

Lógica Fuzzy com Python

de Rafael Noboro Tominaga

Projeto do Edital 02/2018/PROPPI/PROEX Câmpus Florianópolis : Programa de Integração da Pesquisa e Extensão ao Ensino do Câmpus Florianópolis

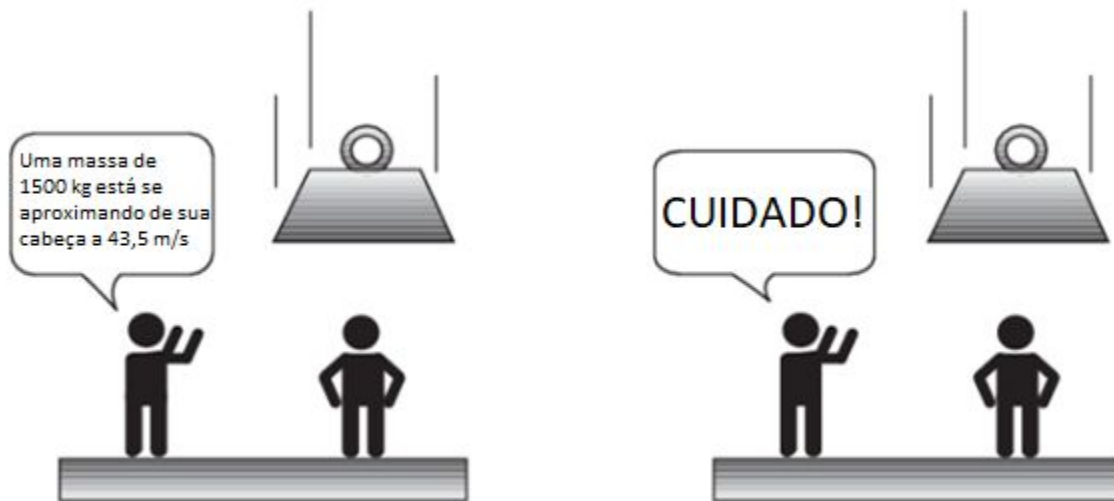
Apresentação disponível em
github.com/PECCE-IFSC

Oficina SNCT 2019

Sumário

- **Introdução**
- Exemplo do Chuveiro
- Comparação com Lógica Booleana
- Processos da Lógica Fuzzy
- Implementação em Python
- Outras Aplicações

Precisão x Relevância



Precisão x Relevância

Uma massa de
1500 kg está se
aproximando de sua
cabeça a 43,5 m/s



Definição

Lógica
Difusa

Lógica
Nebulosa

Lógica
Fuzzy

- Lógica multivalorada
- Emula a capacidade cognitiva humana

Sumário

- Introdução
- Exemplo do Chuveiro
- Comparação com Lógica Booleana
- Processos da Lógica Fuzzy
- Implementação em Python
- Outras Aplicações

Exemplo do Chuveiro

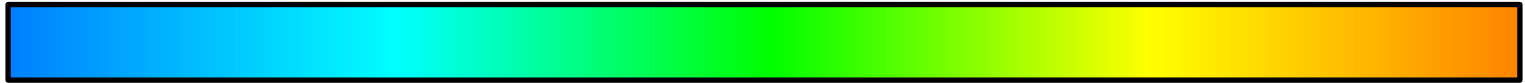
- Quando tomamos banho



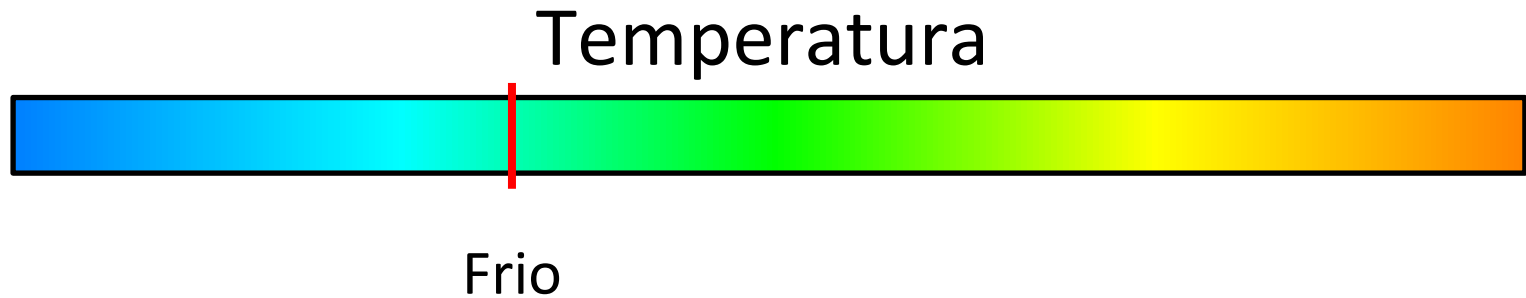
- Você é um controlador fuzzy!
- Variáveis de entrada:
 - Temperatura da água
 - Fluxo da água
- Variáveis de saída:
 - Regulador de temperatura
 - Válvula de fluxo

Exemplo do Chuveiro

Temperatura



Exemplo do Chuveiro



Exemplo do Chuveiro

Temperatura



Muito
Quente

Exemplo do Chuveiro

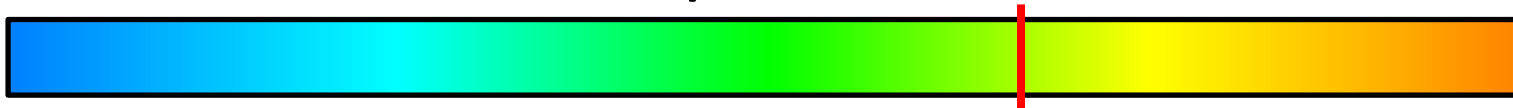
Temperatura



Menos
Quente

Exemplo do Chuveiro

Temperatura



Pouquinho
Menos
Quente

Exemplo do Chuveiro

Temperatura



Perfeito!

Sumário

- Introdução
- Exemplo do Chuveiro
- **Comparação com Lógica Booleana**
- Processos da Lógica Fuzzy
- Implementação em Python
- Outras Aplicações

Água está fria?

Lógica Booleana

- $\{0,1\}$
- Sim (**1**);
- Não (**0**).

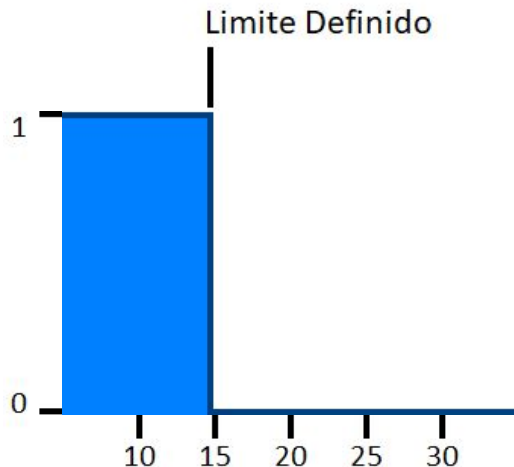
Lógica Fuzzy

- $\{x \in R \mid 0 \leq x \leq 1\}$
- Definitivamente frio (**1**);
- Muito frio (**0,75**);
- Medianamente frio (**0,5**);
- Pouco frio (**0,25**);
- Nem um pouco Frio (**0**).

Água está fria?

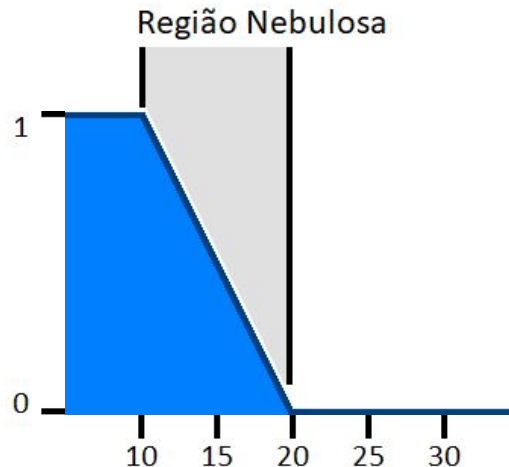
Lógica Booleana

- Estados definidos



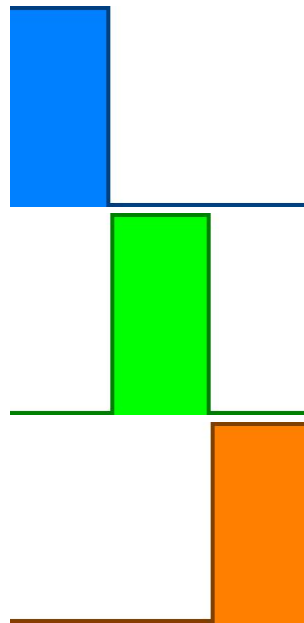
Lógica Fuzzy

- Graus de pertinência



Fria, morna ou quente?

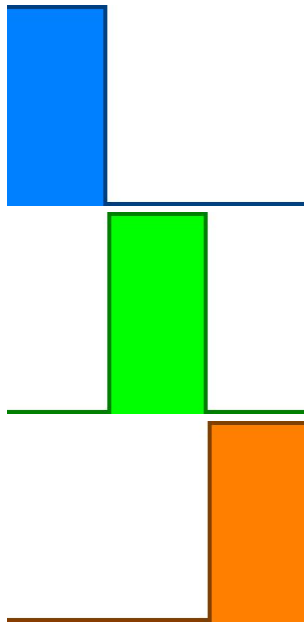
- Fria
- Morna
- Quente



Fria, morna ou quente?

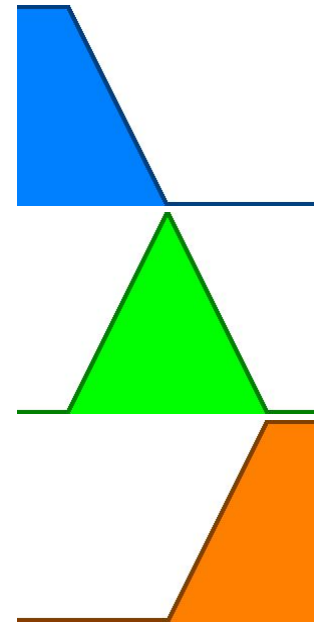
Lógica Booleana

- Fria
- Morna
- Quente



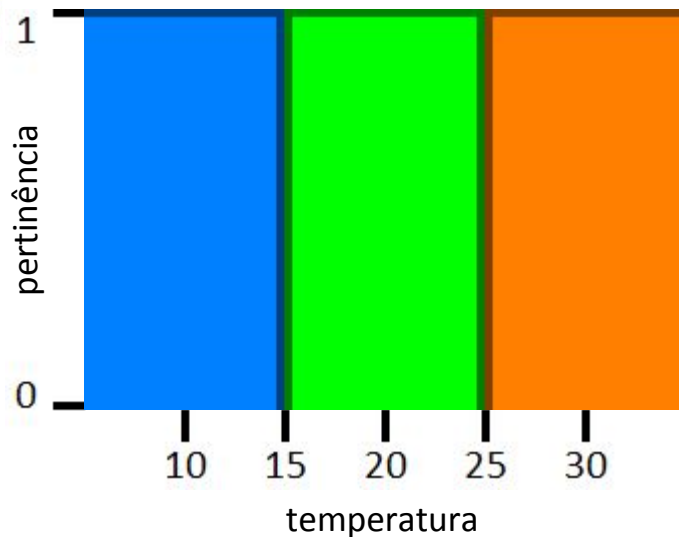
Lógica Fuzzy

- Fria
- Morna
- Quente



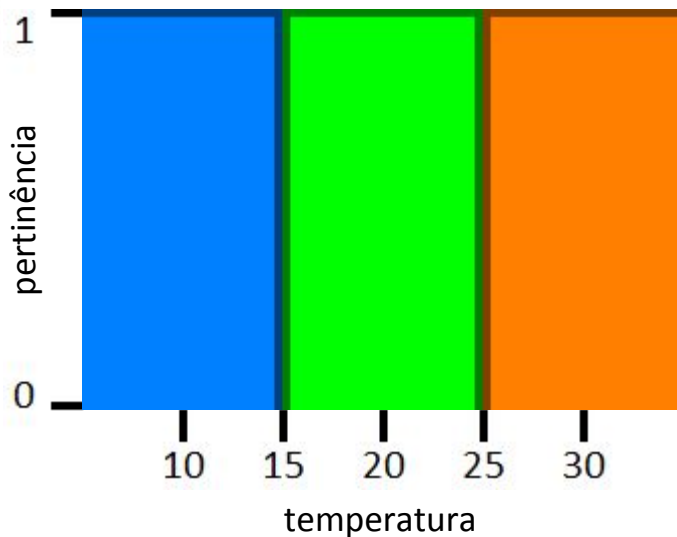
Fria, morna ou quente?

Lógica Booleana

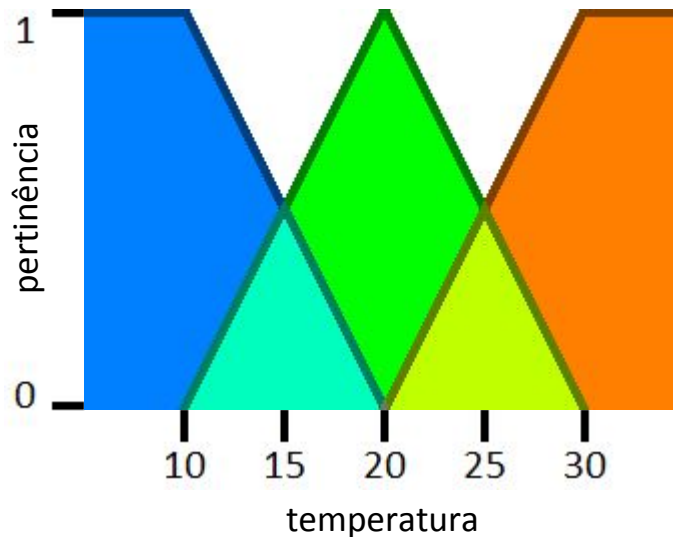


Fria, morna ou quente?

Lógica Booleana

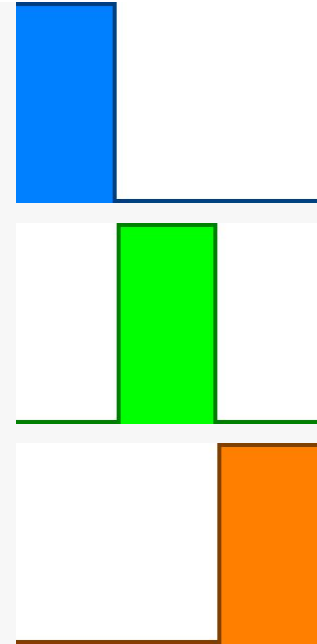


Lógica Fuzzy



Fria, morna ou quente?

```
[ ] def boolean(temperatura):  
    if temperatura <= 15:  
        frio = 1  
        morno = 0  
        quente = 0  
    else:  
        if temperatura <= 25:  
            frio = 0  
            morno = 1  
            quente = 0  
        else:  
            frio = 0  
            morno = 0  
            quente = 1  
    return [frio, morno, quente]
```



Fria, morna ou quente?

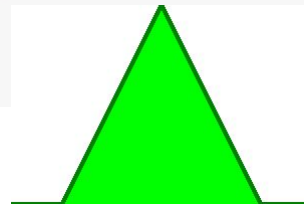
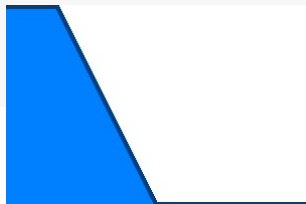
```
[ ] def fuzzy(temperatura):

    #frio
    if temperatura <= 10:
        frio = 1
    else:
        if temperatura <= 20:
            frio = -0.1*temperatura+2
        else:
            frio = 0

    #morno
    if temperatura <=10:
        morno = 0
    else:
        if temperatura <= 20:
            morno = 0.1*temperatura-1
        else:
            if temperatura <= 30:
                morno = -0.1*temperatura+3
            else:
                morno = 0

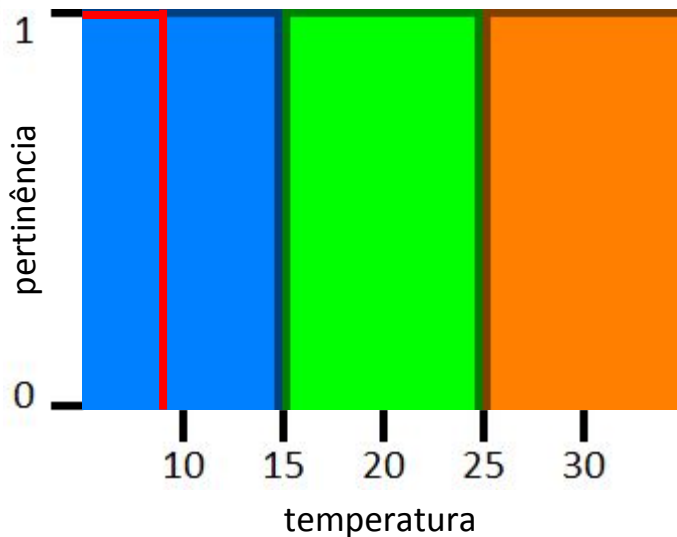
    #quente
    if temperatura <= 20:
        quente = 0
    else:
        if temperatura <= 30:
            quente = 0.1*temperatura-2
        else:
            quente = 1

    return [frio, morno, quente]
```

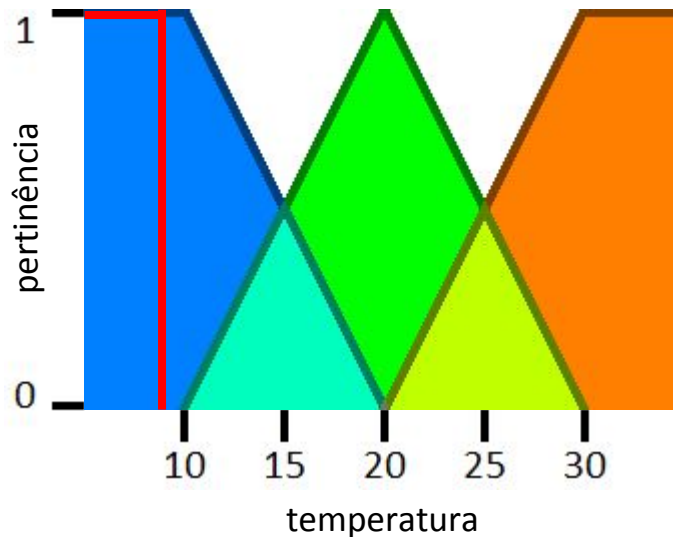


Fria, morna ou quente?

Boolean(9) = [1, 0, 0]

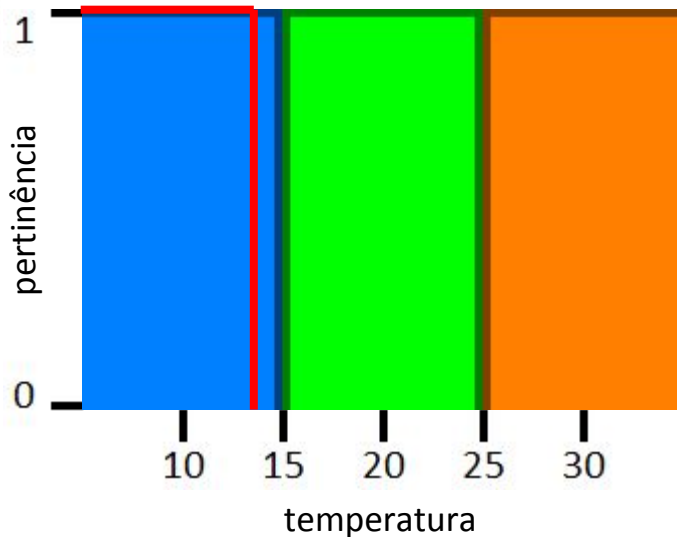


Fuzzy(9) = [1, 0, 0]

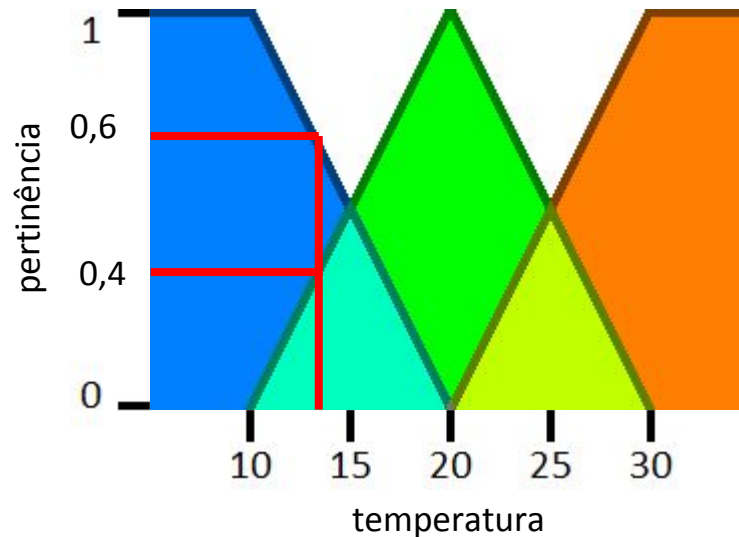


Fria, morna ou quente?

Boolean(14) = [1, 0, 0]

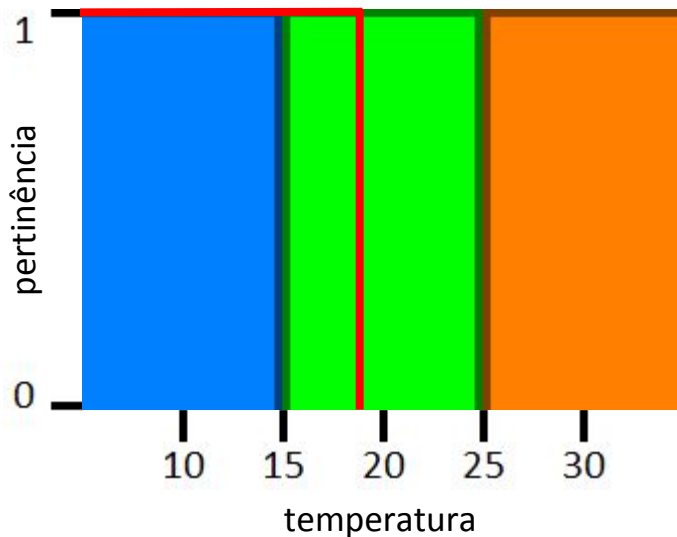


Fuzzy(14) = [0.6, 0.4, 0]

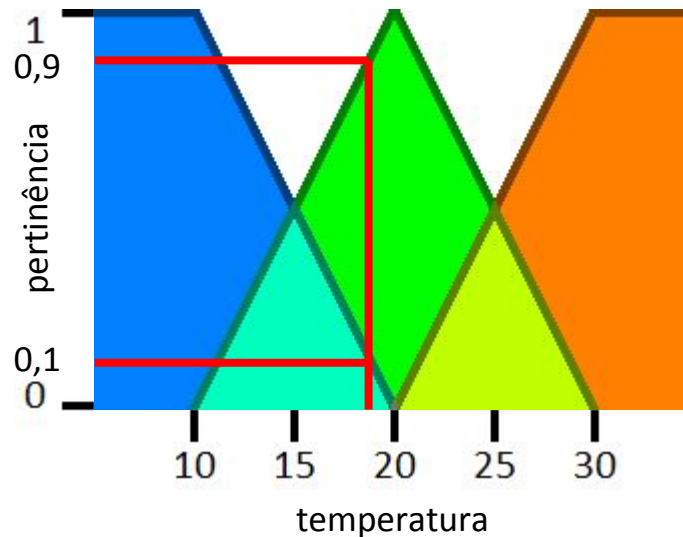


Fria, morna ou quente?

Boolean(19) = [0, 1, 0]



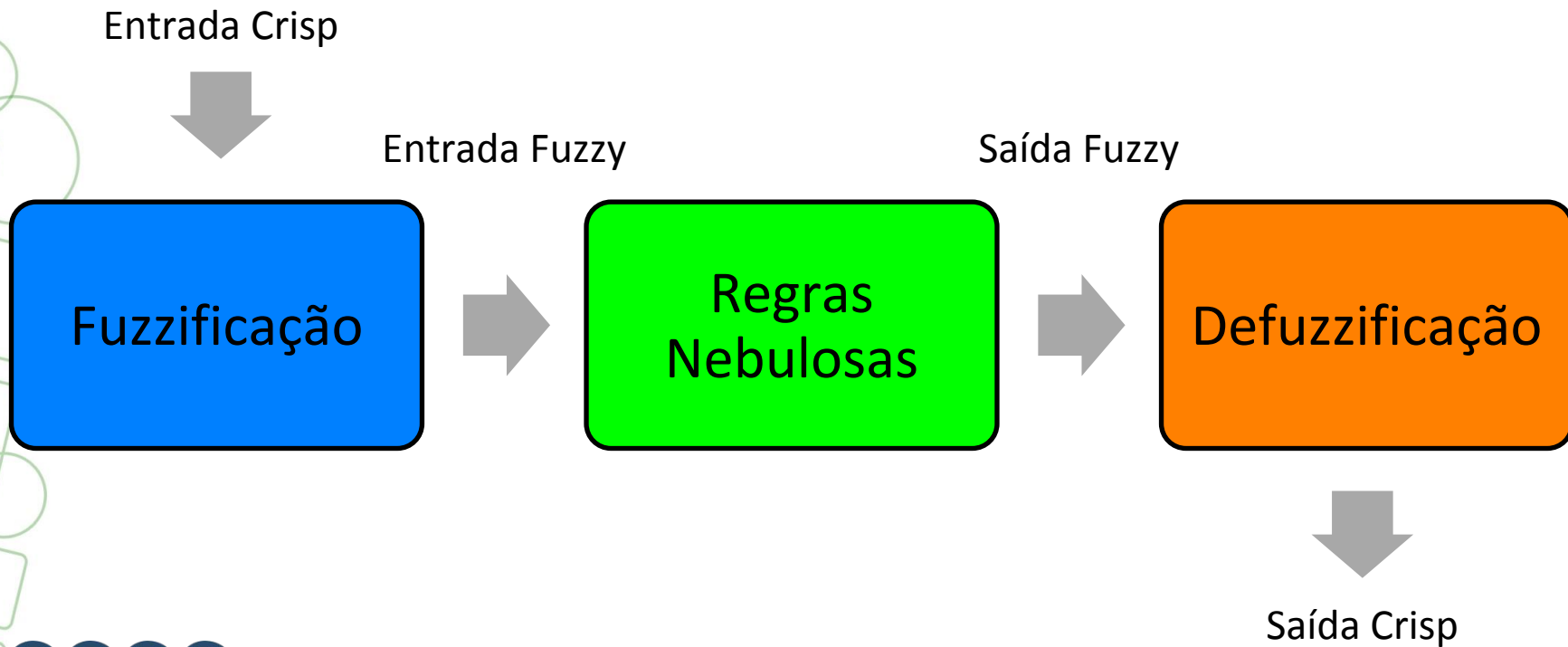
Fuzzy(19) = [0.1, 0.9, 0]



Sumário

- Introdução
- Exemplo do Chuveiro
- Comparação com Lógica Booleana
- **Processos da Lógica Fuzzy**
- Implementação em Python
- Outras Aplicações

Processos da Lógica Fuzzy

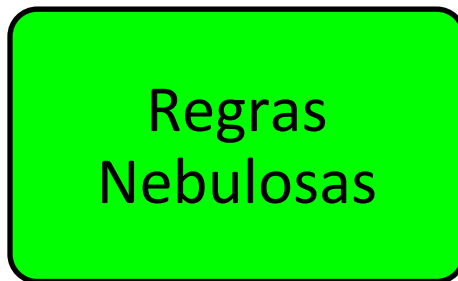
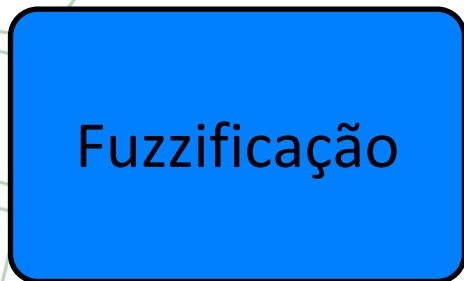


Temperatura da água

14 °C



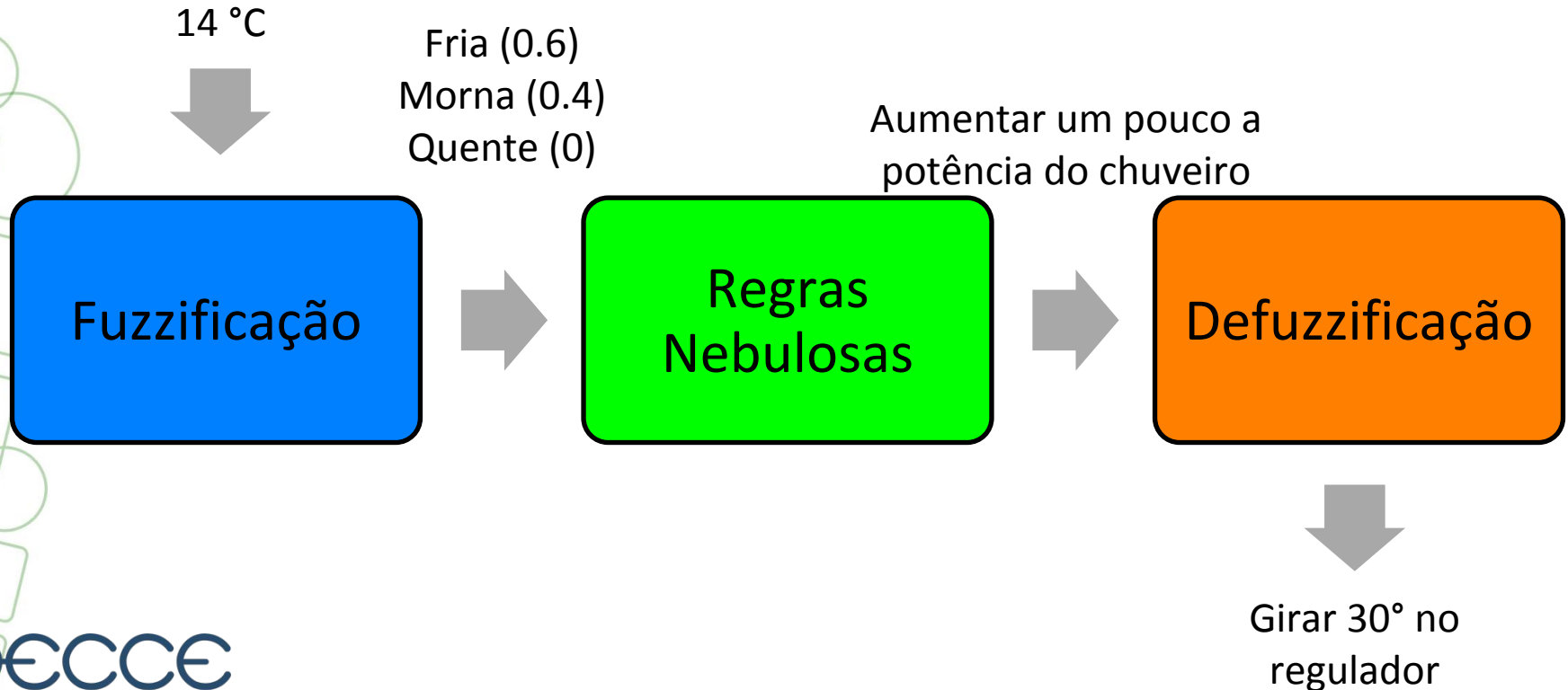
Fria (0.6)
Morna (0.4)
Quente (0)



Regras Nebulosas

- Se a água estiver **MUITO fria**:
 - **Aumentar MUITO** a potência do chuveiro
- Se a água estiver **fria**:
 - **Aumentar um POUCO** a potência do chuveiro
- Se a água estiver **morna**:
 - **Não efetuar ação**
- Se a água estiver **quente**:
 - **Diminuir um POUCO** a potência do chuveiro
- Se a água estiver **MUITO quente**:
 - **Diminuir MUITO** a potência do chuveiro

Temperatura da água



Sumário

- Introdução
- Exemplo do Chuveiro
- Comparação com Lógica Booleana
- Processos da Lógica Fuzzy
- **Implementação em Python**
- Outras Aplicações

Importando os Módulos

```
!pip install networkx==1.9.0 --force-reinstall
```

```
!pip install scikit-fuzzy
```

```
import numpy as np
```

```
import skfuzzy as fuzz
```

```
from skfuzzy import control as ctrl
```

Importando os Módulos

```
!pip install networkx==1.9.0 --force-reinstall
```

```
!pip install scikit-fuzzy
```

```
import numpy as np
```

```
import skfuzzy as fuzz
```

```
from skfuzzy import control as ctrl
```



Definindo as Variáveis Fuzzy

```
flow = ctrl.Antecedent(np.arange(0, 20.1, 0.1), 'flow')
```

```
temperature = ctrl.Antecedent(np.arange(10, 50.1, 0.1), 'temperature')
```

```
water = ctrl.Consequent(np.arange(0, 10.1, 0.1), 'water')
```

Definindo as Variáveis Fuzzy

```
flow = ctrl.Antecedent(np.arange(0, 20.1, 0.1), 'flow')
```

```
temperature = ctrl.Antecedent(np.arange(10, 50.1, 0.1), 'temperature')
```

```
water = ctrl.Consequent(np.arange(0, 10.1, 0.1), 'water')
```

`ctrl.antecedent` → Entrada Fuzzy

`ctrl.consequent` → Saída Fuzzy

Definindo as Variáveis Fuzzy

```
flow = ctrl.Antecedent(np.arange(0, 20.1, 0.1), 'flow')
```

```
temperature = ctrl.Antecedent(np.arange(10, 50.1, 0.1), 'temperature')
```

```
water = ctrl.Consequent(np.arange(0, 10.1, 0.1), 'water')
```

`np.arange(0, 12, 2) = [0, 2, 4, 6, 8, 10]`

Início do intervalo

Final do intervalo

Espaçamento entre valores

Definindo as Variáveis Fuzzy

```
flow = ctrl.Antecedent(np.arange(0, 20.1, 0.1), 'flow')
```

```
temperature = ctrl.Antecedent(np.arange(10, 50.1, 0.1), 'temperature')
```

```
water = ctrl.Consequent(np.arange(0, 10.1, 0.1), 'water')
```

Variável	Vazão (l/min)	Temperatura (°C)	Água (nota)
Mínimo	0	10	0
Máximo	20	50	10

Definindo os Conjuntos Fuzzy

```
flow['low'] = fuzz.trimf(flow.universe, [0, 0, 10])
```

```
flow['medium'] = fuzz.trimf(flow.universe, [0, 10, 20])
```

```
flow['high'] = fuzz.trimf(flow.universe, [10, 20, 20])
```

Definindo os Conjuntos Fuzzy

```
flow['low'] = fuzz.trimf(flow.universe, [0, 0, 10])
```

```
flow['medium'] = fuzz.trimf(flow.universe, [0, 10, 20])
```

```
flow['high'] = fuzz.trimf(flow.universe, [10, 20, 20])
```

fuzz.trimf → Criar um Conjunto Fuzzy triangular



Definindo os Conjuntos Fuzzy

`flow['low'] = fuzz.trimf(flow.universe, [0, 0, 10])`

`flow['medium'] = fuzz.trimf(flow.universe, [0, 10, 20])`

`flow['high'] = fuzz.trimf(flow.universe, [10, 20, 20])`

Vazão (l/min)	Valor Mínimo	Valor Máximo	Valor Mínimo
Baixa	0	0	10
Média	0	10	20
Alta	10	20	20

Definindo os Conjuntos Fuzzy

temperature['cold'] = fuzz.trapmf(temperature.universe, [10, 10, 15, 22.5])

temperature['normal'] = fuzz.trapmf(temperature.universe, [15, 22.5, 27.5, 35])

temperature['hot'] = fuzz.trapmf(temperature.universe, [27.5, 35, 50, 50])

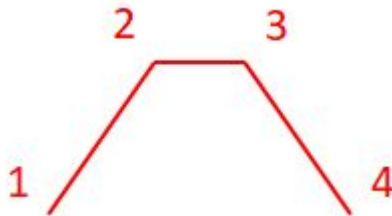
Definindo os Conjuntos Fuzzy

```
temperature['cold'] = fuzz.trapmf(temperature.universe, [10, 10, 15, 22.5])
```

```
temperature['normal'] = fuzz.trapmf(temperature.universe, [15, 22.5, 27.5, 35])
```

```
temperature['hot'] = fuzz.trapmf(temperature.universe, [27.5, 35, 50, 50])
```

fuzz.trapmf → Criar um Conjunto Fuzzy trapezoidal



Definindo os Conjuntos Fuzzy

temperature['cold'] = fuzz.trapmf(temperature.universe, [10, 10, 15, 22.5])

temperature['normal'] = fuzz.trapmf(temperature.universe, [15, 22.5, 27.5, 35])

temperature['hot'] = fuzz.trapmf(temperature.universe, [27.5, 35, 50, 50])

Temperatura (°C)				
	Valor Mínimo	Valor Máximo	Valor Máximo	Valor Mínimo
Fria	10	10	15	22,5
Normal	15	22,5	27,5	35
Quente	27,5	35	50	50

Definindo os Conjuntos Fuzzy

`water['Bad'] = fuzz.gaussmf(water.universe, 0, 1.5)`

`water['Good'] = fuzz.gaussmf(water.universe, 5, 1.5)`

`water['Great'] = fuzz.gaussmf(water.universe, 10, 1.5)`

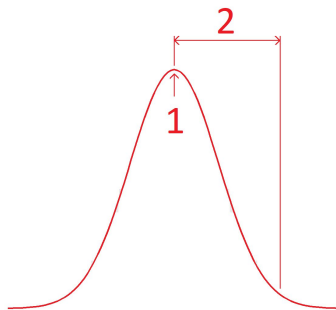
Definindo os Conjuntos Fuzzy

```
water['Bad'] = fuzz.gaussmf(water.universe, 0, 1.5)
```

```
water['Good'] = fuzz.gaussmf(water.universe, 5, 1.5)
```

```
water['Great'] = fuzz.gaussmf(water.universe, 10, 1.5)
```

fuzz.gaussmf → Criar um Conjunto Fuzzy gaussiano



Definindo os Conjuntos Fuzzy

`water['Bad'] = fuzz.gaussmf(water.universe, 0, 1.5)`

`water['Good'] = fuzz.gaussmf(water.universe, 5, 1.5)`

`water['Great'] = fuzz.gaussmf(water.universe, 10, 1.5)`

Água (nota)	Média	Sigma
Ruim	0	1,5
Boa	5	1,5
Ótimo	10	1,5

Definindo os Conjuntos Fuzzy

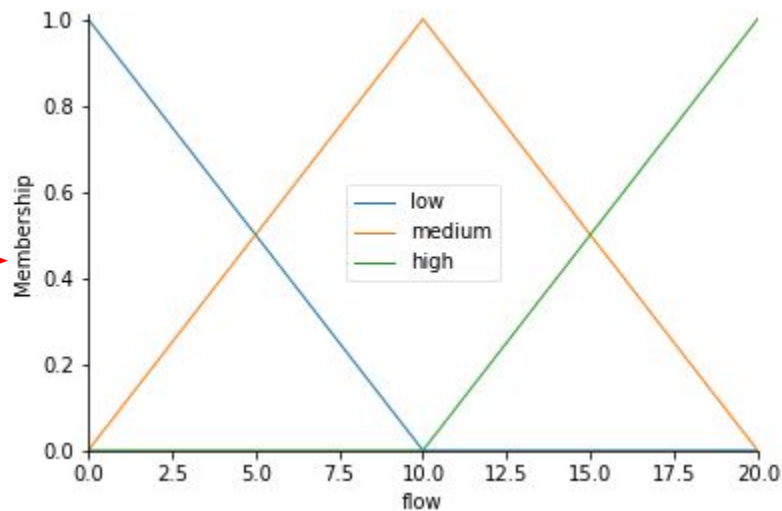
flow.view()

temperature.view()

water.view()

Definindo os Conjuntos Fuzzy

flow.view()
temperature.view()
water.view()

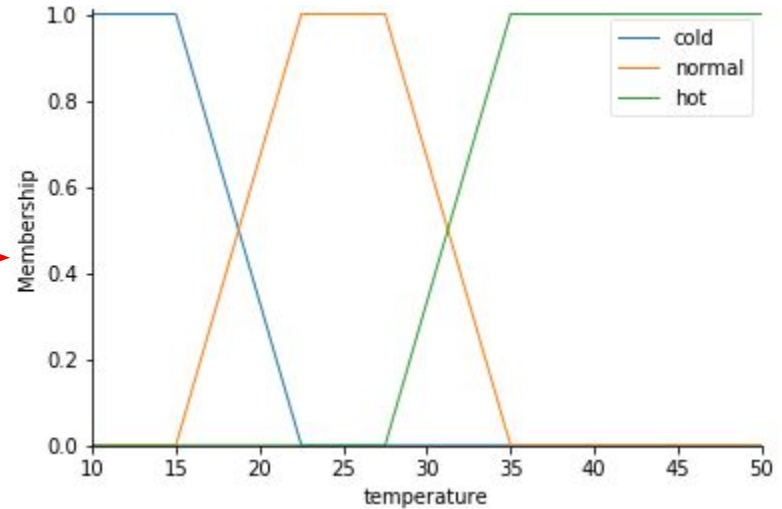


Definindo os Conjuntos Fuzzy

flow.view()

temperature.view()

water.view()

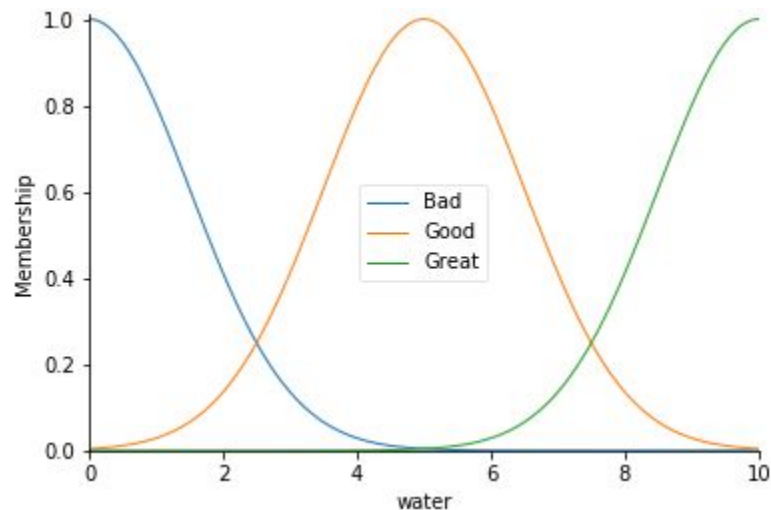


Definindo os Conjuntos Fuzzy

flow.view()

temperature.view()

water.view()



Definindo as Regras Nebulosas

```
rule1 = ctrl.Rule(flow['low'] & temperature['cold'], water['Bad'])
rule2 = ctrl.Rule(flow['low'] & temperature['normal'], water['Good'])
rule3 = ctrl.Rule(flow['low'] & temperature['hot'], water['Bad'])
rule4 = ctrl.Rule(flow['medium'] & temperature['cold'], water['Good'])
rule5 = ctrl.Rule(flow['medium'] & temperature['normal'], water['Great'])
rule6 = ctrl.Rule(flow['medium'] & temperature['hot'], water['Good'])
rule7 = ctrl.Rule(flow['high'] & temperature['cold'], water['Bad'])
rule8 = ctrl.Rule(flow['high'] & temperature['normal'], water['Good']) # modo ECO
rule9 = ctrl.Rule(flow['high'] & temperature['hot'], water['Bad'])
```

Definindo as Regras Nebulosas

Variável	Vazão	Temperatura	Água
Regra 1	Baixa	Fria	Ruim
Regra 2	Baixa	Normal	Boa
Regra 3	Baixa	Quente	Ruim
Regra 4	Média	Fria	Boa
Regra 5	Média	Normal	Ótima
Regra 6	Média	Quente	Boa
Regra 7	Alta	Fria	Ruim
Regra 8	Alta	Normal	Boa
Regra 9	Alta	Quente	Ruim

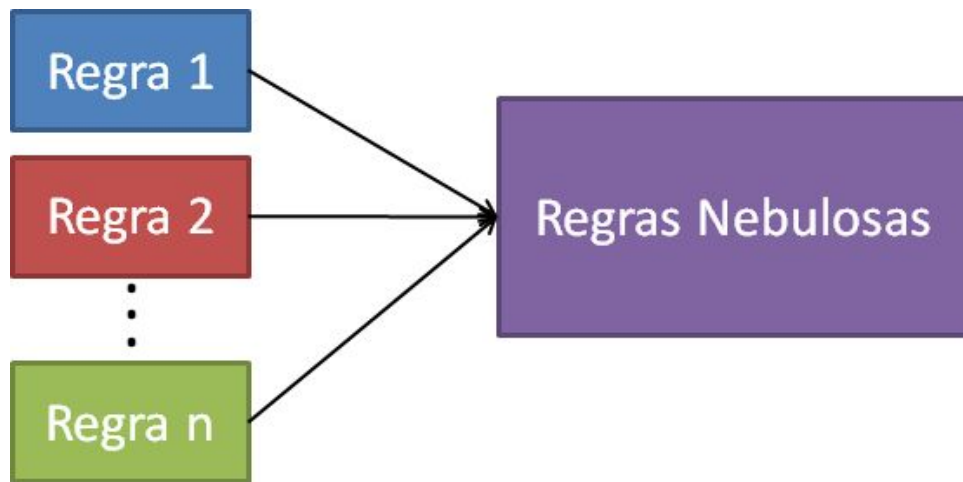
Definindo as Regras Nebulosas

```
shower_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8,  
rule9])  
shower = ctrl.ControlSystemSimulation(shower_ctrl)
```

Definindo as Regras Nebulosas

```
shower_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8,  
rule9])
```

```
shower = ctrl.ControlSystemSimulation(shower_ctrl)
```



Testando o Sistema Fuzzy

```
shower.input['flow'] = 8  
shower.input['temperature'] = 22  
shower.compute()  
print(shower.output['water'])
```


Testando o Sistema Fuzzy

```
shower.input['flow'] = 8  
shower.input['temperature'] = 22  
shower.compute()  
print(shower.output['water'])
```

$f(8 \text{ l/min} , 22 \text{ }^{\circ}\text{C}) = \text{nota } 7$

Testando o Sistema Fuzzy

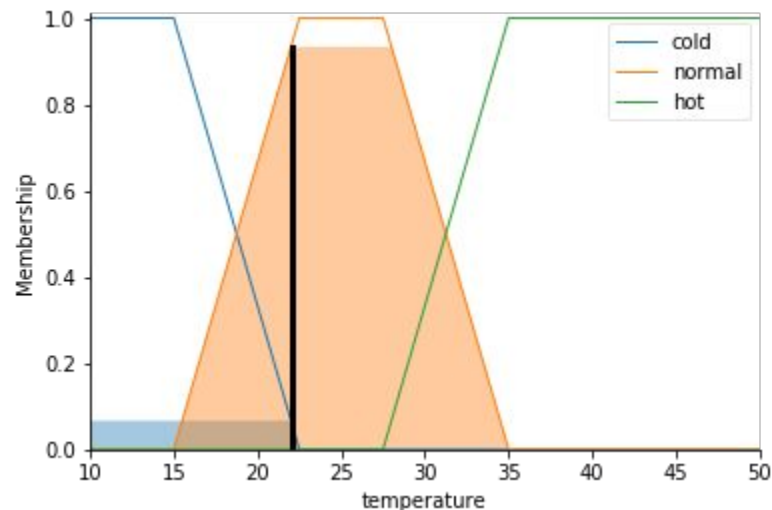
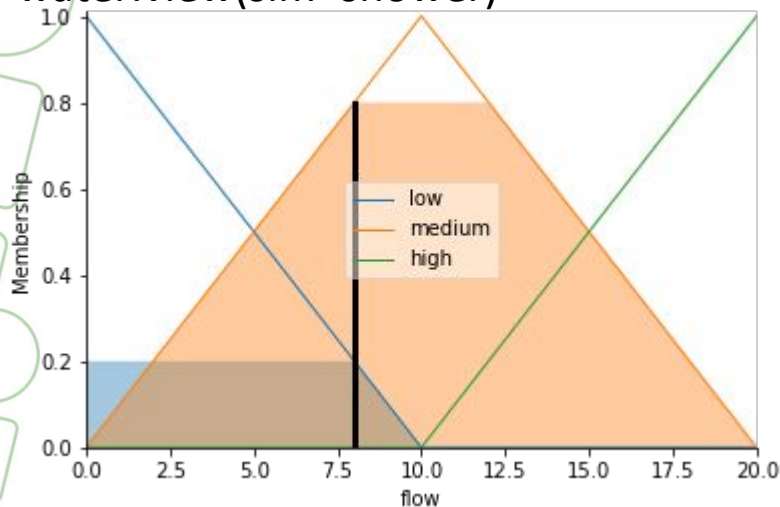
flow.view(sim=shower)
temperature.view(sim=shower)
water.view(sim=shower)

Testando o Sistema Fuzzy

flow.view(sim=shower)

temperature.view(sim=shower)

water.view(sim=shower)

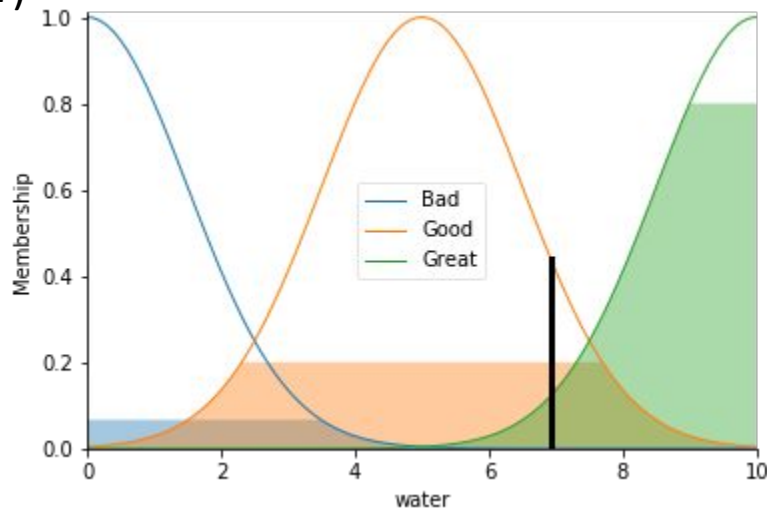


Testando o Sistema Fuzzy

flow.view(sim=shower)

temperature.view(sim=shower)

water.view(sim=shower)



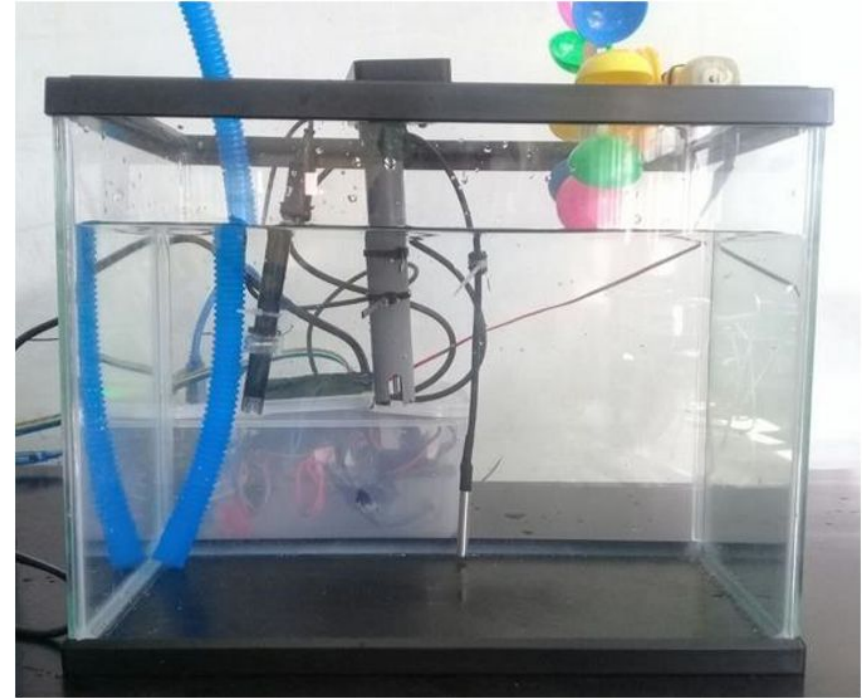
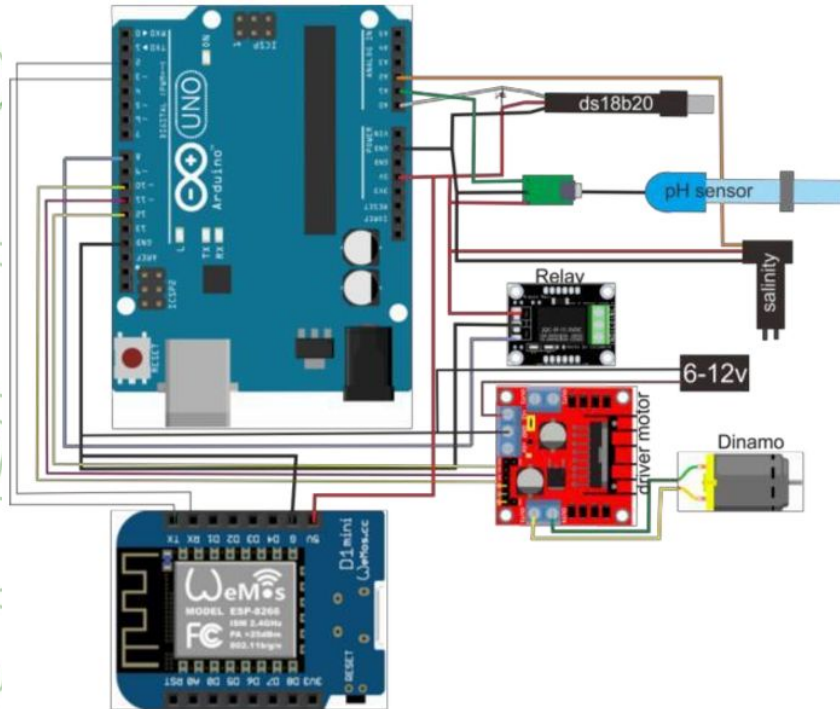
Sumário

- Introdução
- Exemplo do Chuveiro
- Comparação com Lógica Booleana
- Processos da Lógica Fuzzy
- Implementação em Python
- Outras Aplicações

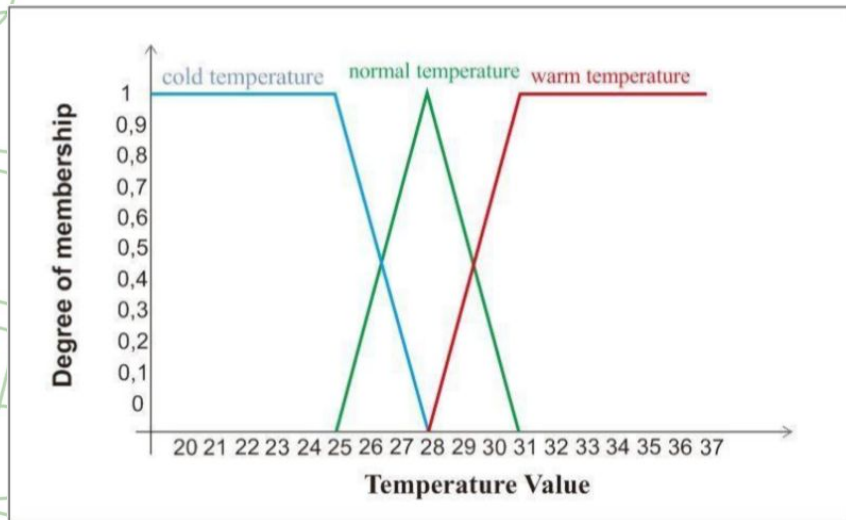
Controle em Tanque para criação de Camarões

- Wardhany, V. A; et al. **Fuzzy Logic Based Control System Temperature, pH and Water Salinity on Vanammei Shrimp Ponds.** IEEE: 2018. Disponível em:
<<https://ieeexplore.ieee.org/document/8615464>>. Acesso em 08 jul 2019.

Controle em Tanque para criação de Camarões



Controle em Tanque para criação de Camarões



Rule Number	Pin Spool Rule		
	<i>Temperature</i>	<i>pH</i>	<i>Pin Spool</i>
Rule 0	$\leq 24^{\circ}\text{C}$	6,5 – 8,5	ON
Rule 1	$25^{\circ} - 31^{\circ}\text{C}$	6,5 – 8,5	OFF
Rule 2	$\geq 32^{\circ}\text{C}$	6,5 – 8,5	ON
Rule 3	$\leq 24^{\circ}\text{C}$	$< 6,5$	ON
Rule 4	$25^{\circ} - 31^{\circ}\text{C}$	$< 6,5$	ON
Rule 5	$\geq 32^{\circ}\text{C}$	$< 6,5$	ON

Controle de Tráfego

- Patil, R; Srinivasaraghavan, A. **Smart traffic controller using fuzzy inference system(STCFIS)**. IEEE: 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7877437>>. Acesso em 08 jul 2019.

Controle de Tráfego



Controle de Tráfego

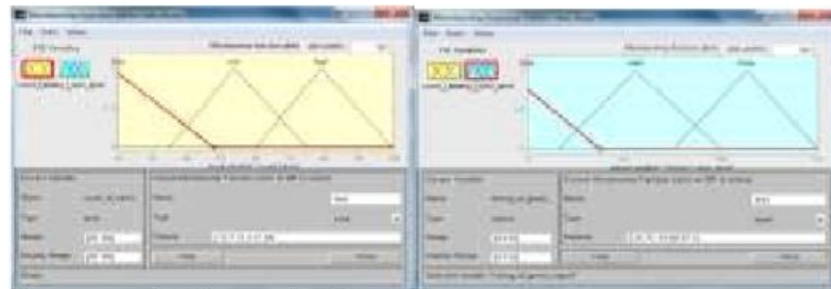
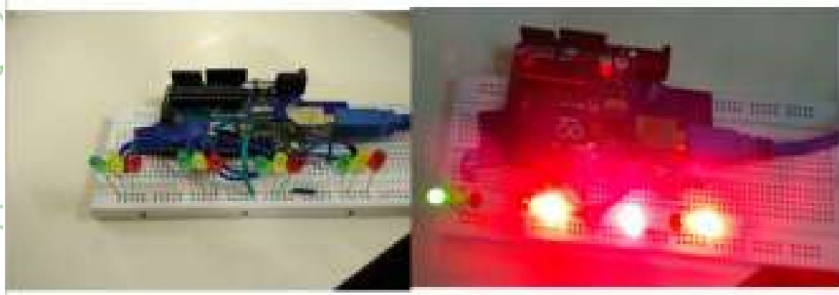


Fig.11. Membership Function for Input and Output

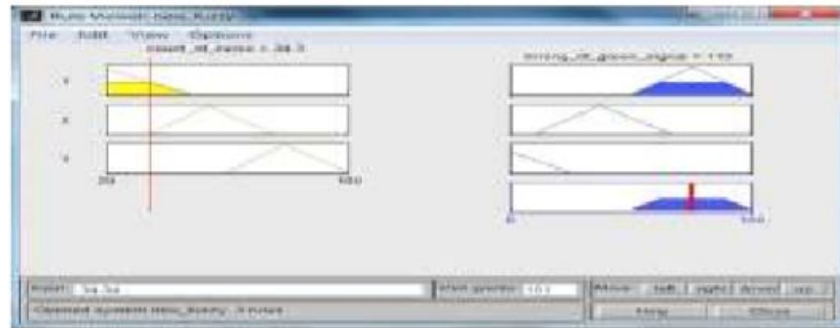


Fig.12. Fuzzy Rules

Lógica Fuzzy com Python

Email: rafael.noboro@gmail.com

<http://sites.florianopolis.ifsc.edu.br/pecce/>

Projeto do Edital 02/2018/PROPPI/PROEX Câmpus Florianópolis : Programa de Integração da Pesquisa e Extensão ao Ensino do Câmpus Florianópolis