Relatório sobre Conjuntos, Funções e Operadores Fuzzy

Doutorado CEFET

May 7, 2025

1 Introdução

Este relatório apresenta a implementação e análise de funções de pertinência, fuzzificação, operações fuzzy e relações fuzzy. As funções de pertinência são amplamente utilizadas em lógica fuzzy para representar graus de pertencimento de elementos a conjuntos fuzzy.

2 Funções de Pertinência

2.1 Implementação de Funcões de Pertinência

As funções de pertinência implementadas incluem:

2.1.1 Função Triangular

A função triangular é definida por três parâmetros (a, b, c), onde:

- a é o ponto inicial onde a pertinência começa a aumentar;
- b é o ponto onde a pertinência atinge o valor máximo (1);
- c é o ponto final onde a pertinência retorna a 0.

A fórmula é dada por:

$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, & \text{se } a \leq x < b, \\ \frac{c-x}{c-b}, & \text{se } b \leq x < c, \\ 0, & \text{caso contrário.} \end{cases}$$

O código Python correspondente é:

```
def triangular(x, a, b, c):
    if a <= x < b:
        return (x - a) / (b - a)
    elif b <= x < c:
        return (c - x) / (c - b)
    else:
        return 0
```

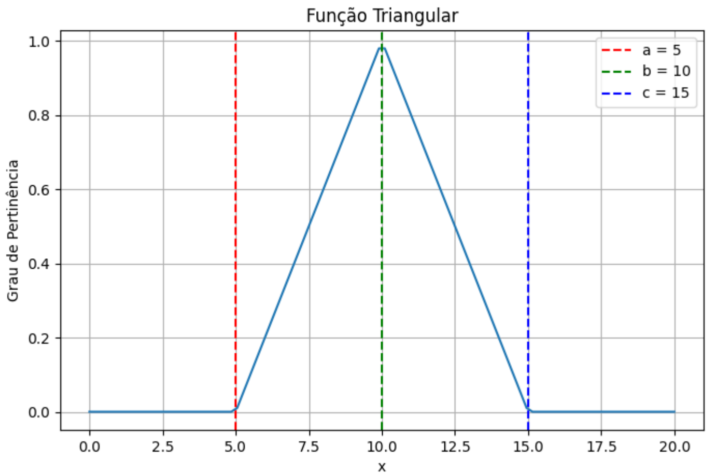

Figure 1: Exemplo de função triangular com $a = 5$, $b = 10$, $c = 15$.

2.1.2 Função Trapezoidal

A função trapezoidal é definida por quatro parâmetros (a, b, c, d), onde:

- a e d são os pontos onde a pertinência é 0;
- b e c definem a região onde a pertinência é 1.

A fórmula é:

$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, & \text{se } a \leq x < b, \\ 1, & \text{se } b \leq x \leq c, \\ \frac{d-x}{d-c}, & \text{se } c < x \leq d, \\ 0, & \text{caso contrário.} \end{cases}$$

O código Python correspondente é:

```
def trapezoidal(x, a, b, c, d):
    if a <= x < b:
```

```
        return (x - a) / (b - a)
    elif b <= x <= c:
        return 1
    elif c < x <= d:
        return (d - x) / (d - c)
    else:
        return 0
```

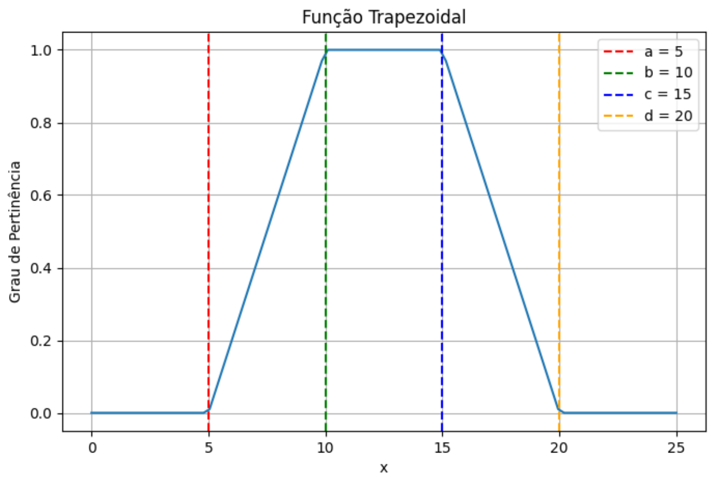

Figure 2: Exemplo de função trapezoidal com $a = 5$, $b = 10$, $c = 15$, $d = 20$.

2.1.3 Função Gaussiana

A função gaussiana é definida por dois parâmetros (c, σ), onde:

- c é o centro da curva, onde a pertinência é máxima (1);
- σ controla a largura da curva.

A fórmula é:

$$\mu(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}.$$

O código Python correspondente é:

```
def gaussian(x, c, sigma):
    return np.exp(-0.5 * ((x - c) / sigma) ** 2)
```

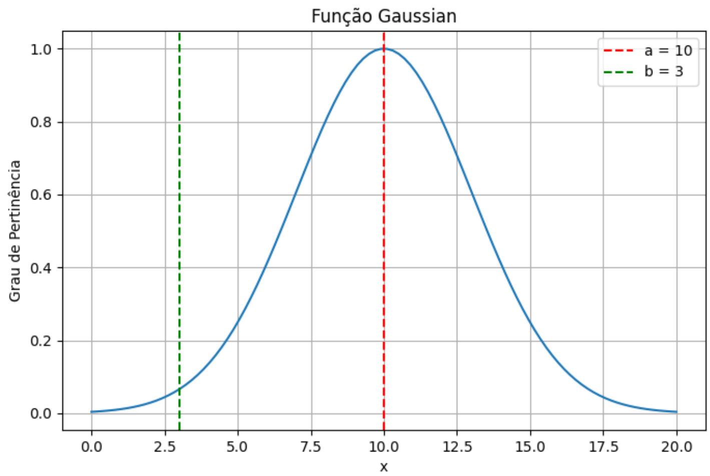

Figure 3: Exemplo de função gaussiana com $c = 10$, $\sigma = 3$.

2.1.4 Função Sigmoidal

A função sigmoidal é definida por dois parâmetros (a, c), onde:

- a controla a inclinação da curva;
- c é o ponto central onde a pertinência é 0.5.

A fórmula é:

$$\mu(x) = \frac{1}{1 + e^{-a(x-c)}}.$$

O código Python correspondente é:

```
def sigmoidal(x, a, c):
    return 1 / (1 + np.exp(-a * (x - c)))
```

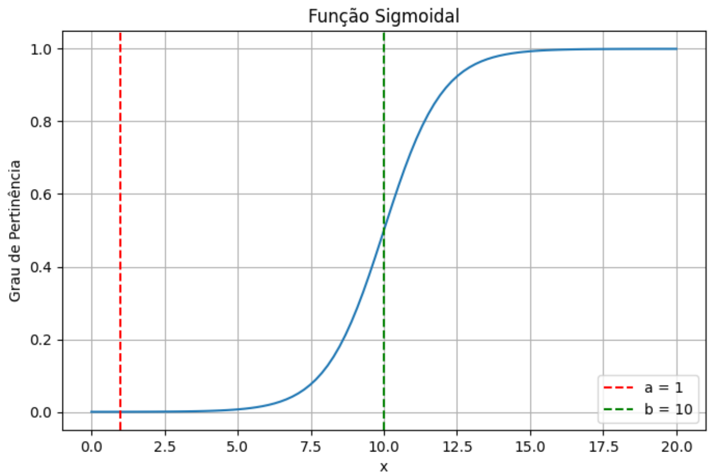

Figure 4: Exemplo de função sigmoidal com $a = 1$, $c = 10$.

2.1.5 Função Sinoidal (Bell)

A função Bell é definida por três parâmetros (a, b, c), onde:

- a controla a largura da curva;
- b controla a inclinação;
- c é o centro da curva.

A fórmula é:

$$\mu(x) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}.$$

O código Python correspondente é:

```
def bell_function(x, a, b, c):
    return 1 / (1 + abs((x - c) / a) ** (2 * b))
```

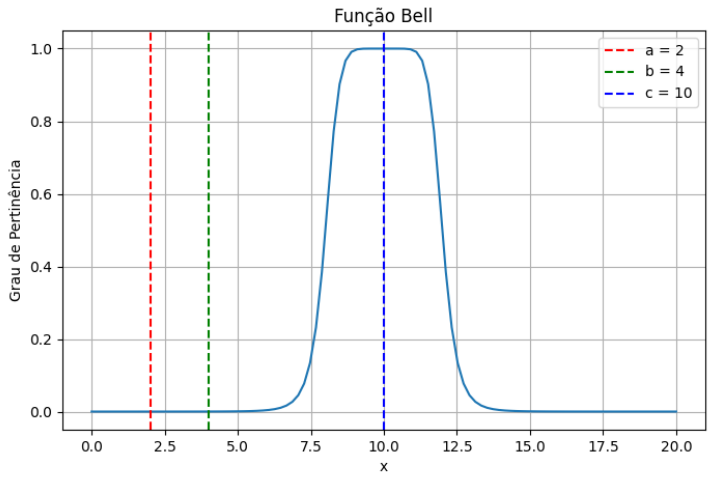

Figure 5: Exemplo de função Bell com $a = 2$, $b = 4$, $c = 10$.

2.1.6 Função S

A função S é definida por dois parâmetros (a, b), onde:

- a é o ponto onde a pertinência começa a aumentar;
- b é o ponto onde a pertinência atinge 1.

A fórmula é:

$$\mu(x) = \begin{cases} 0, & \text{se } x \leq a, \\ 2\left(\frac{x-a}{b-a}\right)^2, & \text{se } a < x < b, \\ 1, & \text{se } x \geq b. \end{cases}$$

O código Python correspondente é:

```
def s_function(x, a, b):
    if x <= a:
        return 0
    elif a < x < b:
        return 2 * ((x - a) / (b - a)) ** 2
    elif x >= b:
        return 1
```

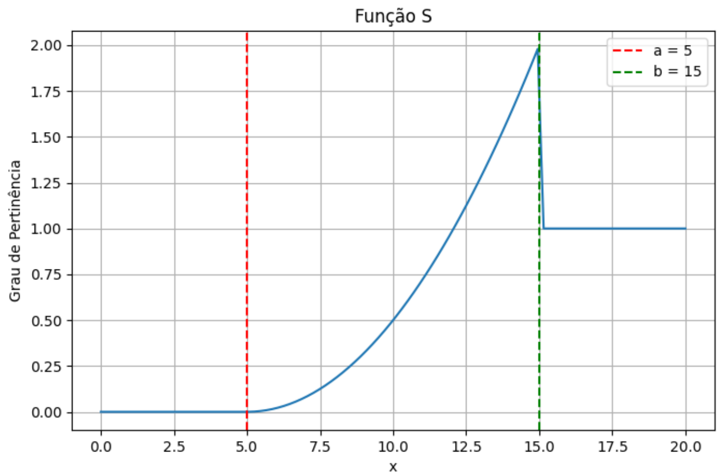

Figure 6: Exemplo de função S com $a = 5$, $b = 15$.

2.1.7 Função Z

A função Z é definida por dois parâmetros (a, b), onde:

- a é o ponto onde a pertinência começa a diminuir;
- b é o ponto onde a pertinência atinge 0.

A fórmula é:

$$\mu(x) = \begin{cases} 1, & \text{se } x \leq a, \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2, & \text{se } a < x < b, \\ 0, & \text{se } x \geq b. \end{cases}$$

O código Python correspondente é:

```
def z_function(x, a, b):
    if x <= a:
        return 1
    elif a < x < b:
        return 1 - 2 * ((x - a) / (b - a)) ** 2
    else:
        return 0
```

2.1.8 Função Cauchy

A função Cauchy é definida por dois parâmetros (c, γ), onde:

- c é o centro da curva;
- γ controla a largura da curva.

A fórmula é:

$$\mu(x) = \frac{1}{1 + \left(\frac{x-c}{\gamma}\right)^2}.$$

O código Python correspondente é:

```
def cauchy_function(x, c, gamma):
    return 1 / (1 + ((x - c) / gamma) ** 2)
```

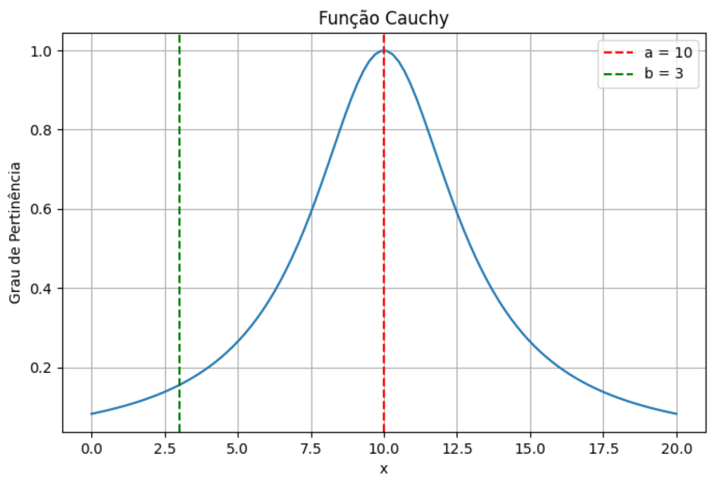

Figure 7: Exemplo de função Cauchy com $c = 10$, $\gamma = 3$.

2.1.9 Função Gaussiana Dupla

A função Gaussiana Dupla é definida por três parâmetros (c, σ_1, σ_2), onde:

- c é o centro da curva;
- σ_1 controla a largura da curva para $x \leq c$;
- σ_2 controla a largura da curva para $x > c$.

A fórmula é:

$$\mu(x) = \begin{cases} e^{-\frac{1}{2}\left(\frac{x-c}{\sigma_1}\right)^2}, & \text{se } x \leq c, \\ e^{-\frac{1}{2}\left(\frac{x-c}{\sigma_2}\right)^2}, & \text{se } x > c. \end{cases}$$

O código Python correspondente é:

```
def double_gaussian(x, c, sigma1, sigma2):
    if x <= c:
        return np.exp(-0.5 * ((x - c) / sigma1) ** 2)
    else:
        return np.exp(-0.5 * ((x - c) / sigma2) ** 2)
```

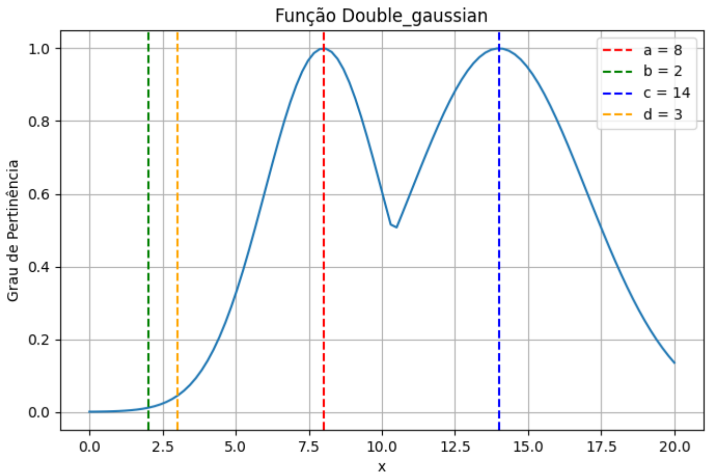

Figure 8: Exemplo de função Gaussiana Dupla com $c = 10$, $\sigma_1 = 3$, $\sigma_2 = 5$.

2.1.10 Função Logarítmica

A função Logarítmica é definida por dois parâmetros (a, b), onde:

- a controla o deslocamento da curva;
- b controla a inclinação da curva.

A fórmula é:

$$\mu(x) = \begin{cases} 0, & \text{se } x \leq a, \\ \log_b(x - a + 1), & \text{se } x > a. \end{cases}$$

O código Python correspondente é:

```
import math

def logarithmic_function(x, a, b):
    if x <= a:
        return 0
    else:
        return math.log(x - a + 1, b)
```

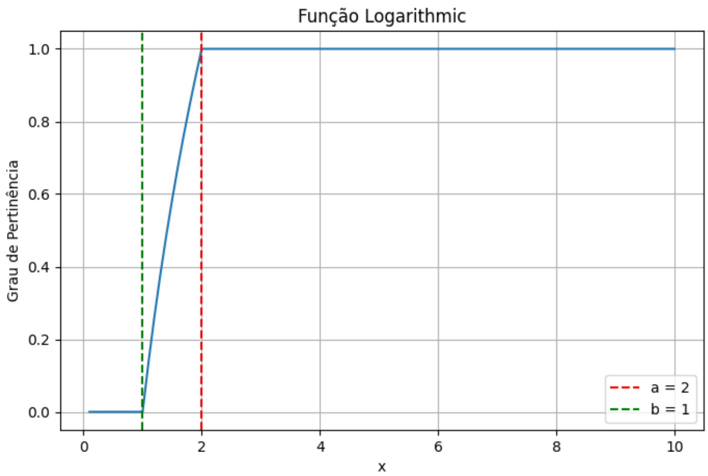

Figure 9: Exemplo de função Logarítmica com $a = 5$, $b = 2$.

2.1.11 Função Retangular

A função Retangular é definida por dois parâmetros (a, b), onde:

- a é o início do intervalo onde a pertinência é 1;
- b é o final do intervalo onde a pertinência é 1.

A fórmula é:

$$\mu(x) = \begin{cases} 1, & \text{se } a \leq x \leq b, \\ 0, & \text{caso contrário.} \end{cases}$$

O código Python correspondente é:

```
def rectangular(x, a, b):
    if a <= x <= b:
        return 1
    else:
        return 0
```

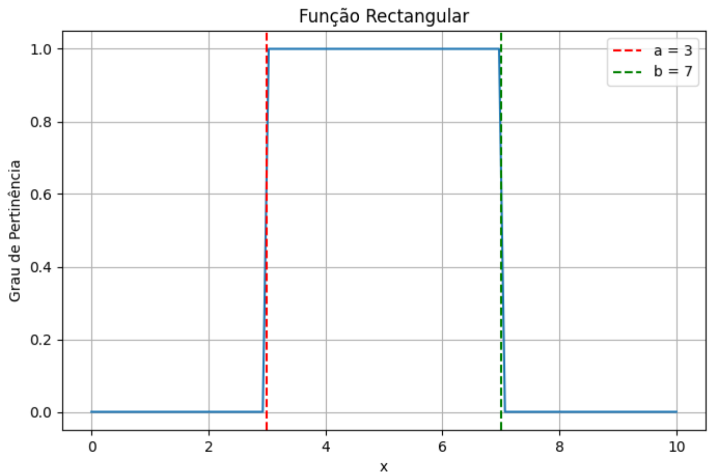

Figure 10: Exemplo de função Retangular com $a = 10$, $b = 20$.

2.2 Fuzzificação e Análise Comparativa

Para a fuzzificação, escolhemos uma variável de entrada com universo de discurso definido e particionamos o domínio em funções de pertinência uniformemente espaçadas. A seguir, apresentamos os resultados para duas amostras distintas.

Este relatório apresenta a definição de funções de pertinência fuzzy, a fuzzificação de amostras e a análise gráfica e textual dos resultados. As funções de pertinência analisadas incluem as funções triangular, trapezoidal, gaussiana e sigmoidal.

3 Definição do Universo de Discurso

A variável de entrada escolhida é a **temperatura ambiente**, com o universo de discurso definido como $[0, 40]$ graus Celsius.

4 Particionamento do Domínio

O domínio foi particionado em quatro funções de pertinência uniformemente espaçadas para cada tipo de função. Os parâmetros utilizados são descritos abaixo:

4.1 Função Triangular

$$\mu_{\text{triangular}}(x; a, b, c) = \begin{cases} 0, & x \leq a \text{ ou } x \geq c, \\ \frac{x-a}{b-a}, & a < x \leq b, \\ \frac{c-x}{c-b}, & b < x < c. \end{cases}$$

Parâmetros: $a = 0$, $b = 10$, $c = 20$; $a = 10$, $b = 20$, $c = 30$; $a = 20$, $b = 30$, $c = 40$.

4.2 Função Trapezoidal

$$\mu_{\text{trapezoidal}}(x; a, b, c, d) = \begin{cases} 0, & x \leq a \text{ ou } x \geq d, \\ \frac{x-a}{b-a}, & a < x \leq b, \\ 1, & b < x \leq c, \\ \frac{d-x}{d-c}, & c < x < d. \end{cases}$$

Parâmetros: $a = 0$, $b = 5$, $c = 15$, $d = 20$; $a = 10$, $b = 15$, $c = 25$, $d = 30$; $a = 20$, $b = 25$, $c = 35$, $d = 40$.

4.3 Função Gaussiana

$$\mu_{\text{gaussiana}}(x; c, \sigma) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right)$$

Parâmetros: $c = 10$, $\sigma = 5$; $c = 20$, $\sigma = 5$; $c = 30$, $\sigma = 5$.

4.4 Função Sigmoidal

$$\mu_{\text{sigmoidal}}(x; a, c) = \frac{1}{1 + e^{-a(x-c)}}$$

Parâmetros: $a = 1$, $c = 10$; $a = 1$, $c = 20$; $a = 1$, $c = 30$.

5 Fuzzificação de Amostras

As amostras escolhidas para fuzzificação foram $x = 15$ e $x = 25$. Os graus de ativação para cada função de pertinência foram calculados e apresentados nos gráficos a seguir.

6 Resultados Gráficos

Os gráficos abaixo mostram o universo de discurso, as funções de pertinência e os graus de ativação das amostras.

6.1 Função Triangular

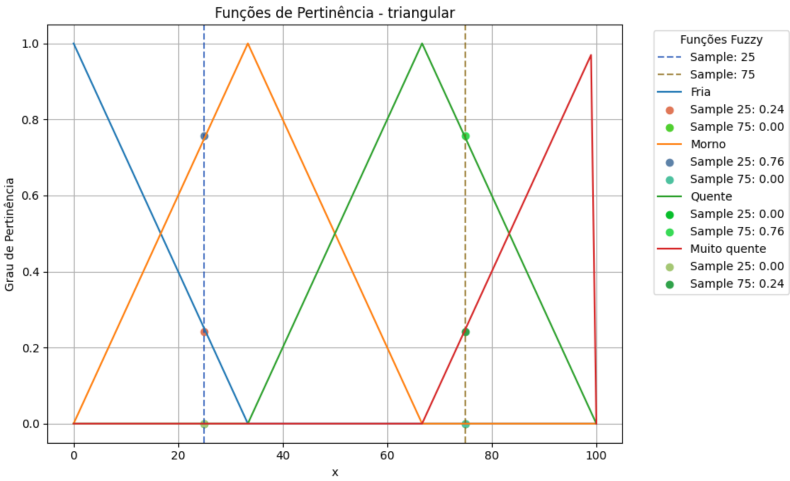

Figure 11: Funções de pertinência triangulares e graus de ativação.

6.2 Função Trapezoidal

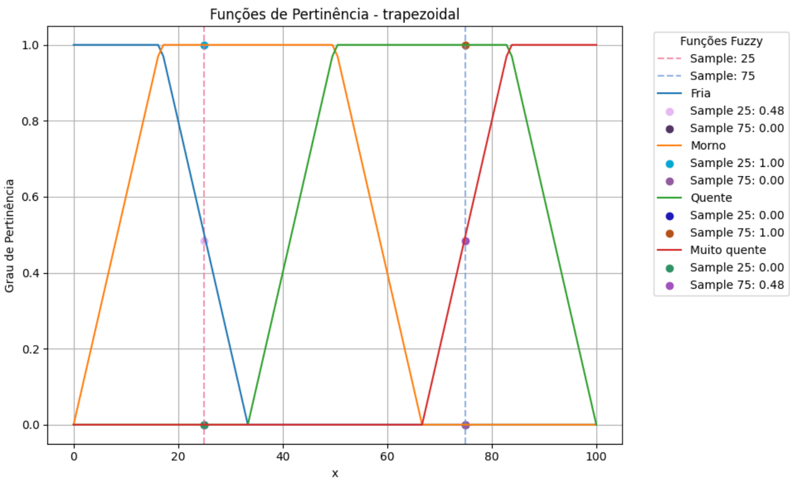

Figure 12: Funções de pertinência trapezoidais e graus de ativação.

6.3 Função Gaussiana

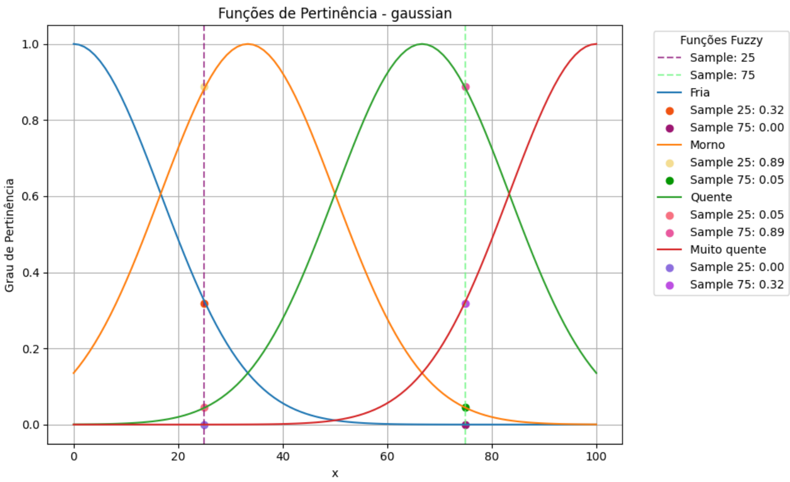

Figure 13: Funções de pertinência gaussianas e graus de ativação.

6.4 Função Sigmoidal

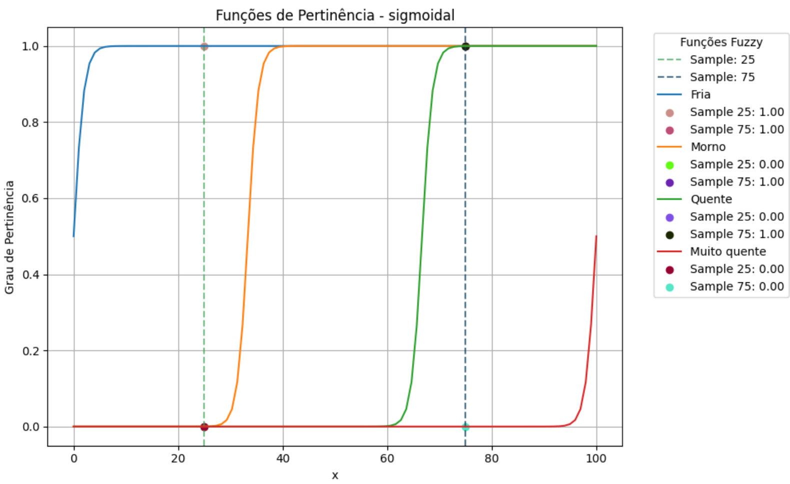

Figure 14: Funções de pertinência sigmoidais e graus de ativação.

7 Análise Comparativa

- As funções triangulares e trapezoidais apresentam transições mais abruptas entre os graus de pertinência, enquanto as funções gaussianas e sigmoidais possuem transições mais suaves.

- As funções gaussianas são mais sensíveis a variações próximas ao centro, enquanto as funções sigmoidais apresentam maior suavidade em todo o domínio.
- As funções trapezoidais são úteis para representar intervalos com pertinência máxima constante, enquanto as funções triangulares são mais adequadas para transições lineares.

8 Conclusão

Este relatório apresentou a definição, particionamento e análise de funções de pertinência para a variável temperatura ambiente. A análise gráfica e textual destacou as diferenças entre os tipos de funções, evidenciando suas características e aplicações.