

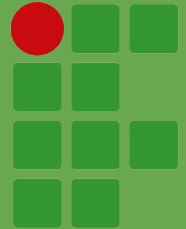
Lógica Fuzzy

Disciplina: Inteligência Artificial

Docente: Leila Roling Scariot da Silva

Discente: Gustavo Neri Pereira Dias

Discente: Paulo César Costa da Silva



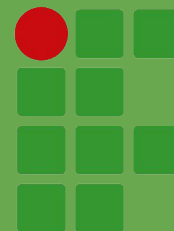
**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

Roteiro apresentação

- 1) Apresentar problema
- 2) Lógica Fuzzy
- 3) Relacionar o problema com a lógica fuzzy
- 3) Fuzzificação
- 4) Inferência Fuzzy
- 5) Método de visualização da solução
- 6) Resultado
- 7) Código



**INSTITUTO
FEDERAL**
Goiano

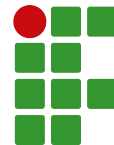
Campus
Morrinhos

Devido a complexidade da conservação de insulinas este trabalho propõe o uso da solução fuzzy para esse problema.

As insulinas apresentam boa estabilidade e tem sua ação biológica preservada, desde que sejam seguidas as orientações de conservação e transporte (GROSSI, 2004; SBD, 2009).

Quando mantida em geladeira, a insulina deve ser retirada com 30 minutos de antecedência da aplicação, para evitar desconforto e irritação no local.

O local mais adequado para armazenar a insulina, na geladeira, é a prateleira próxima à gaveta de legumes, pois as prateleiras e gavetas próximas ao freezer, podem expôr a insulina à temperaturas inferiores a 2° C, ocasionando congelamento e perda de efeito. A porta da geladeira também não é indicada para seu armazenamento, já que, as frequentes aberturas de porta causam grande mobilidade no frasco e variação da temperatura da insulina, podendo danificá-la.



**INSTITUTO
FEDERAL**

Goiano

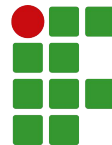
Campus
Morrinhos

A Lógica Fuzzy (Nebulosa ou Difusa), é uma extensão da lógica booleana e uma lógica multivalorada capaz de capturar informações vagas (linguagem natural), e convertê-las para um formato numérico.

- Grau de pertinência é um valor qualquer no intervalo de $[0, 1]$
- Determina o quanto um determinado elemento pertence a um conjunto, permitindo uma transição gradual da falsidade para a verdade.
- Desse modo, é possível avaliar conceitos não quantificáveis.

Exemplos:

- Avaliar a temperatura (quente, morno, frio, ...)
- Sentimento de felicidade (radiante, feliz, apático, triste, ...)



**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

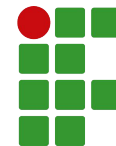
Conservação de insulinas

As insulinas em uso, podem ser mantidas na geladeira (entre 2 e 8°C), ou em temperatura ambiente (15 e 30°C), em local fresco, ao abrigo da luz e de oscilações bruscas de temperatura.

Os fabricantes não recomendam guardar a caneta recarregável em geladeira, porque pode danificar o mecanismo interno e, em alguns casos, interferir no registro da dose correta; devendo então, ser mantida em temperatura ambiente. Já as canetas descartáveis, podem ser armazenadas tanto em geladeira, quanto em temperatura ambiente.

Quando mantida em geladeira, a insulina deve ser retirada com 30 minutos de antecedência da aplicação, para evitar desconforto e irritação no local.

As insulinas lacradas (aquelas que ainda não estão em uso), necessitam ser armazenadas em geladeira (temperatura entre 2 e 8°C) e a data de validade a ser considerada, é a que está impressa na embalagem.



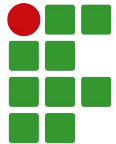
**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

Relacionar com o Fuzzy

Insulina	Temperatura	Validade
Insulina lacrada	2° a 8° C	2 a 3 anos, impressa na embalagem
Insulina em uso	2° a 8°C (geladeira) até 30° C (ambiente)	4 a 6 semanas após a data de abertura e início do uso



**INSTITUTO
FEDERAL**
Goiano

Campus
Morrinhos

Engloba:

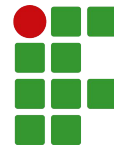
- Análise do Problema
- Definição das Variáveis
- Definição das Funções de pertinência (Triangular, Trapezoidal, Gaussiana)
- Criação das Regiões

Variáveis de Entrada:

- **Temperatura:**
 - Universo: $[0, 30]$
 - Funções de Pertinência: fria, amena, quente
- **Umidade:**
 - Universo: $[0, 100]$
 - Funções de Pertinência: baixa, média, alta

Variáveis de Saída:

- **Validade:**
 - Universo: $[0, 42]$
 - Funções de Pertinência: curta, aceitável, longa



**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

Fuzzificação

```
# [2] Fuzzificação
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

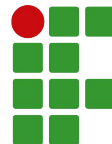
# Criando as variáveis do problema
temperatura = ctrl.Antecedent(np.arange(0, 31, 1), 'temperatura')
umidade = ctrl.Antecedent(np.arange(0, 101, 1), 'umidade')
validade = ctrl.Consequent(np.arange(0, 43, 1), 'validade')

# Criando as funções de pertinência para a temperatura
temperatura['fria'] = fuzz.trapmf(temperatura.universe, [0, 0, 2, 8])
temperatura['amena'] = fuzz.gaussmf(temperatura.universe, 15, 5)
temperatura['quente'] = fuzz.trapmf(temperatura.universe, [20, 25, 30, 30])

# Criando as funções de pertinência para a umidade
umidade['baixa'] = fuzz.trapmf(umidade.universe, [0, 0, 20, 60])
umidade['média'] = fuzz.trapmf(umidade.universe, [0, 50, 60, 100])
umidade['alta'] = fuzz.trimf(umidade.universe, [60, 100, 100])

# Criando as funções de pertinência para validade em uso
validade['curta'] = fuzz.trapmf(validade.universe, [0, 0, 5, 15])
validade['aceitável'] = fuzz.trimf(validade.universe, [10, 22, 35])
validade['longa'] = fuzz.trapmf(validade.universe, [25, 35, 42, 42])

# Visualizando as funções de pertinência para cada variável
temperatura.view()
umidade.view()
validade.view()
```

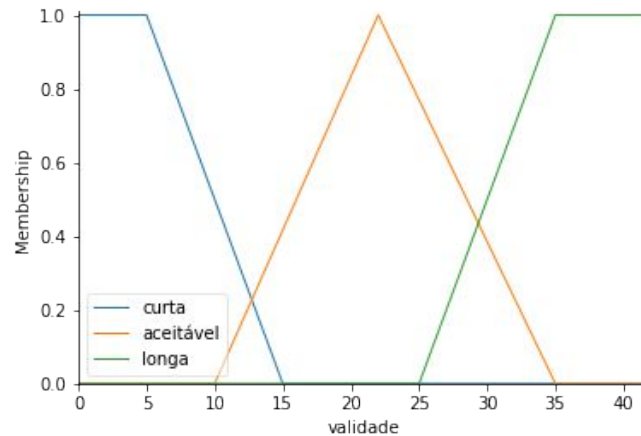
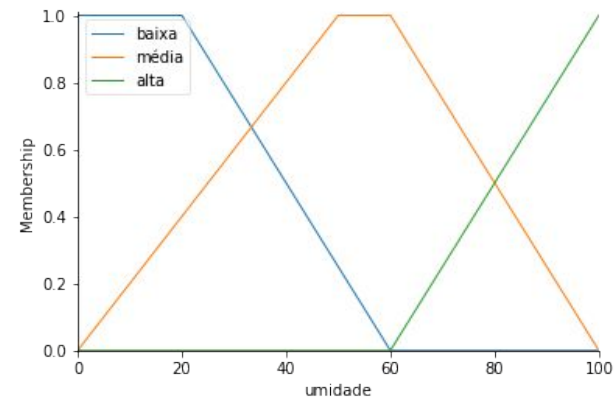
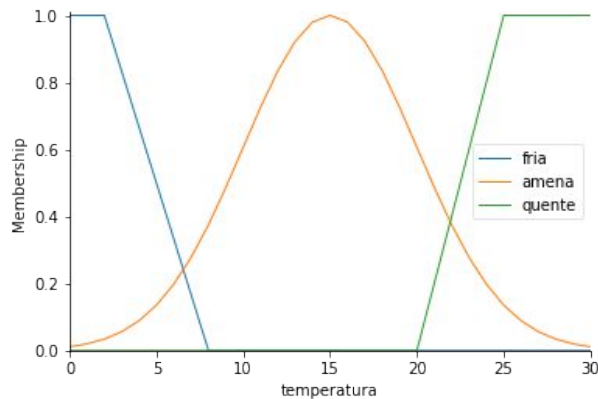


**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

Fuzzificação



Etapa na qual as proposições (regras) são definidas e depois são examinadas paralelamente.

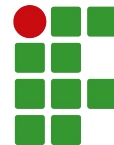
Engloba:

- Definição das proposições
- Análise das regras
- Criação da região resultante

Definição das Proposições para o Sistema Fuzzy de controle da validade.

- Base de Conhecimento/Regras
 - **SE** a temperatura é fria **E** umidade é baixa **ENTÃO** a validade é longa
 - **SE** a temperatura é quente **OU** umidade é alta **ENTÃO** a validade é curta
 - **SE** a umidade é média **ENTÃO** a validade é aceitável

```
# Base de Conhecimento/Regras
rule1 = ctrl.Rule(temperatura['fria'] & umidade['baixa'], validade['longa'])
rule2 = ctrl.Rule(temperatura['quente'] | umidade['alta'], validade['curta'])
rule3 = ctrl.Rule(umidade['média'], validade['aceitável'])
```



**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

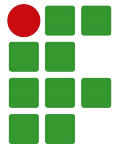
Etapa na qual as regiões resultantes são convertidas em valores para a variável de saída do sistema

```
# Sistema Fuzzy e Simulação
validade_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
validade_simulador = ctrl.ControlSystemSimulation(validade_ctrl)

# Entrada da temperatura
while True:
    temp = float(input('Digite a T (°C): '))
    if(temp<0 or temp >30):
        print('A temperatura deve estar no intervalo [0, 30]')
        continue
    validade_simulador.input['temperatura'] = temp
    break

# Entrada da umidade relativa
while True:
    ur = float(input('Digite a UR (%): '))
    if(ur<0 or temp >100):
        print('A umidade deve estar no intervalo [0, 100]')
        continue
    validade_simulador.input['umidade'] = ur
    break

# Computando o resultado (Inferência Fuzzy + Defuzzificação)
validade_simulador.compute()
print('A validade é de %d dias' % round(validade_simulador.output['validade']))
```



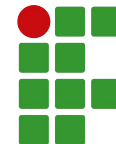
**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

Métodos para a visualização da solução fuzzy:

```
temperatura.view(sim=validade_simulador)  
umidade.view(sim=validade_simulador)  
validade.view(sim=validade_simulador)
```

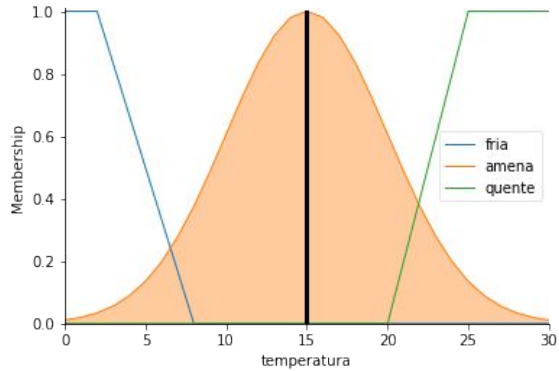


**INSTITUTO
FEDERAL**

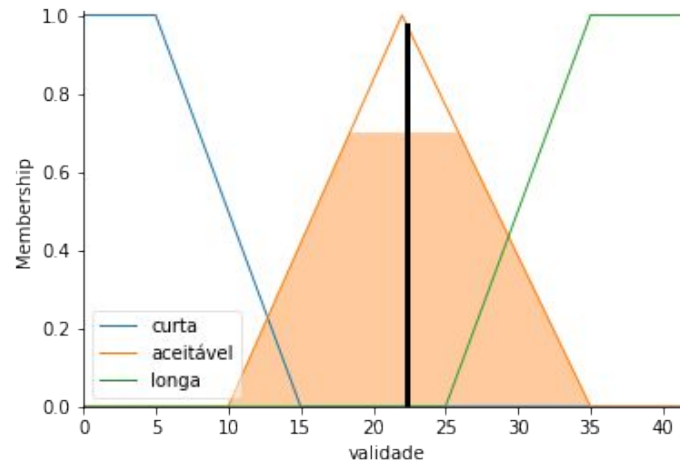
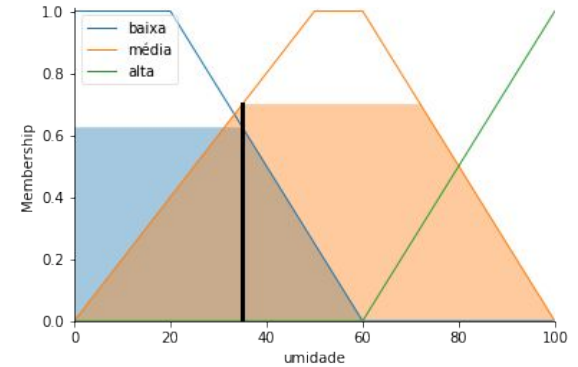
Goiano

Campus
Morrinhos

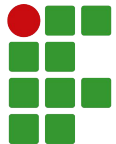
Resultado



```
Digite a T (°C): 32
A temperatura deve estar no intervalo [0, 30]
Digite a T (°C): 15
Digite a UR (%): 35
A validade é de 22 dias
```



```
# [1] Instalação do pacote skfuzzy  
!pip install scikit-fuzzy
```



**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

Código parte 2

```
# [2] Fuzzificação
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

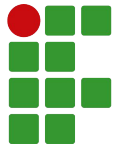
# Criando as variáveis do problema
temperatura = ctrl.Antecedent(np.arange(0, 31, 1), 'temperatura')
umidade = ctrl.Antecedent(np.arange(0, 101, 1), 'umidade')
validade = ctrl.Consequent(np.arange(0, 43, 1), 'validade')

# Criando as funções de pertinência para a temperatura
temperatura['fria'] = fuzz.trapmf(temperatura.universe, [0, 0, 2, 8])
temperatura['amena'] = fuzz.gaussmf(temperatura.universe, 15, 5)
temperatura['quente'] = fuzz.trapmf(temperatura.universe, [20, 25, 30, 30])

# Criando as funções de pertinência para a umidade
umidade['baixa'] = fuzz.trapmf(umidade.universe, [0, 0, 20, 60])
umidade['média'] = fuzz.trapmf(umidade.universe, [0, 50, 60, 100])
umidade['alta'] = fuzz.trimf(umidade.universe, [60, 100, 100])

# Criando as funções de pertinência para validade em uso
validade['curta'] = fuzz.trapmf(validade.universe, [0, 0, 5, 15])
validade['aceitável'] = fuzz.trimf(validade.universe, [10, 22, 35])
validade['longa'] = fuzz.trapmf(validade.universe, [25, 35, 42, 42])

# Visualizando as funções de pertinência para cada variável
temperatura.view()
umidade.view()
validade.view()
```



**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos

```
# [3] Inferência Fuzzy e Defuzzificação

# Base de Conhecimento/Regras
rule1 = ctrl.Rule(temperatura['fria'] & umidade['baixa'], validade['longa'])
rule2 = ctrl.Rule(temperatura['quente'] | umidade['alta'], validade['curta'])
rule3 = ctrl.Rule(umidade['média'], validade['aceitável'])

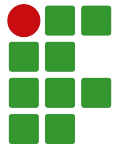
# Sistema Fuzzy e Simulação
validade_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
validade_simulador = ctrl.ControlSystemSimulation(validade_ctrl)

# Entrada da temperatura
while True:
    temp = float(input('Digite a T (°C): '))
    if(temp<0 or temp >30):
        print('A temperatura deve estar no intervalo [0, 30]')
        continue
    validade_simulador.input['temperatura'] = temp
    break

# Entrada da umidade relativa
while True:
    ur = float(input('Digite a UR (%): '))
    if(ur<0 or temp >100):
        print('A umidade deve estar no intervalo [0, 100]')
        continue
    validade_simulador.input['umidade'] = ur
    break

# Computando o resultado (Inferência Fuzzy + Defuzzificação)
validade_simulador.compute()
print('A validade é de %d dias' % round(validade_simulador.output['validade']))

# Visualizando as regiões
temperatura.view(sim=validade_simulador)
umidade.view(sim=validade_simulador)
validade.view(sim=validade_simulador)
```



**INSTITUTO
FEDERAL**

Goiano

Campus
Morrinhos