the terms (or combinations) in relation to the target variable or in relation to the input variables.

T:

$\cos(x_0)$

$x_1^2$

X:

$x_0$

$x_1$

Plot T x Y graphic

Plot T x X graphic

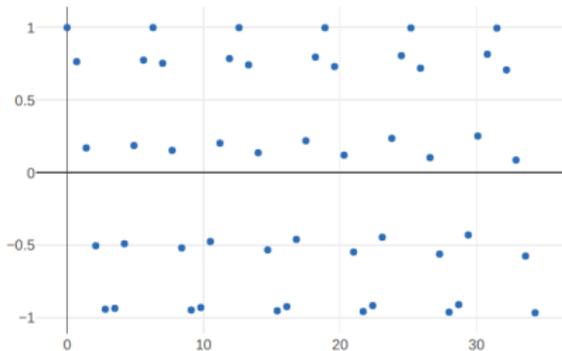


Figura 6: Main Interface

Além do algoritmo SymTree, o LabAssistant também possui implementação de uma busca local (IT-LS) e uma Estratégia Evolutiva (IT-ES) para gerar expressões IT, porém essas ainda estão em fase de ajuste.

## Compared to Eureka

---

| <b>Lab Assistant</b> | <b>Eureka</b>                            |
|----------------------|--|
| Interpreted          | Compiled                                 |
| Compatible Browsers  | GUI (Windows, Linux, OSX)                |
| Small data           | Large data                               |
| No pre-processing    | Pre-processing, function set, parameters |
| Single-core          | Multicore and Cloud Computing            |
| 2 different plots    | 9 different plots                        |
| Free                 | Commercial use only                      |

---

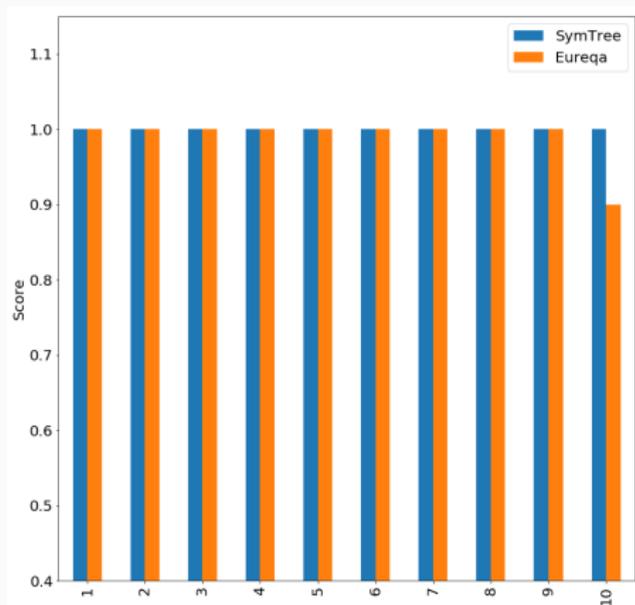
# Experimentos

---

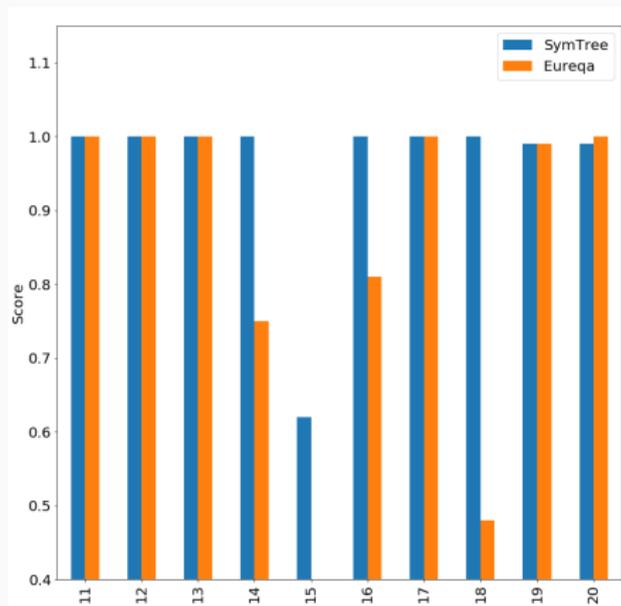
- 20 equações diferentes da Física e da Engenharia
  - 14 delas podem ser representadas na forma de uma expressão IT
- 30 execuções de cada algoritmo
- Comparação com o Eureka
- $\text{Score} = \frac{1}{1+MAE}$

- Sem nenhum pré-processamento (assim como o Lab Assistant)
- Com um tempo de execução total de 3 minutos (mais do que o necessário pelo Lab Assistant)

# Results



**Figure 7:** Score for the first 10 functions



**Figure 8:** Score for the next 10 functions

## Conclusão

---

A representação Interação-Transformação permite definir um espaço de busca de expressões matemáticas simples mas capaz de aproximar diversas bases de dados, sendo competitivo com algoritmos do estado-da-arte de regressão.

Além disso, o algoritmo SymTree é capaz de encontrar uma boa expressão IT com poucas iterações, sendo um algoritmo simples e computacionalmente leve.

Muitas possibilidades de estudos futuros:

- Generalizar a representação como uma tipo de dado algébrico
- Utilizar essa representação em outros contextos
- Aumentar o espaço de busca permitindo outras expressões simples ainda não compreendidas
- Criar novos algoritmos de busca para esse espaço de busca
- Muitos outros. . .

# O que estudar?

- Paradigma Funcional de programação
- Haskell
- Teoria das Categorias e dos Tipos
- Análise de Conceitos Formais

## Try it!

Você pode testar o algoritmo através do site:

<https://galdeia.github.io/>

Funciona até mesmo em Smartpones e Tablets!