Centro Federal de Educação Tecnológica de Minas Gerais
Professor: Alisson Marques da Silva   Disciplina: I.C
Aluno: Celso Vinícius Sudário   Matrícula: 20203003611

Divinópolis, 04 de Novembro de 2024

Atividade 01 - Conjuntos, Funções e Operadores Fuzzy

Problema 1 : Implemente uma função para calcular a pertinência de um valor em uma função de pertinência: 1. Triangular, 2. Trapezoidal, 3. Gaussiana, 4. Sigmoidal, 5. Sino, 6. Função-S, 7. Função-Z, 8. Cauchy, 9. Gaussiana Dupla, 10. e 11. Definidas pelo usuário.

Resposta:

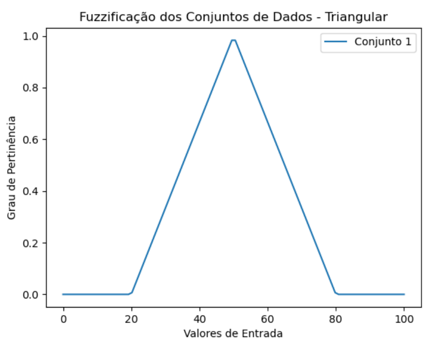
Na Etapa 1, o domínio é definido como uma sequência de 100 pontos uniformemente espaçados entre 0 e 100, representando o universo de discurso para as funções de pertinência fuzzy. Esse domínio busca avaliar os graus de pertinência de cada valor, fornecendo uma visão geral das diferentes funções de pertinência aplicadas.

1.1 Triangular

A função Triangular [Fig. 1] é calculada pela interpolação linear, crescendo de zero para o ponto médio b e decrescendo após ele. A fórmula para x entre a e b é $\frac{x-a}{b-a}$ e entre b e c é $\frac{c-x}{c-b}$.

```python
# 1. Triangular
def triangular_mf(x, a, b, c):
    """
    Função de pertinência triangular.
    x: valor de entrada
    a: limite inferior (Mi = 0)
    b: centro (Mi = 1)
    c: limite superior (Mi = 0)
    Retorna o grau de pertinência Mi.
    """
    if x <= a or x >= c:
        return 0.0
    elif a < x <= b:
        return (x - a) / (b - a)
    else:
        return (c - x) / (c - b)
```

(a)

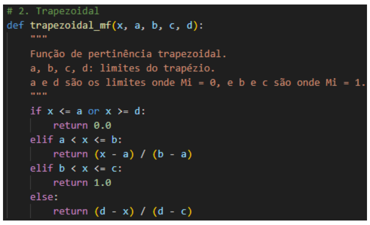


(b)

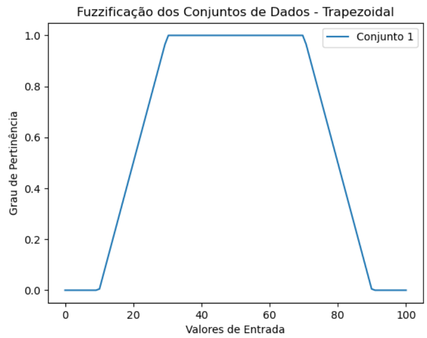
Figura 1: Representação da função `Triangular`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

1.2 Trapezoidal

A função Trapezoidal [Fig. 2] também utiliza a interpolação linear, mas possui uma região plana onde a pertinência é 1, entre os pontos b e c. A fórmula para x entre a e b é $\frac{x-a}{b-a}$, entre b e c é 1, e entre c e d é $\frac{d-x}{d-c}$.



(a)

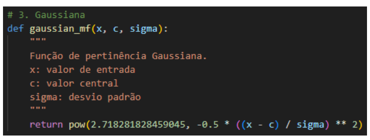


(b)

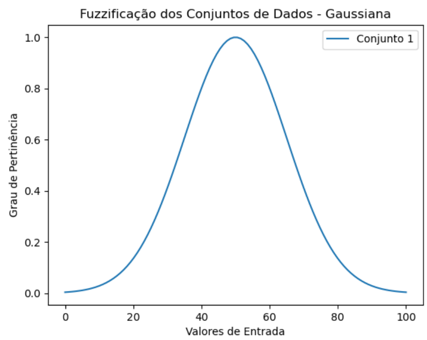
Figura 2: Representação da função `Trapezoidal`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

1.3 Gaussiana

A função Gaussiana [Fig. 3] utiliza a fórmula da curva normal: $e^{-0.5\cdot\left(\frac{x-c}{\sigma}\right)^2}$, onde c é o centro e σ controla a largura da curva. Essa fórmula gera uma curva simétrica em torno de c.



(a)



(b)

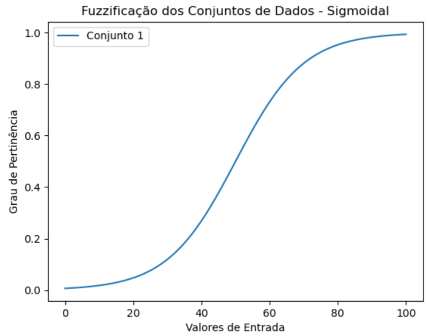
Figura 3: Representação da função `Gaussiana`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

1.4 Sigmoidal

A função Sigmoidal [Fig. 4] é calculada por $\frac{1}{1+e^{-\alpha(x-c)}}$, onde α controla a inclinação da curva e c é o ponto de inflexão, determinando onde a pertinência é 0.5.

```python
# 4. Sigmoidal
def sigmoidal_mf(x, alpha, c):
    """
    Função de pertinência Sigmoidal.
    x: valor de entrada
    alpha: inclinação da curva
    c: ponto central
    """
    return 1 / (1 + pow(2.718281828459045, -alpha * (x - c)))
```

(a)



(b)

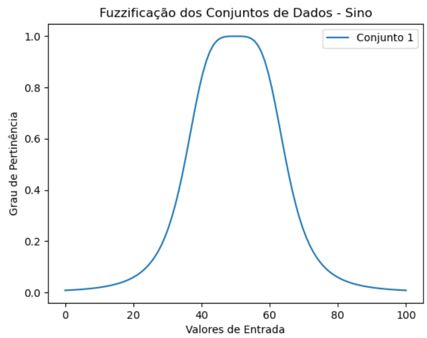
Figura 4: Representação da função `Sigmoidal`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

1.5 Sino

A função Sino [Fig. 5] utiliza a fórmula $\frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}$, onde a controla a largura, b a inclinação, e c o centro. Esse formato é uma versão ajustável da curva gaussiana.

```python
# 5. Sino
def bell_mf(x, a, b, c):
    """
    Função de pertinência Sino.
    a: largura da curva
    b: inclinação
    c: valor central
    """
    return 1 / (1 + abs((x - c) / a) ** (2 * b))
```

(a)



(b)

Figura 5: Representação da função `Sino`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

1.6 S (S-Shaped)

A função S (S-Shaped) [Fig. 6] começa em zero e cresce suavemente até 1. Para x entre a e $(a+b)/2$, a fórmula é $2\left(\frac{x-a}{b-a}\right)^2$, enquanto para x entre $(a+b)/2$ e b, é $1-2\left(\frac{b-x}{b-a}\right)^2$.

```python
# 6. Função-S
def s_shaped_mf(x, a, b):
    """
    Função de pertinência em 'S' (S-shaped).
    x: valor de entrada
    a: limite inferior (Mi = 0)
    b: limite superior (Mi = 1)
    """
    if x <= a:
        return 0.0
    elif a < x <= (a + b) / 2:
        return 2 * ((x - a) / (b - a)) ** 2
    elif (a + b) / 2 < x <= b:
        return 1 - 2 * ((b - x) / (b - a)) ** 2
    else:
        return 1.0
```

(a)

(b)

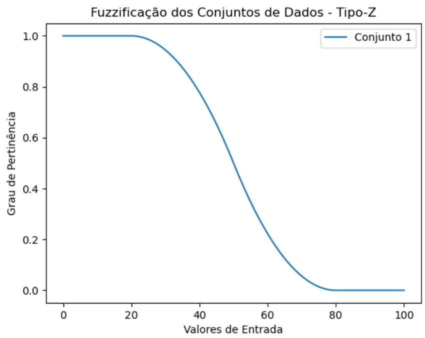
Figura 6: Representação da função `S-Shaped`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

1.7 Z (Z-shaped)

A função Z (Z-shaped) [Fig. 7] é a inversa da S, decrescendo suavemente de 1 para zero. Para x entre a e $(a+b)/2$, a fórmula é $1-2\left(\frac{x-a}{b-a}\right)^2$, enquanto para x entre $(a+b)/2$ e b, é $2\left(\frac{b-x}{b-a}\right)^2$.

```python
# 7. Função-Z
def z_shaped_mf(x, a, b):
    """
    Função de pertinência em 'Z' (Z-shaped).
    x: valor de entrada
    a: limite inferior (Mi = 1)
    b: limite superior (Mi = 0)
    """
    if x <= a:
        return 1.0
    elif a < x <= (a + b) / 2:
        return 1 - 2 * ((x - a) / (b - a)) ** 2
    elif (a + b) / 2 < x <= b:
        return 2 * ((b - x) / (b - a)) ** 2
    else:
        return 0.0
```

(a)

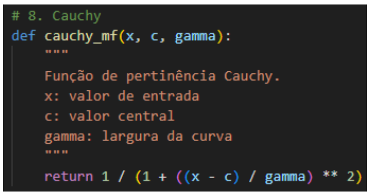


(b)

Figura 7: Representação da função `Z-shaped`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

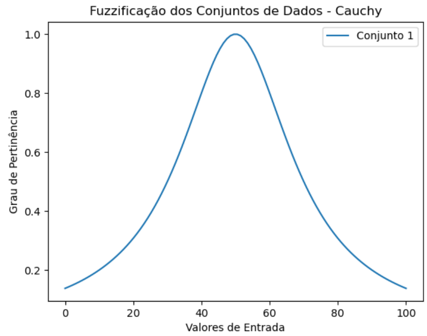
1.8 Cauchy

A função de Cauchy [Fig. 8] é calculada por $\frac{1}{1+\left(\frac{x-c}{\gamma}\right)^2}$, onde c é o centro e γ controla a largura, gerando uma curva semelhante à gaussiana, mas com caudas mais longas.



```python
# 8. Cauchy
def cauchy_mf(x, c, gamma):
    """
    Função de pertinência Cauchy.
    x: valor de entrada
    c: valor central
    gamma: largura da curva
    """
    return 1 / (1 + ((x - c) / gamma) ** 2)
```

(a)

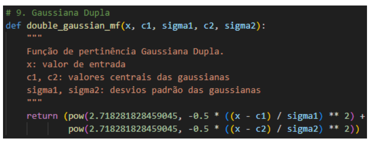


(b)

Figura 8: Representação da função `Cauchy`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

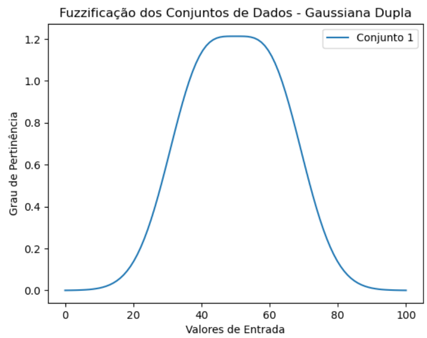
1.9 Gaussiana Dupla

A função Gaussiana Dupla [Fig. 9] combina duas gaussianas para gerar duas regiões de máxima pertinência. Sua fórmula é $e^{-0.5\cdot\left(\frac{x-c_1}{\sigma_1}\right)^2} + e^{-0.5\cdot\left(\frac{x-c_2}{\sigma_2}\right)^2}$, onde c_1 e c_2 são os centros e σ_1 e σ_2 as larguras.



```python
# 9. Gaussiana Dupla
def double_gaussian_mf(x, c1, sigma1, c2, sigma2):
    """
    Função de pertinência Gaussiana Dupla.
    x: valor de entrada
    c1, c2: valores centrais das gaussianas
    sigma1, sigma2: desvios padrão das gaussianas
    """
    return (pow(2.718281828459045, -0.5 * ((x - c1) / sigma1) ** 2) +
            pow(2.718281828459045, -0.5 * ((x - c2) / sigma2) ** 2))
```

(a)



(b)

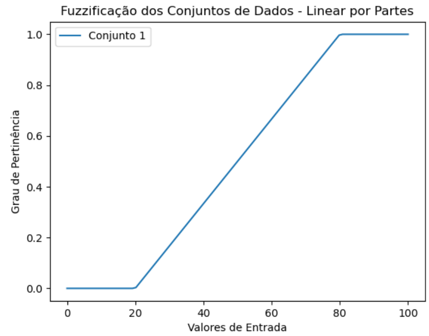
Figura 9: Representação da função `Gaussiana Dupla`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

1.10 Linear por Partes

A função Linear por Partes [Fig. 10] aumenta linearmente entre a e b, com a fórmula $\frac{x-a}{b-a}$. Para valores fora desse intervalo, a pertinência é 0 antes de a e 1 após b.

```python
# 10. Linear por Partes
def linear_piecewise_mf(x, a, b):
    """
    Função Linear por Partes
    a: limite inferior
    b: limite superior
    """
    if x < a:
        return 0.0
    elif a <= x <= b:
        return (x - a) / (b - a)
    else:
        return 1.0
```

(a)



(b)

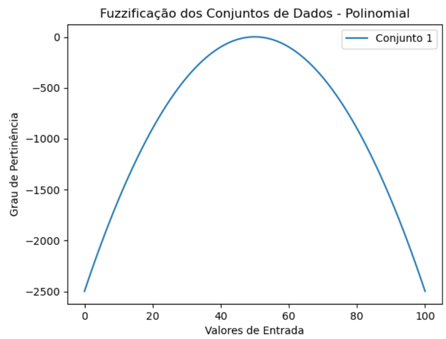
Figura 10: Representação da função `Linear por Partes`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

1.11 Polinomial

A função Polinomial [Fig. 11] calcula a pertinência usando $1 - (x - c)^n$, onde c é o centro e n o expoente, o que procura controlar a curva de decaimento conforme n.

```python
# 11. Polinomial
def polynomial_mf(x, c, n):
    """
    Função Polinomial.
    c: valor central
    n: expoente do polinômio
    """
    return 1 - (x - c) ** n
```

(a)



(b)

Figura 11: Representação da função `Polinomial`: código implementado em Python (a) e seu gráfico correspondente (b), gerado com a biblioteca `Matplotlib`.

Problema 2 : Defina o domínio (universo de discurso) para uma varíavel de entrada a sua escolha. Particione esse universo de discurso em no mínimo 4 funções de pertinência uniformemente espaçadas. Realize a fuzzificação (cálculo do grau de ativação das funções) para duas amostras considerando cada uma das funções apresentadas na atividade anterior.

Resposta:

Para esta etapa, foi definido o domínio de uma variável de entrada no intervalo de 0 a 100, representando o universo de discurso da variável. Este domínio foi particionado em quatro funções de pertinência uniformemente espaçadas para cada uma das seguintes funções: Triangular, Trapezoidal, Gaussiana e Sigmoidal. Em cada função, o grau de pertinência de duas amostras (25 e 75) foi calculado e exibido. Os gráficos a seguir ilustram as sobreposições dessas funções de pertinência para cada tipo de função.

2.1 Função Triangular

A função triangular, conforme exibido no gráfico da Figura 12, possui um formato linear que cresce até um valor máximo e depois decresce simetricamente. A definição do valor de pertinência segue três parâmetros: a (limite inferior), b (centro, onde a pertinência é máxima) e c (limite superior). A função é dada por:

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

Para as amostras 25 e 75, os graus de pertinência encontrados foram:

- Para a amostra 25: Pertinência no Conjunto 1: 0.75; Conjunto 2: 0.25
- Para a amostra 75: Pertinência no Conjunto 3: 0.25; Conjunto 4: 0.75

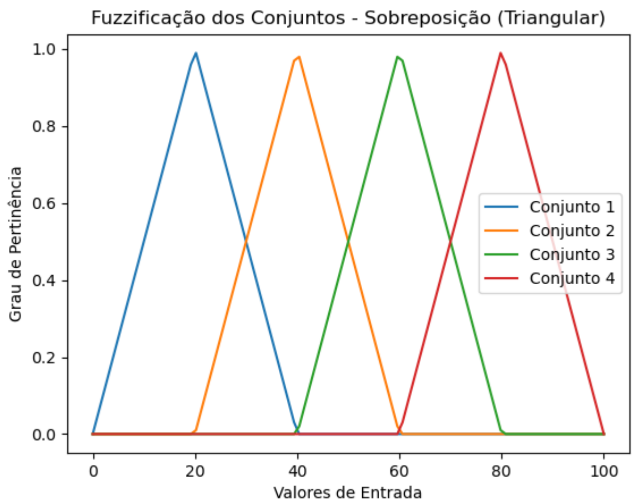


Figura 12: Fuzzificação dos Conjuntos - (Triangular)

A função triangular é eficaz para capturar gradientes de transição em variáveis contínuas e apresenta uma mudança linear entre os limites de baixa e alta pertinência.

2.2 Função Trapezoidal

A função trapezoidal, exibida na Figura 13, tem uma base onde a pertinência é máxima e uma transição gradual nas extremidades. Esta função é caracterizada por quatro parâmetros: a (início da subida), b (início da base), c (final da base) e d (final da descida). Sua fórmula é:

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

Os resultados para as amostras foram:

- Para a amostra 25: Pertinência no Conjunto 1: 1.0; Conjunto 2: 0.33
- Para a amostra 75: Pertinência no Conjunto 3: 0.33; Conjunto 4: 1.0

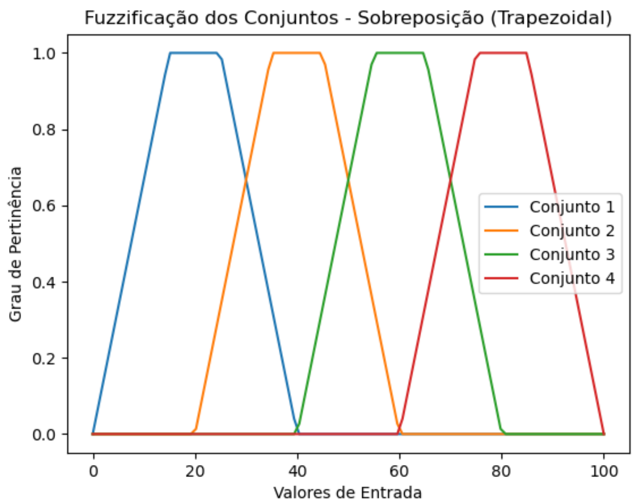


Figura 13: Fuzzificação dos Conjuntos - (Trapezoidal)

A função trapezoidal é ideal para representar variáveis com regiões estáveis e extremidades que variam gradualmente, proporcionando uma transição suave.

2.3 Função Gaussiana

A função gaussiana, ilustrada na Figura 14, possui uma curva suave e simétrica, ideal para transições graduais. É definida pelos parâmetros c (centro da curva) e σ (largura da curva), com a fórmula:

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

Os valores de pertinência obtidos foram:

- Para a amostra 25: Pertinência no Conjunto 1: 0.88; Conjunto 2: 0.32
- Para a amostra 75: Pertinência no Conjunto 3: 0.32; Conjunto 4: 0.88

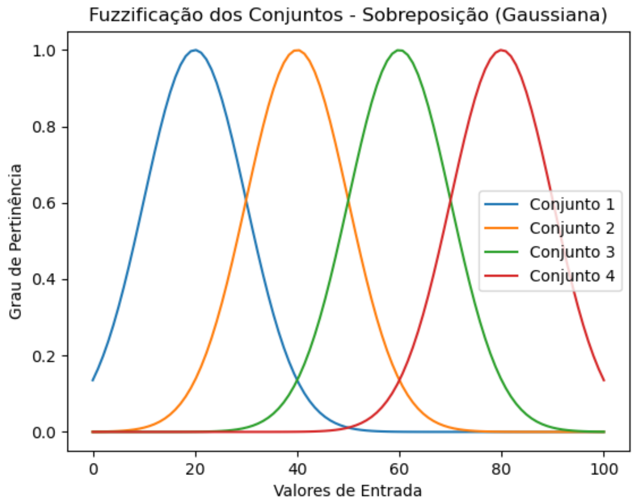


Figura 14: Fuzzificação dos Conjuntos - (Gaussiana)

A gaussiana é apropriada para variáveis onde se deseja uma suavização contínua e simétrica, sendo muito usada em sistemas de controle.

2.4 Função Sigmoidal

A função sigmoidal, mostrada na Figura 15, tem uma curva em "S"que caracteriza uma transição gradual entre pertinência baixa e alta. A função é parametrizada por α (inclinação da curva) e c (ponto central), e é calculada por:

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

Os graus de pertinência para as amostras foram:

- Para a amostra 25: Pertinência no Conjunto 1: 0.62; Conjunto 2: 0.18
- Para a amostra 75: Pertinência no Conjunto 3: 0.82; Conjunto 4: 0.37

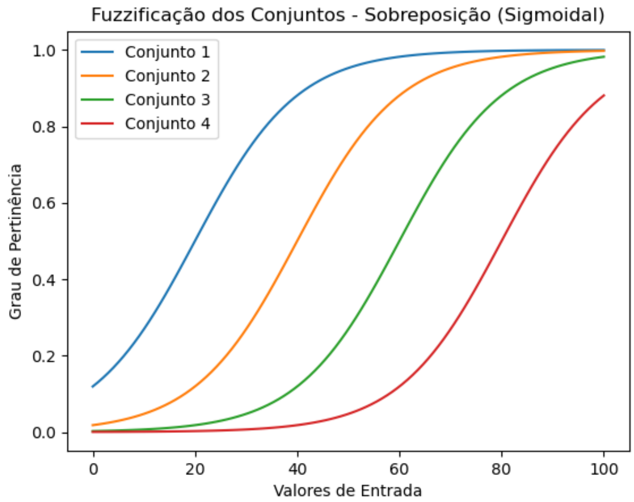


Figura 15: Fuzzificação dos Conjuntos - (Sigmoidal)

A função sigmoidal é útil em contextos onde se deseja uma mudança rápida de pertinência com uma transição controlada, sendo adequada para variáveis que requerem suavidade.

Cada função de pertinência analisada apresentou comportamentos distintos para as amostras 25 e 75. As funções triangular e trapezoidal fornecem transições lineares e são úteis em contextos de variação controlada. A função gaussiana oferece suavidade simétrica, enquanto a função sigmoidal proporciona uma transição acelerada em um intervalo específico. Estes resultados são adequados para aplicações diversas em sistemas fuzzy, onde se deseja modular a suavidade e rapidez da transição entre os conjuntos fuzzy.

Problema 3 : Implemente uma função para realizar operações fuzzy disponíveis nas notas de aula: Complemento, União, Interseção e as Normas Duais. Utilize os conjuntos fuzzy propostos nas atividades anteriores para realizar as operações implementadas.

Resposta:

Nesta etapa, realizou-se operações fuzzy de complemento, união, interseção, T-Norm produto e S-Norm probabilística sobre dois conjuntos fuzzy definidos por funções de pertinência trapezoidais. Para facilitar a análise, foram selecionados pontos específicos do domínio (0, 25, 50, 75 e 100) e comparados os graus de pertinência resultantes em cada uma dessas operações. Abaixo, serão discutidos os resultados de cada operação em termos do impacto nos valores de pertinência, com referência às figuras correspondentes.

3.1 Complemento

O complemento inverte o grau de pertinência do conjunto em relação ao seu limite máximo, resultando em $1-\mu(x)$ para cada ponto x. Como mostra a Figura 16, observou-se que, nos pontos 0, 75 e 100, o grau de pertinência é máximo, com valor 1.00, indicando que esses pontos têm o valor mais baixo de pertinência nos conjuntos originais. No entanto, no ponto 25, onde o grau de pertinência era 1.00 nos conjuntos originais, o complemento reverte o valor para 0.00. No ponto intermediário (50), o complemento resulta em um valor de 0.97, próximo de 1, o que indica que a pertinência original nesse ponto era baixa, mas não nula. O complemento destaca, assim, os valores que eram originalmente mínimos nos conjuntos fuzzy, invertendo a intensidade das regiões de relevância.

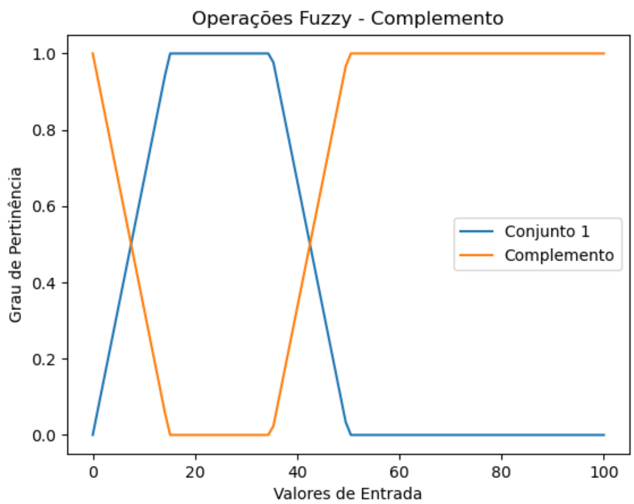


Figura 16: Operação de Complemento aplicada aos conjuntos fuzzy.

3.2 União

A união dos conjuntos fuzzy maximiza o grau de pertinência entre os conjuntos em cada ponto, ou seja, $\max(\mu_A(x), \mu_B(x))$. Como resultado, nos pontos 25 e 50, onde ambos os conjuntos possuem pertinência alta, a união atinge o valor máximo de 1.00. Nos pontos extremos (0 e 100), o grau de pertinência é 0.00, o que pode refletr na ausência de relevância nessas regiões em ambos os conjuntos. No ponto 75, onde há uma sobreposição menor entre os conjuntos, o grau de pertinência reduz-se para 0.35. Esse comportamento, ilustrado na Figura 17, demonstra que a união preserva as áreas de alta pertinência e permite a expansão da influência de cada conjunto nos pontos de interseção.

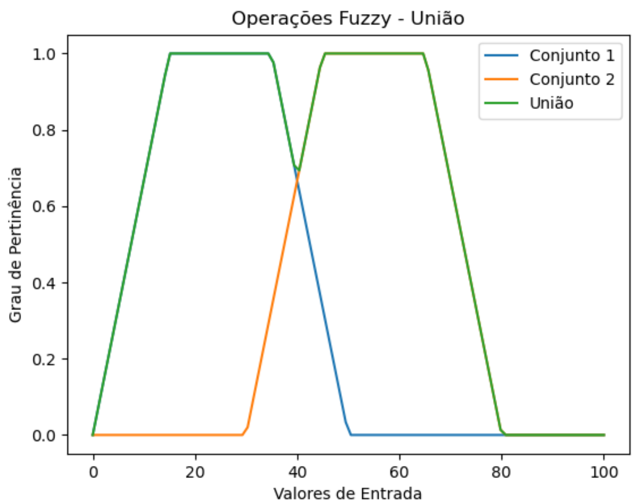


Figura 17: Operação de União entre os conjuntos fuzzy.

3.3 Interseção

A interseção dos conjuntos fuzzy calcula o menor grau de pertinência entre os conjuntos, isto é, $\min(\mu_A(x), \mu_B(x))$. Como esperado, a interseção resulta em 0.00 nos pontos 0, 25, 75 e 100, indicando que pelo menos um dos conjuntos não possui relevância nessas regiões. No ponto intermediário 50, a interseção alcança um valor baixo de 0.03, o que pode sugerir uma mínima sobreposição entre os conjuntos nesse ponto específico. A Figura 18 mostra que a operação de interseção evidencia as áreas onde ambos os conjuntos possuem baixa pertinência, destacando as regiões de interseção mínima.

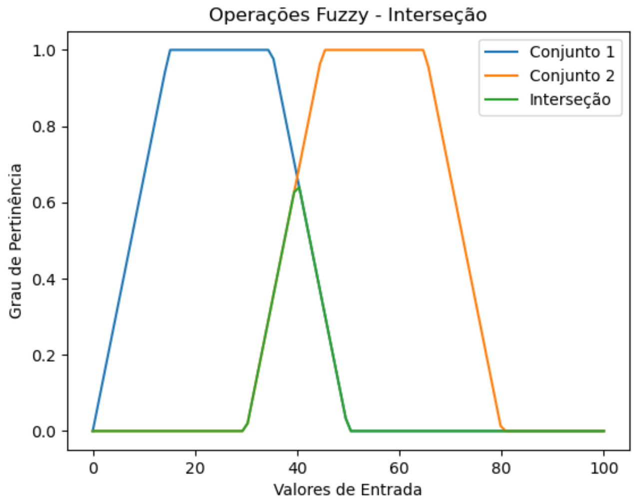


Figura 18: Operação de Interseção entre os conjuntos fuzzy.

3.4 T-Norm: Produto

A T-Norm produto multiplica os graus de pertinência dos conjuntos em cada ponto, $\mu_A(x) \cdot \mu_B(x)$. Devido à natureza multiplicativa, os resultados são ainda menores do que na interseção, com valores de 0.00 nos pontos 0, 25, 75 e 100, e um valor de 0.03 no ponto 50. Esse comportamento, como ilustrado na Figura 19, destaca que a T-Norm produto é mais restritiva, acentuando ainda mais as áreas de baixa pertinência mútua. Portanto, essa operação é particularmente útil para identificar regiões de relevância comum mínima entre os conjuntos.

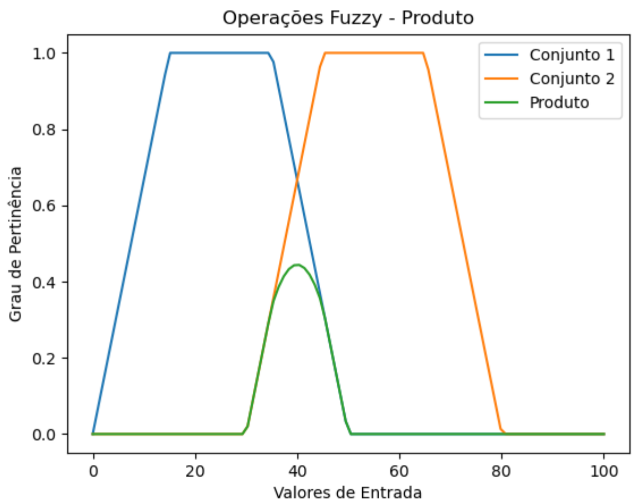


Figura 19: Operação de T-Norm Produto entre os conjuntos fuzzy.

3.5 S-Norm: Soma Probabilística

A S-Norm probabilística calcula o grau de pertinência pela soma dos valores de pertinência dos conjuntos, ajustados para evitar que o valor exceda 1, usando a fórmula $\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$. Observou-se que os resultados são semelhantes aos da união nos pontos de maior pertinência (25 e 50), atingindo 1.00 nesses pontos. Em regiões de menor sobreposição, como o ponto 75, o valor reduz-se para 0.35, mantendo a coerência com a intensidade da sobreposição. Nos pontos extremos 0 e 100, o grau de pertinência é 0.00, podendo-se refletir novamente a ausência de pertinência. A Figura 20 mostra que a S-Norm probabilística suaviza as transições, proporcionando uma alternativa à união que preserva valores de pertinência menores em regiões de menor sobreposição.

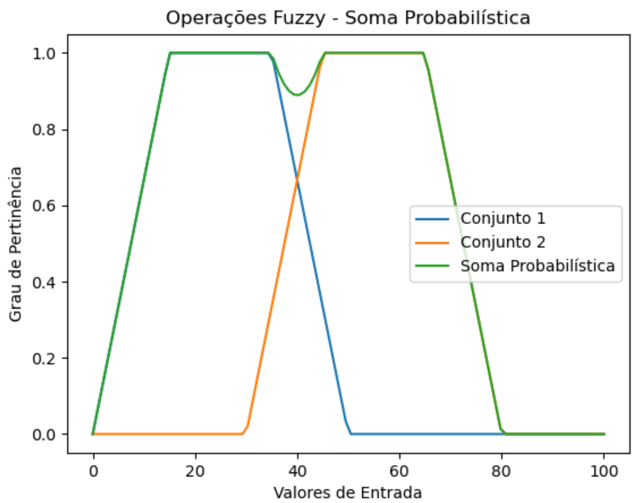


Figura 20: Operação de S-Norm Probabilística entre os conjuntos fuzzy.

Problema 4 : Implemente uma função para calcular uma relação fuzzy. A função deve receber como entrada a T-Norma (já implementadas anteriormente), o tamanho dos dois conjuntos fuzzy, os graus de pertinência de cada elemento do conjunto. A função deve retornar o resultado da relação. Cuidado com as exceções e operações indevidas. Apresenta um exemplo de execução comparando o resultado utilizando ao menos dois operadores para T-Norma e dois operadores para S-Norma.

Resposta:

Nesta etapa, realizou-se a análise de relações fuzzy aplicando t-normas e s-normas sobre um conjunto de dados retirado dos slides de aula[1], que representa a relação entre altura (conjunto fuzzy "ALTO") e idade (conjunto fuzzy "MEIA-IDADE") de indivíduos. O conjunto "ALTO"abrange alturas de 1.60m a 1.90m, enquanto o conjunto "MEIA-IDADE"abrange idades de 25 a 55 anos. Ambos conjuntos foram definidos com funções trapezoidais representados na figura 21 para obter os graus de pertinência em seus respectivos domínios, buscando uma análise das relações fuzzy entre esses atributos.

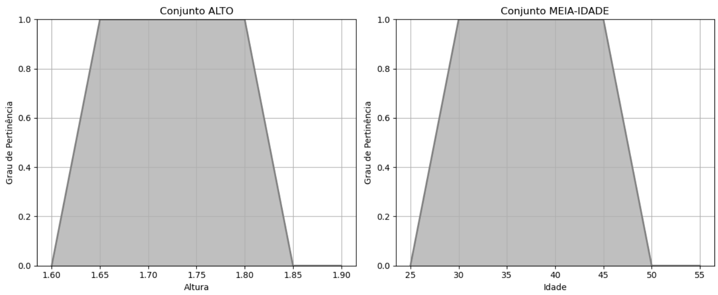


Figura 21: Funções trapezoidais representando os graus de pertinência.

T-Normas Utilizadas

- **Diferença Limitada**: Definida por $\max(0, x + y - 1)$, onde x e y são os graus de pertinência dos conjuntos. Esta operação reduz a interseção quando a soma dos graus não atinge 1, resultando em valores mais restritos.

- **Interseção Mínima**: Calculada como $\min(x, y)$, essa t-norma determina a interseção fuzzy diretamente pelo menor grau de pertinência entre os conjuntos comparados.

S-Normas Utilizadas

- **Soma Probabilística**: Definida por $x + y - (x \cdot y)$, permite que a união fuzzy ultrapasse os valores individuais, criando uma união probabilística.

- **Soma Limitada**: Calculada por $\min(1, x + y)$, restringe a união fuzzy ao valor 1, limitando valores excessivos.

4.1 Diferença Limitada

A t-norma de diferença limitada gera uma matriz de relação fuzzy onde as regiões de interseção entre os conjuntos são significativamente restritas. Essa operação limita as áreas de interseção aos valores que excedem uma soma de 1, enquanto os valores abaixo deste limiar são truncados para zero. A figura 22 ilustra a matriz de relação fuzzy resultante da diferença limitada, destacando uma área central densa de pertinência onde os conjuntos têm interseção.

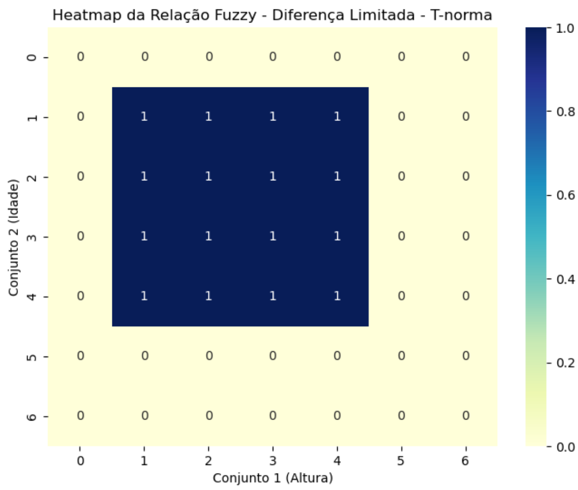


Figura 22: Heatmap da Relação Fuzzy - Diferença Limitada - T-Norma

4.2 Interseção Mínima

Na interseção mínima, a matriz de relação fuzzy exibe valores que são limitados pelo menor grau de pertinência entre os conjuntos "ALTO"e "MEIA-IDADE". Isso resulta em uma interseção fuzzy direta, conforme ilustrado na figura 23. A matriz obtida é semelhante à da diferença limitada, mas sem os cortes bruscos para zero, proporcionando uma visão mais direta das áreas de sobreposição fuzzy.

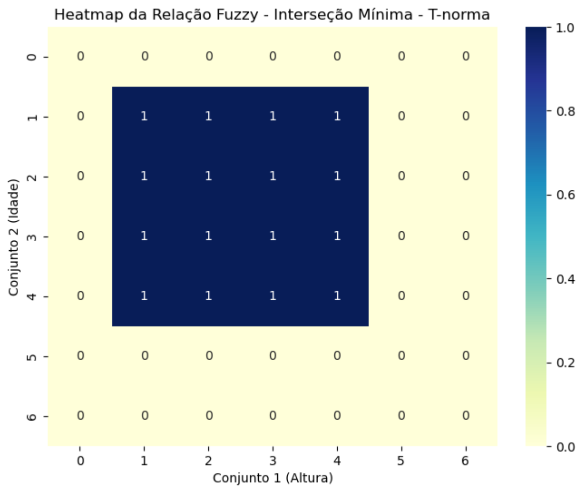


Figura 23: Heatmap da Relação Fuzzy - Interseção Mínima - T-Norma

4.3 Soma Probabilística

A s-norma de soma probabilística exibe uma matriz de relação com áreas de pertinência mais altas devido ao cálculo probabilístico que permite exceder o valor 1 em regiões de forte sobreposição. Como mostrado na figura 24, essa operação destaca as áreas onde ambos os conjuntos possuem altos graus de pertinência, proporcionando uma união fuzzy ampliada.

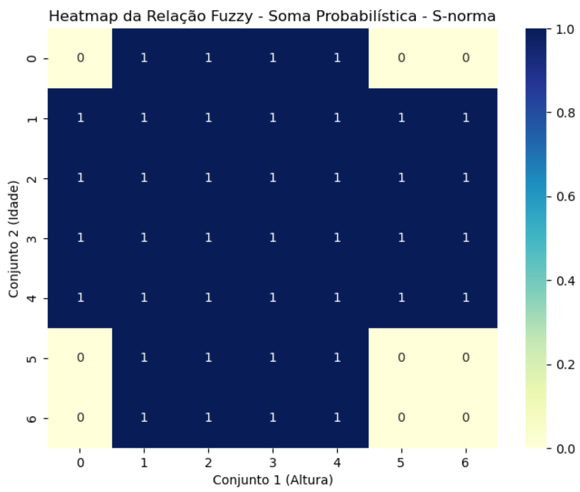


Figura 24: Heatmap da Relação Fuzzy - Soma Probabilística - S-Norma

4.4 Soma Limitada

A s-norma de soma limitada proporciona uma união controlada entre os conjuntos "ALTO"e "MEIA-IDADE", evitando que a soma ultrapasse o valor

1. Essa operação mantém uma restrição na união, como ilustrado na figura 25, mostrando um comportamento similar ao da soma probabilística, mas com valores truncados a 1, especialmente nas bordas.

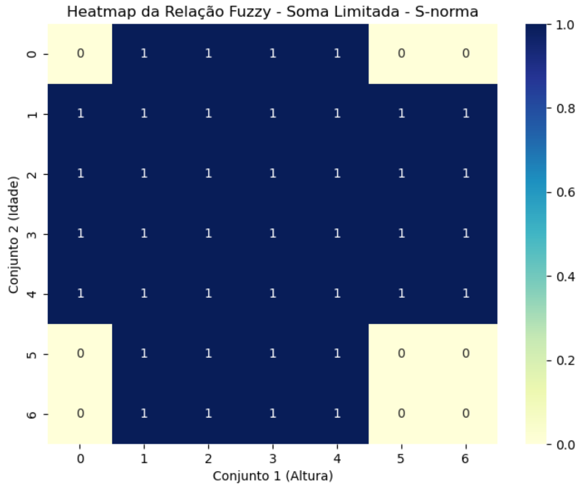


Figura 25: Heatmap da Relação Fuzzy - Soma Limitada - S-Norma

Problema 5 : Implemente uma função para calcular as composições Max-Min, Min-Max e Max-Prod. Cuidado com as exceções e operações indevidas. Execute o mesmo exemplo para as três composições.

Resposta:

Nesta etapa, realizou-se uma análise comparativa entre três métodos de composição fuzzy: *Max-Min*, *Min-Max* e *Max-Prod*, aplicados ao mesmo conjunto de dados do problema anterior, que representa a relação entre altura (conjunto "ALTO") e idade (conjunto "MEIA-IDADE"). Esses métodos de composição foram aplicados sobre os graus de pertinência obtidos para alturas entre 1.60m e 1.90m e idades entre 25 e 55 anos, ambos definidos por funções trapezoidais. A Figura 26 apresenta os resultados das três composições fuzzy.

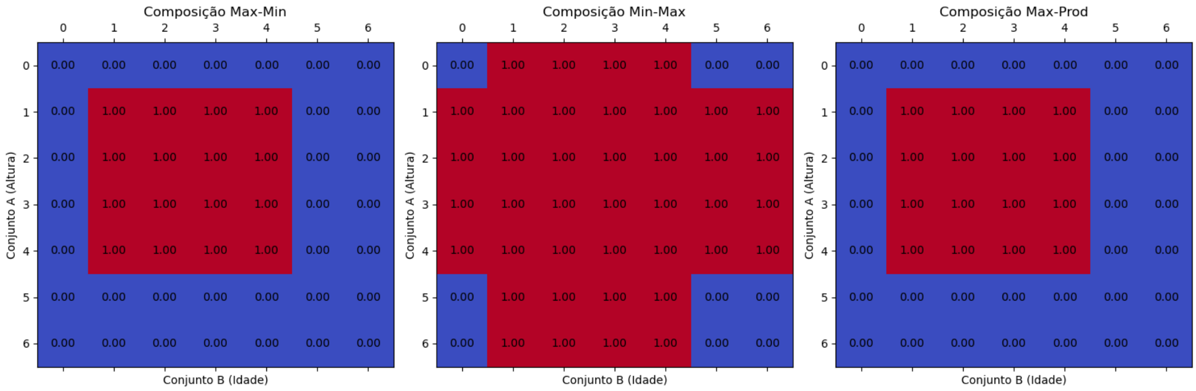


Figura 26: Comparação entre as composições fuzzy: Max-Min, Min-Max e Max-Prod.

Cada composição destaca características específicas das relações entre os conjuntos "ALTO"e "MEIA-IDADE", o que pode permitir observar suas diferenças e particularidades. A seguir, detalham-se os cálculos envolvidos em cada tipo de composição, buscando ilustrar como cada abordagem modela as interações fuzzy entre altura e idade.

5.1 Composição Max-Min

A composição Max-Min é calculada da seguinte forma:

$$R_{\text{Max-Min}}(x, y) = \max_{z} \left(\min \left(A(x, z), B(z, y) \right) \right)$$

onde $A(x, z)$ e $B(z, y)$ são os graus de pertinência dos conjuntos fuzzy para os elementos x, y e z. Neste método, calcula-se o mínimo entre os valores de pertinência correspondentes de A e B para cada par de elementos, e então toma-se o máximo desses mínimos para determinar o valor final. O resultado é uma matriz que apresenta uma área central com valores elevados (1.0), onde ambos os conjuntos têm alta pertinência.

5.2 Composição Min-Max

Na composição Min-Max, o cálculo é realizado da seguinte forma:

$$R_{\text{Min-Max}}(x, y) = \min_{z} \left(\max \left(A(x, z), B(z, y) \right) \right)$$

Neste método, primeiro calcula-se o máximo entre os valores de pertinência de A e B para cada par de elementos x, y e z, e depois toma-se o mínimo desses máximos. O resultado, como pode ser visto no segundo gráfico, é uma área expandida de valores altos na matriz. Isso ocorre porque a composição Min-Max utiliza o mínimo dos máximos, ampliando a união entre os conjuntos.

5.3 Composição Max-Prod

A composição Max-Prod utiliza o seguinte cálculo:

$$R_{\text{Max-Prod}}(x, y) = \max_{z} \left(A(x, z) \cdot B(z, y)\right)$$

onde $A(x, z) \cdot B(z, y)$ representa o produto dos graus de pertinência dos conjuntos A e B para cada par (x, z) e (z, y). O valor final para cada par (x, y) é o máximo desses produtos. Esse método, como mostrado no terceiro gráfico, destaca uma área central com valores altos, mas apresenta variações nas bordas da região de alta pertinência devido à ponderação dos produtos.

5.4 Comparação entre Composições

Dessa forma, é possível concluir que as três composições fuzzy — Max-Min, Min-Max e Max-Prod — revelam padrões semelhantes de distribuição de pertinência entre os conjuntos de altura e idade, com diferenças sutis na forma de agregar os valores. A composição Max-Min mostra uma área central bem delimitada com valores altos, que pode refletir uma interseção mais rigorosa onde ambos os conjuntos têm elevada pertinência. A Min-Max expande esses valores de 1.0 para quase toda a matriz, exceto nas extremidades, tornando-se a mais inclusiva, pois maximiza a união dos conjuntos. Por outro lado, a Max-Prod equilibra entre os dois métodos, mantendo uma área central semelhante à Max-Min, mas suavizando gradativamente nas bordas devido ao produto dos graus de pertinência. Essas diferenças indicam que, embora similares, cada método de composição pode capturar aspectos ligeiramente diferentes das relações fuzzy, dependendo da aplicação desejada.

Modelo e Reprodução

Para garantir a reprodutibilidade dos resultados obtidos neste estudo de relações e composições fuzzy, todo o código-fonte, incluindo as implementações das operações fuzzy (Complemento, União, Interseção, T-Normas, S-Normas) e das composições (Max-Min, Min-Max e Max-Prod), está disponível em um repositório GitHub público[2]. O diretório `src/` organiza os scripts em módulos Python que permitem tanto a reprodução direta das análises quanto a possibilidade de adaptações e experimentos adicionais.

Referências

[1] Alisson Marques da Silva. Inteligência computacional - sistemas fuzzy: Operações e relações. Apresentação de slides, 2024. Disponível para acesso mediante solicitação.

[2] Celso Vinícius. https://github.com/celzin/Atividade-01-IC, 2024. Acesso em: 04 nov. 2024.