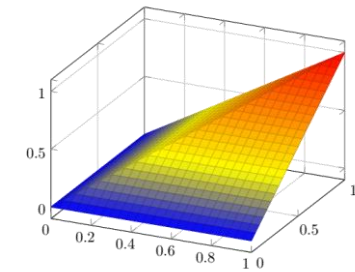
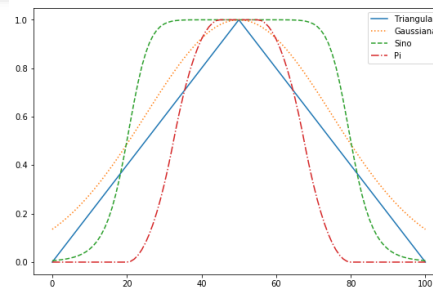
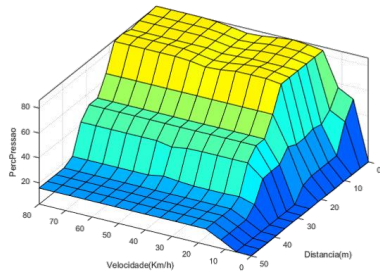


Prof. Geraldo Xexéo, D.Sc.

# Conjuntos e Lógica Fuzzy

## Introdução a Teoria e Aplicações



# Slides para acompanhar o livro

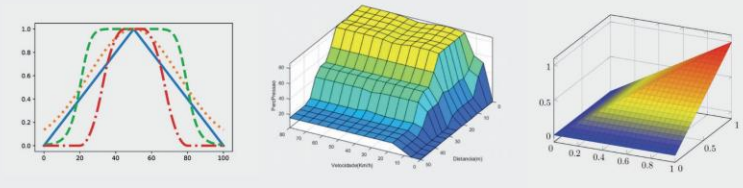
Prof. Geraldo Xexéo, D.Sc.

---

## Conjuntos e Lógica Fuzzy

### Introdução à Teoria e Aplicações

---



EDITORA CIÊNCIA MODERNA

# Material

- Xexéo, G. (2022) Conjuntos e Lógica Fuzzy: Introdução a Teoria e Aplicações. Ciência Moderna, Rio de Janeiro
- <https://lcm.com.br/site/livros/detalhesLivro/f/conjuntos-e-logica-fuzzy---introducao-a-teoria-e-aplicacoes.html>

# Sumário

- Motivação
- Conjuntos Nebulosos
- Operações em Conjuntos Fuzzy
- Princípio da Extensão
- Aritmética Nebulosa
- Quão Nebuloso é um Sistema Nebuloso
- Lógica Nebulosa
- Sistemas de Controle Fuzzy
- Exemplo: Controle Fuzzy+Matlab
- Desenvolvimento de Sistemas Fuzzy
- Determinando Funções de Pertinência
- Determinando Conjuntos de Regras
- Bancos de Dados Fuzzy
- Conclusões

# Motivação



# Qual a cor da Terra?



Imagem cortesia da Nasa Johnson Space Center

# Respostas Apropriadas

- A Terra é na sua maior parte azul
- A Terra é quase azul
- A Terra é 66% azul
- A Terra é azulada
- A Terra é muito azul



# Motivação

- “Tão próximas as leis da matemática estejam da realidade, menos próximas da certeza elas estarão. E tão próximas elas estejam da certeza, menos elas se referirão à realidade”
- (Albert Einstein)



# Duas Regras da Lógica Tradicional

## Princípio da Bivalência

- Toda proposição é verdadeira ou falsa, mas não simultaneamente.

## Terceiro excluído

- Toda proposição é verdadeira ou não verdadeira

# Lógica de Boole - 1848

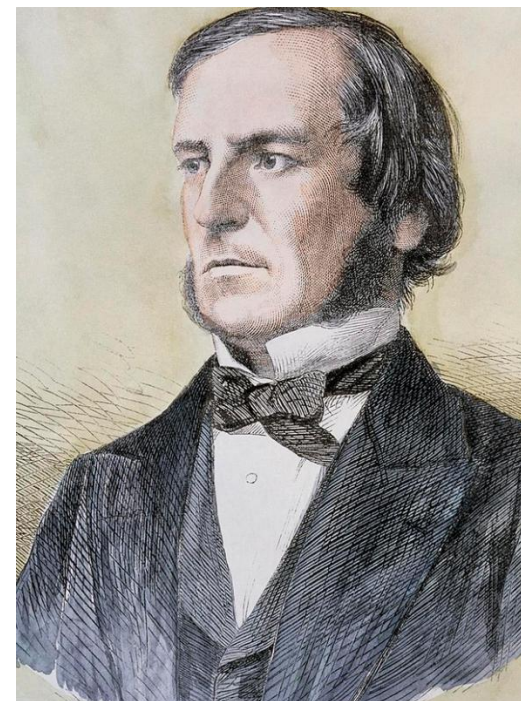
$\wedge$	F	V
F	F	F
V	F	V

$\vee$	F	V
F	F	V
V	V	V

$\neg$	
F	V
V	F

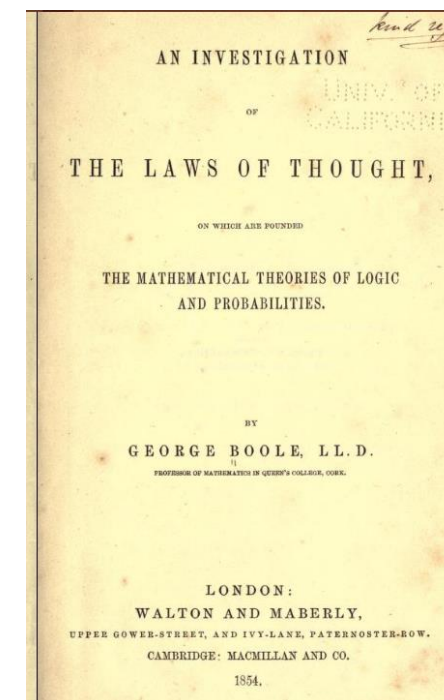
$\Rightarrow$	F	V
F	V	V
V	F	V

$\Leftrightarrow$	F	V
F	V	F
V	F	V



George Boole (circa 1864)

Imagem em domínio público



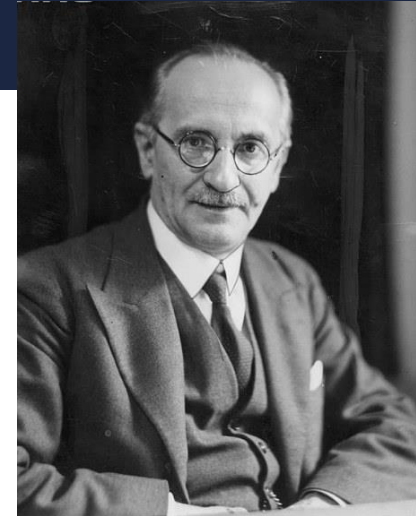
# A questão

- O mundo real apresenta situações onde as respostas verdadeiro e falso não são suficientes para representar a realidade
  - Tanto a resposta Verdadeiro quanto a resposta Falso não são apropriadas
  - Qual seria a resposta apropriada?

**GRAUS DE VERDADE**

# 1920!!!!

- Jan Łukasiewicz inventa uma lógica tri-valorada
  - Verdadeiro, falso e talvez (ou indeterminado)
  - 0, 1 e  $u$
  - 0, 1 e  $1/2!$
- Mais tarde, estende para uma quantidade infinita de valores entre 0 e 1
- Lógica de Łukasiewicz



Jan Łukasiewicz (1935)

Imagem em domínio público

# Lógica de 3 valores

<b>OU</b>	<b>0</b>	<b>u</b>	<b>1</b>
<b>0</b>	0	u	1
<b>u</b>	u	u	1
<b>1</b>	1	1	1

<b>E</b>	<b>0</b>	<b>u</b>	<b>1</b>
<b>0</b>	0	0	0
<b>u</b>	0	u	u
<b>1</b>	0	u	1

<b>Não</b>	
<b>0</b>	1
<b>u</b>	u
<b>1</b>	0

<b>=&gt;</b>	<b>0</b>	<b>u</b>	<b>1</b>
<b>0</b>	1	1	1
<b>u</b>	u	1	1
<b>1</b>	0	u	1

<b>&lt;=&gt;</b>	<b>0</b>	<b>u</b>	<b>1</b>
<b>0</b>	1	u	0
<b>u</b>	u	1	u
<b>1</b>	0	u	1

# 1960-1965



Lotfali Askar-Zadeh  
Lotfi Zadeh

Foto CC-SA do próprio

- Lotfi Zadeh inventa a lógica fuzzy:
  - Fuzzy, nebulosa
  - ou cabeluda/peluda
- Existe uma gradação entre falso e verdadeiro

**Problemas ao escolher o nome**

# Nítido x Fuzzy



# Fim dos limites

- Modelos de Fronteiras Nítidas

- Normalmente escolhidas de forma ad-hoc, mesmo que com apoio de estatísticas e especialistas

- Modelos de Fronteiras Fluídas

- representando melhor a incerteza ao criar o conceito

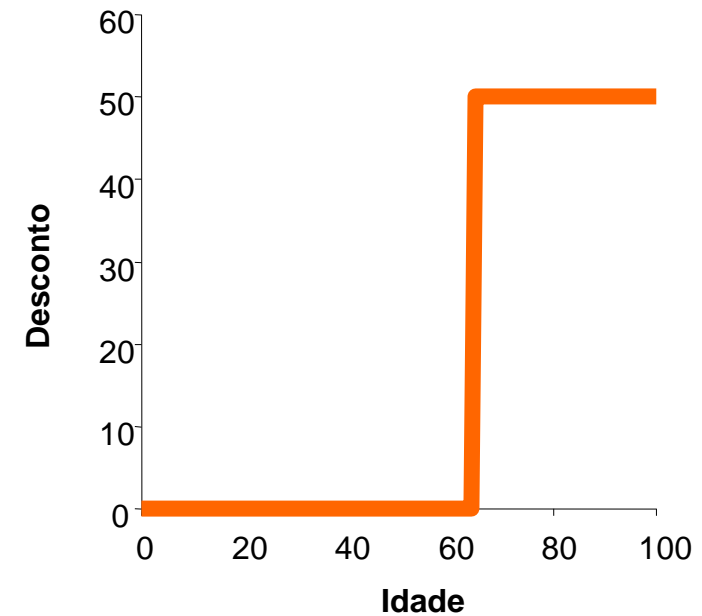


# Conceitos Vagos

- Alto, Baixo
- Novo, Velho, Usado, Semi-novos
- Perto, Longe
- Leve, Pesado, Suportável
- Gelado, Frio, Morno, Tépido, Quente
- Crítico, Condicionante, Dispensável

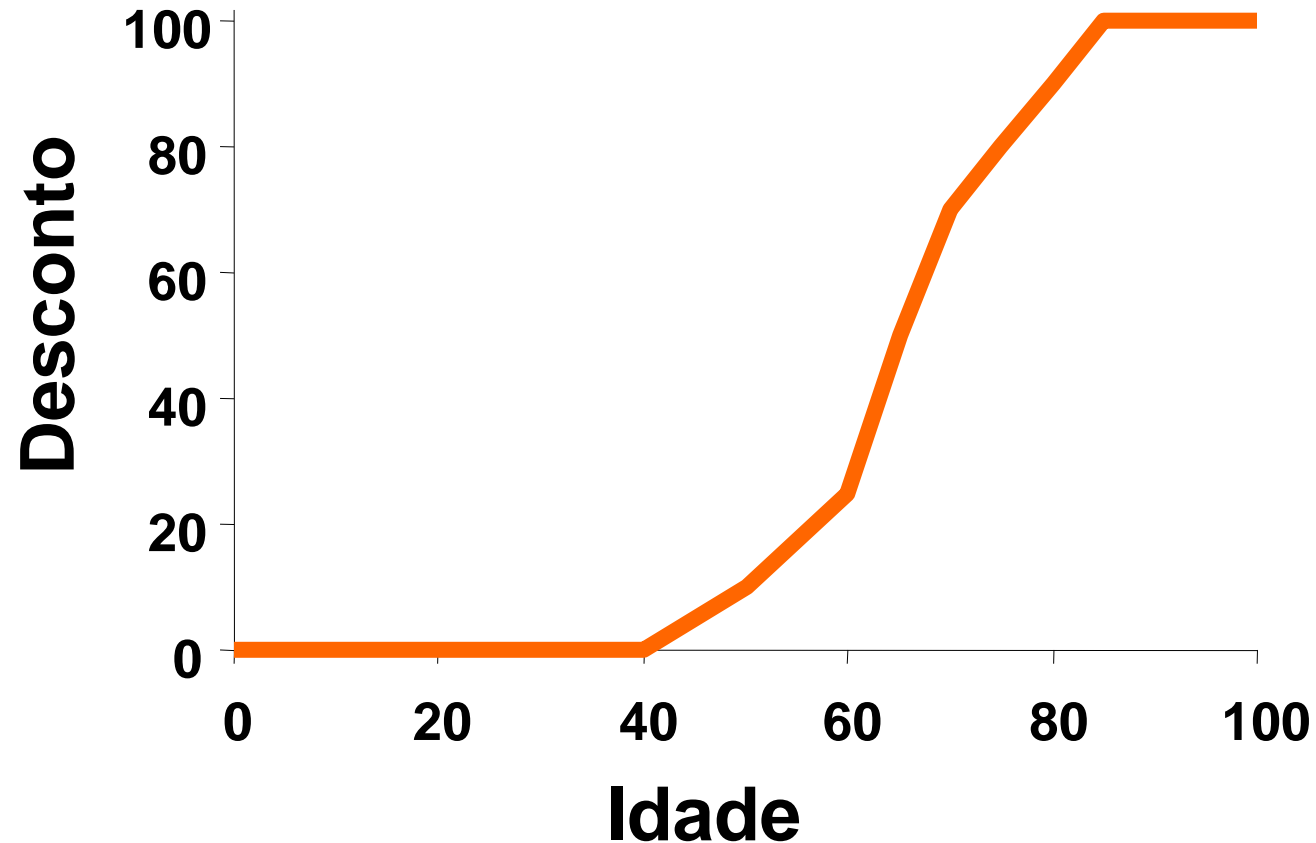
# Exemplo Simples Nítido

- Direito a desconto por ser idoso:
  - 65 anos = 50% de desconto
  - Um dia antes de fazer aniversário
    - Desconto 0%
  - No dia do aniversário
    - Desconto 50%
- Qual a “lógica” disso?



# Exemplo Simples Fuzzy

- Descontos progressivos com a idade!



# Vários Tipos de Incerteza

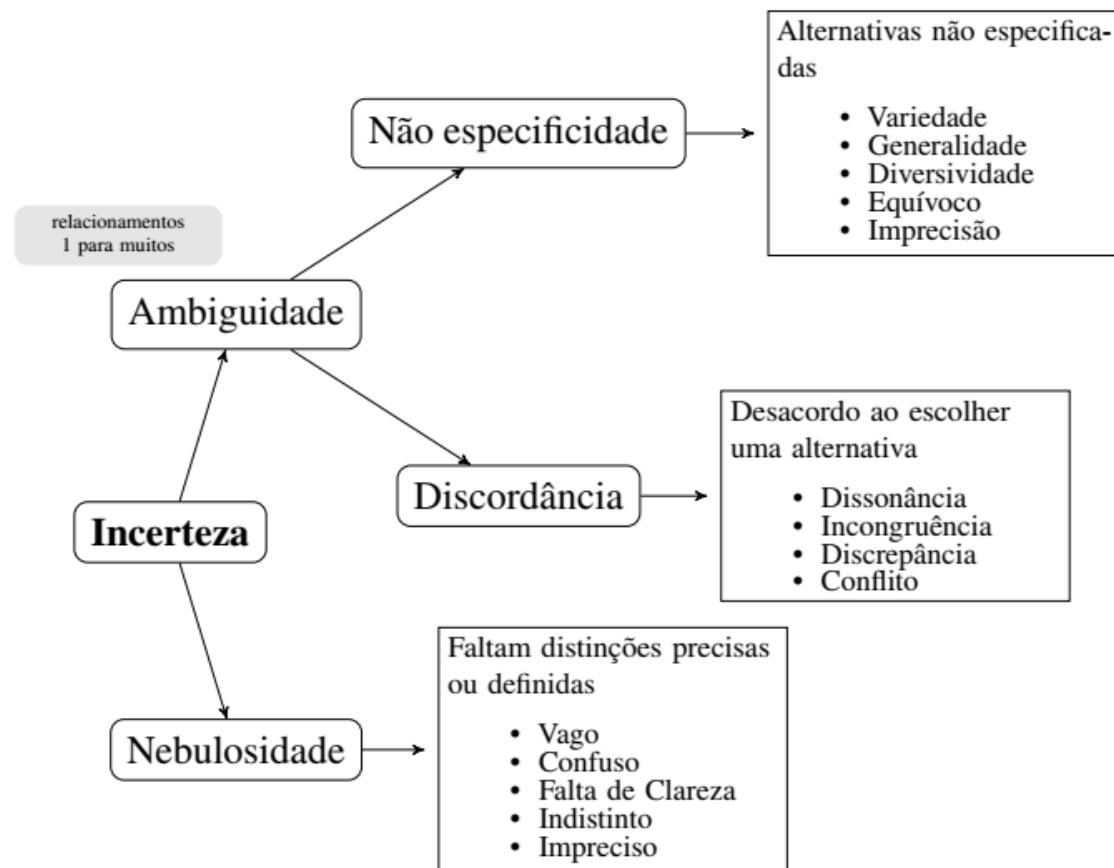
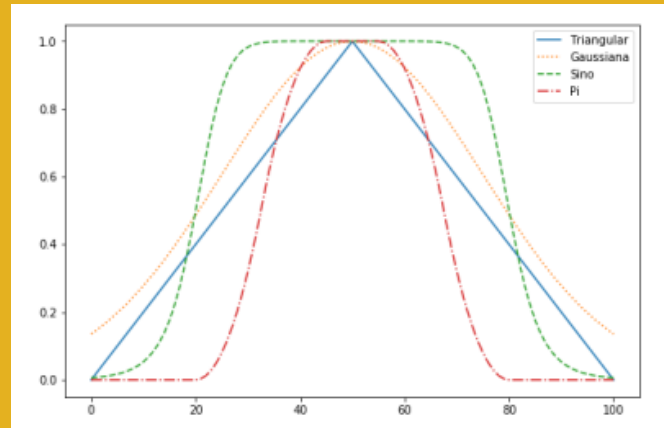


Figura 1.2: Tipos de incerteza (Klir e Yuan, 1995)

# Conjuntos Nebulosos

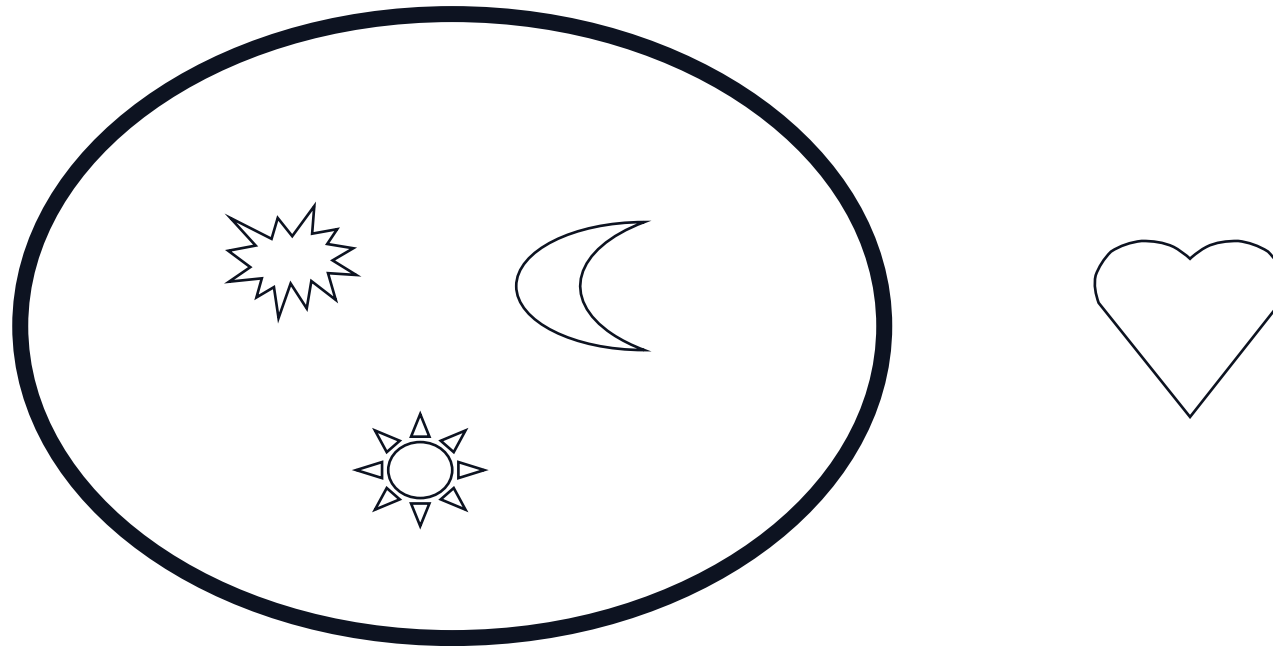


# Teoria dos Conjuntos Nebulosos

- Uma generalização da teoria dos conjuntos,
  - onde elementos pertencem ou não pertencem a conjuntos,
- para os sistemas do mundo real
  - onde, os problemas ,em sua maioria, são compostos de conjuntos de contornos não definidos.

# Conjuntos Tradicionais – Coisas Pontudas

- Elementos pertencem ou não ao conjunto



# Teoria Ingênua dos Conjuntos

- Termos não definidos
  - Conjunto
  - Elemento
  - Pertinência
- Conceitos definidos
  - União  $A \cup B$ 
    - $\{x \mid x \in A \vee x \in B\}$
  - Interseção  $A \cap B$ 
    - $\{x \mid x \in A \wedge x \in B\}$

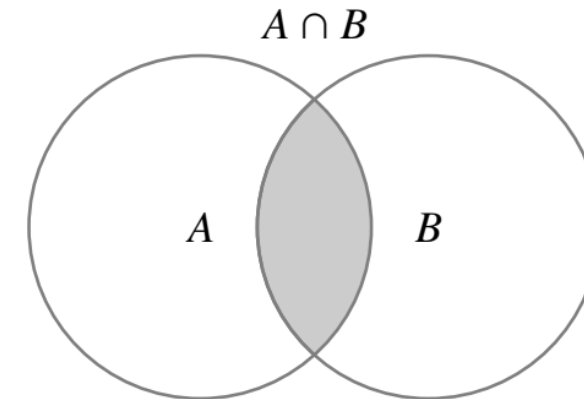


Figura 2.3:  $A \cap B$

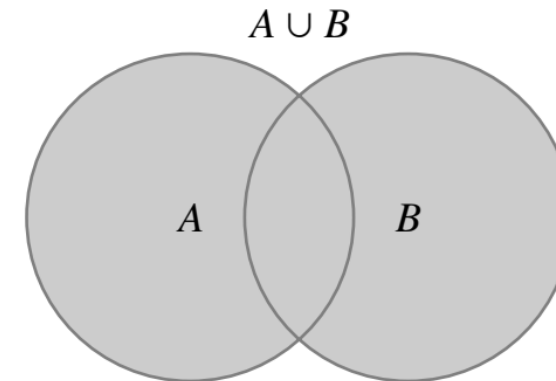


Figura 2.4:  $A \cup B$



# Conjuntos Nítidos

- Conjuntos Nítidos: elementos pertencem ou não (função característica)

$$\chi_A(x) = \begin{cases} \mathbf{1} & \text{para } x \in A \\ \mathbf{0} & \text{para } x \notin A \end{cases}$$

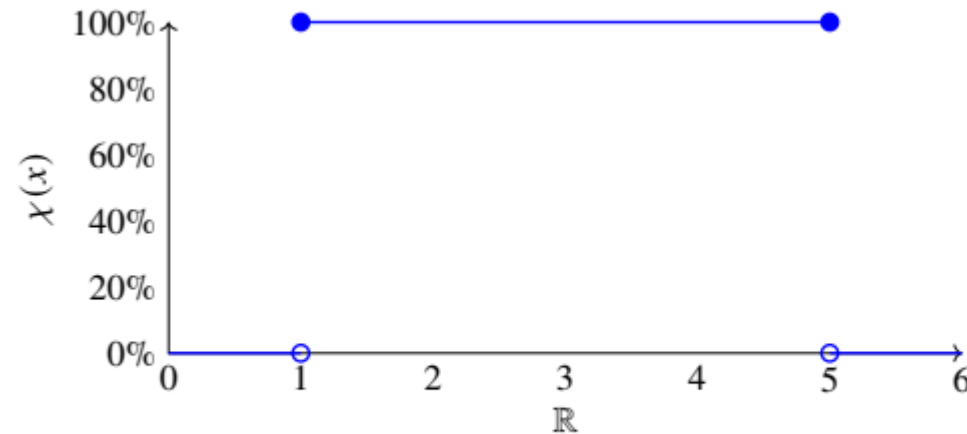
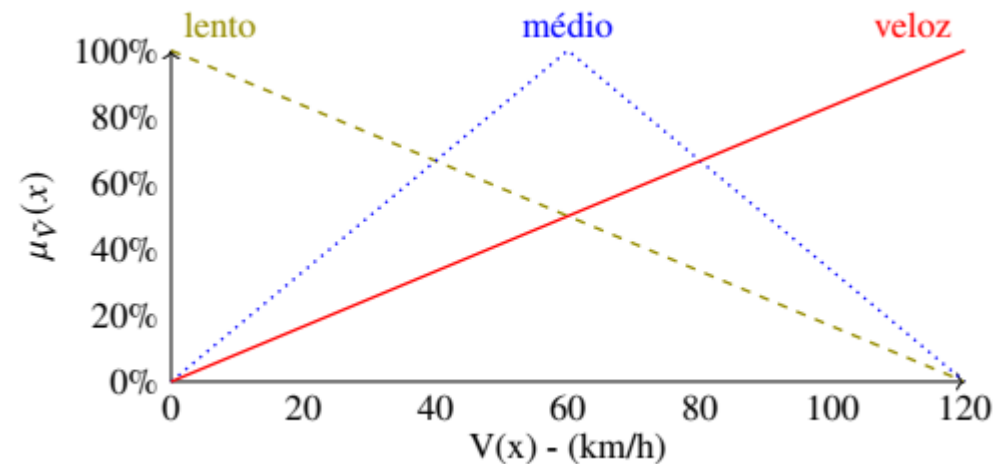


Figura 2.2: Função característica do conjunto  $\{x|(x \in \mathbb{R}) \wedge (1 \leq x \leq 5)\}$ .

# Conjuntos Nebulosos

- Um elemento pode pertencer "parcialmente" a um conjunto.
- Chamamos de possibilidade à quantificação da pertinência do elemento ao conjunto
  - grau de pertinência =  $\mu_A(x)$  ou  $m_A(x)$

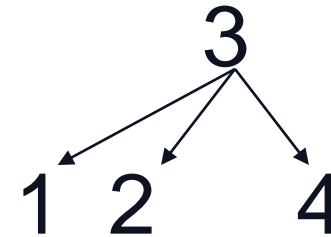


# Conjuntos Nebulosos

$$\mu_A(x) : U \rightarrow [0, 1]$$

- A é um subconjunto Fuzzy de U
- $\mu_A$  é a função de pertinência
- Cada elemento de A têm um grau de pertinência
- Permitem criar conjuntos que representam conceitos “humanos”

P=Próximo de 3



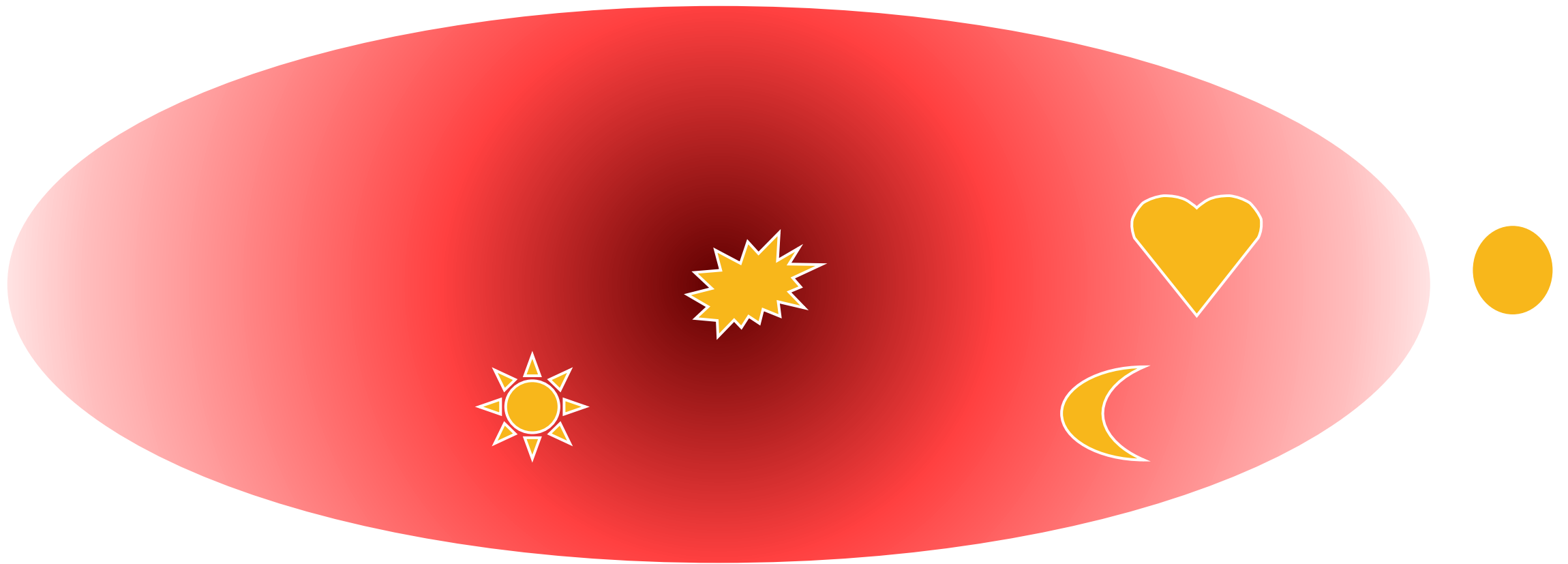
$$\mu_P(3)=1$$

$$\mu_P(2)=0,5$$

$$\mu_P(4)=0,5$$

$$\mu_P(1)=0,25$$

# Conjuntos Fuzzy – Coisas Pontudas



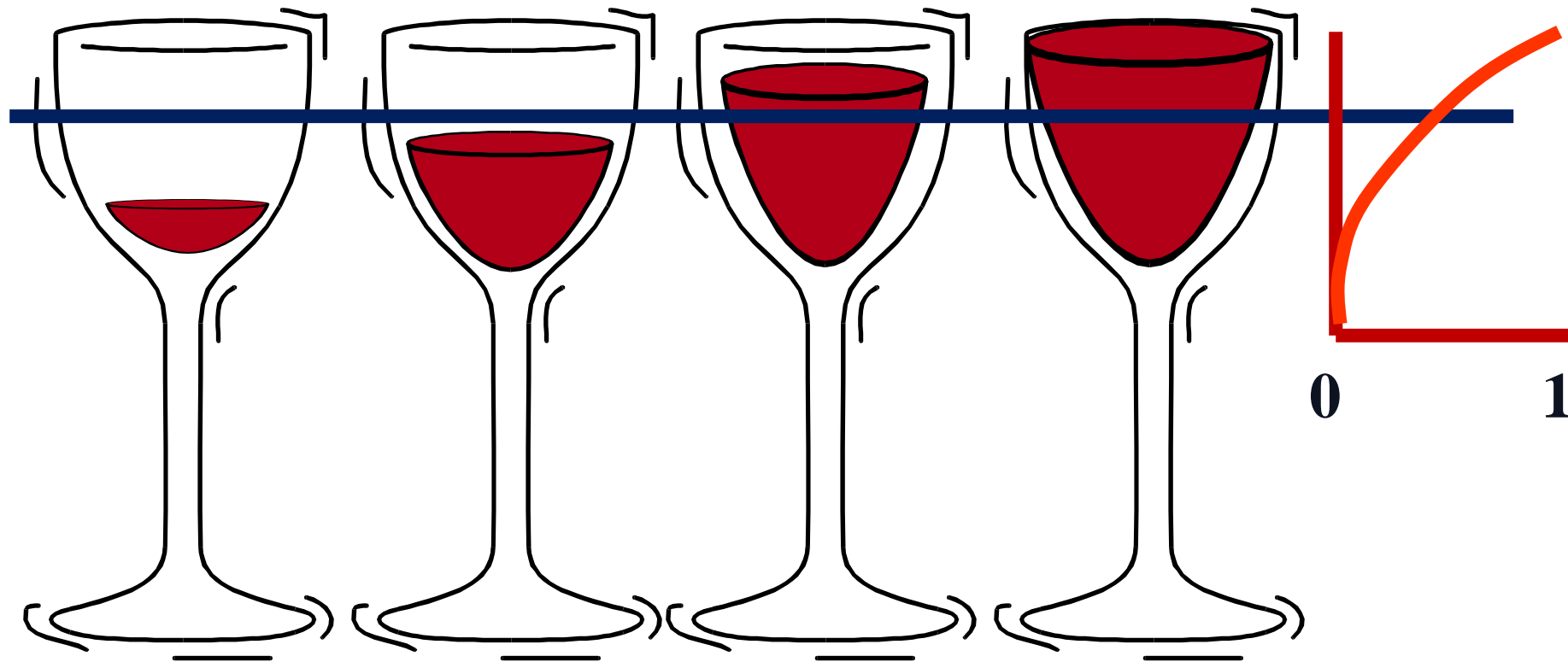
# Função Pertinência mantém a Função Característica

$$x \in A \Rightarrow \mu_A(x) = 1$$

$$x \notin A \Rightarrow \mu_A(x) = 0$$

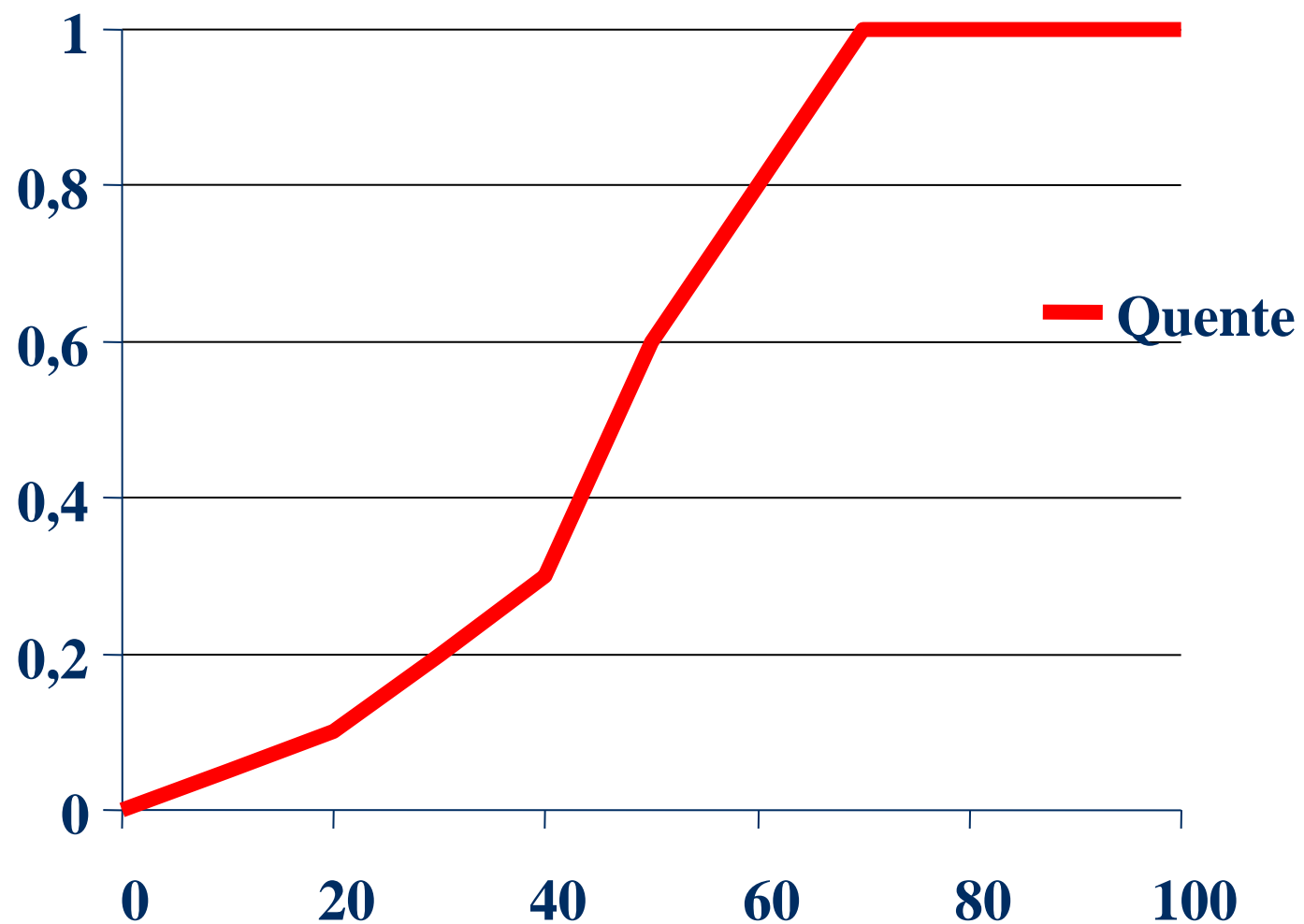
# Copos Cheios?

Cheio



Vazio

# A água está quente



# Funções de Pertinência

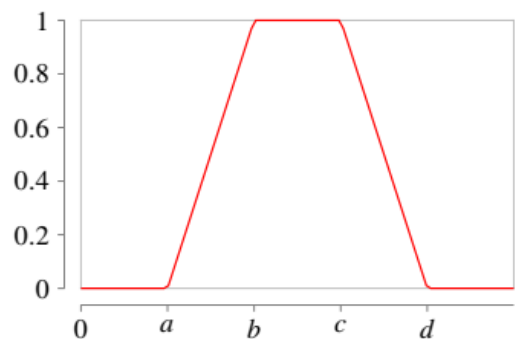


Figura 2.14: Função linear segundo a Eq. 2.37.

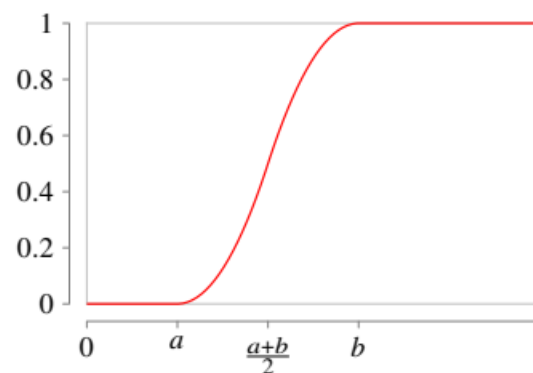
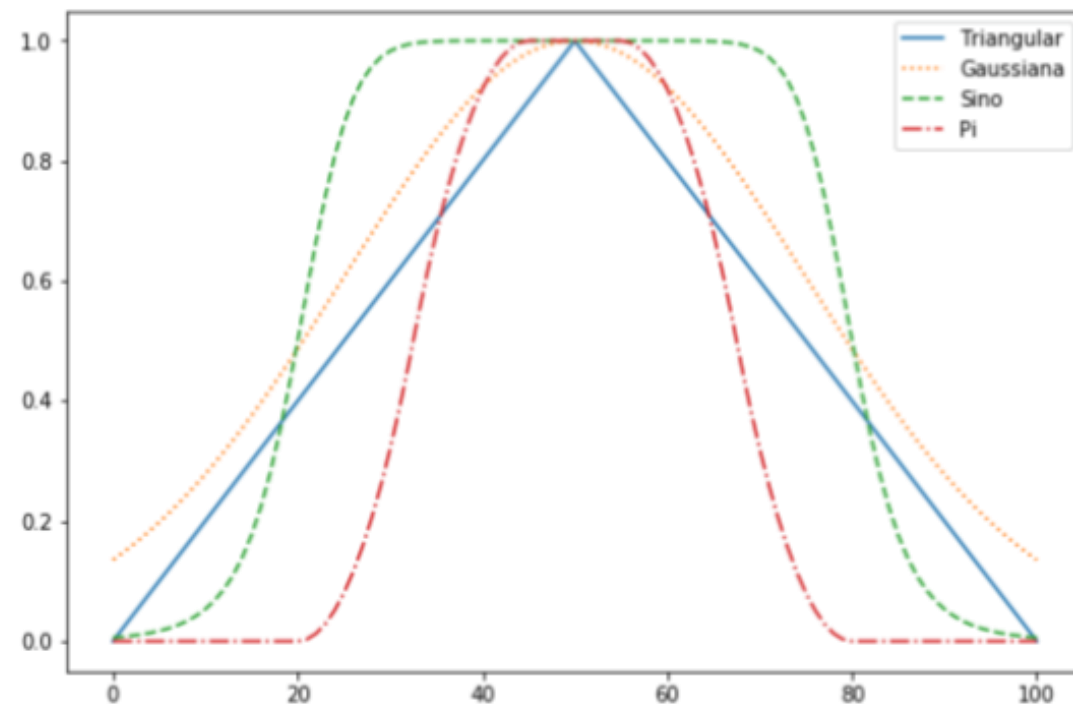
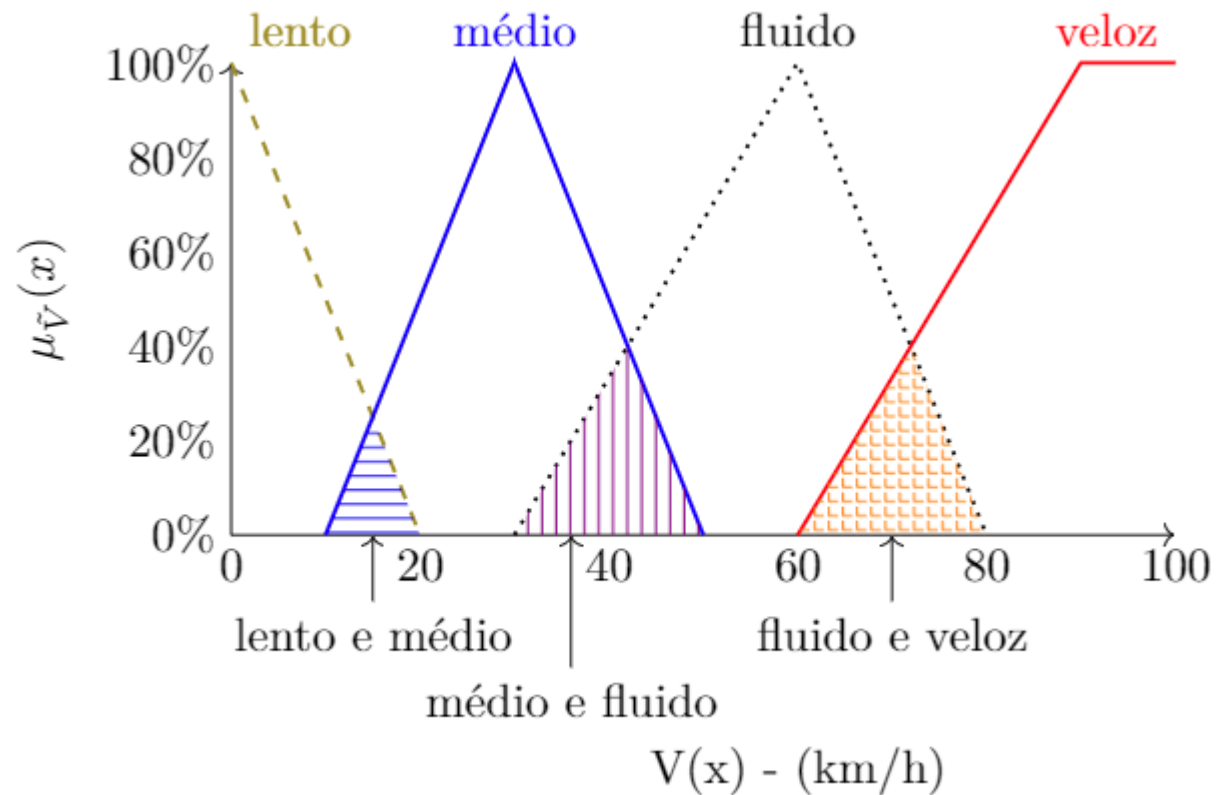


Figura 2.15: Função S segundo a Eq. 2.38.





# Nebulosidade



# Operações Fuzzy

- Interseção
  - $\mu(A \cap B) = \text{Min}(\mu(A), \mu(B))$
- União
  - $\mu(A \cup B) = \text{Máx}(\mu(A), \mu(B))$
- Complemento
  - $\mu(A^c) = 1 - \mu(A)$

**Existem outros  
modelos mais  
complexos**

# Interseção

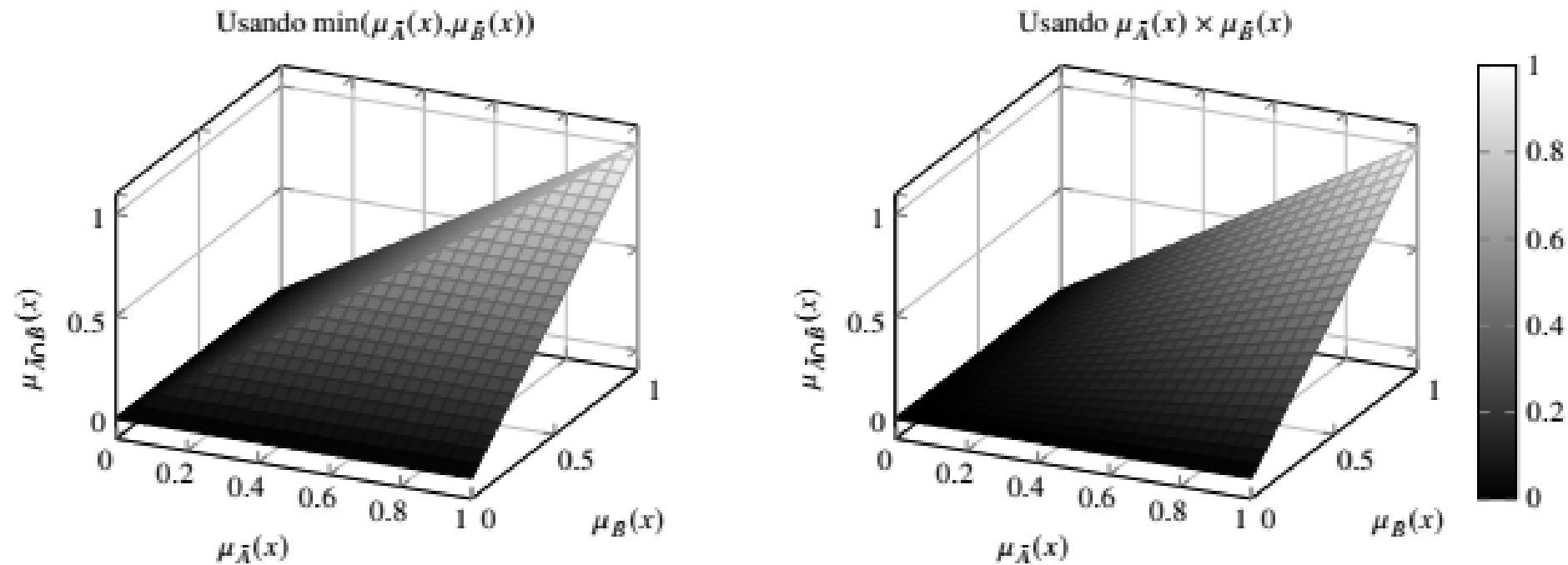


Figura 2.10: Gráficos para dois operadores possíveis para a interseção.

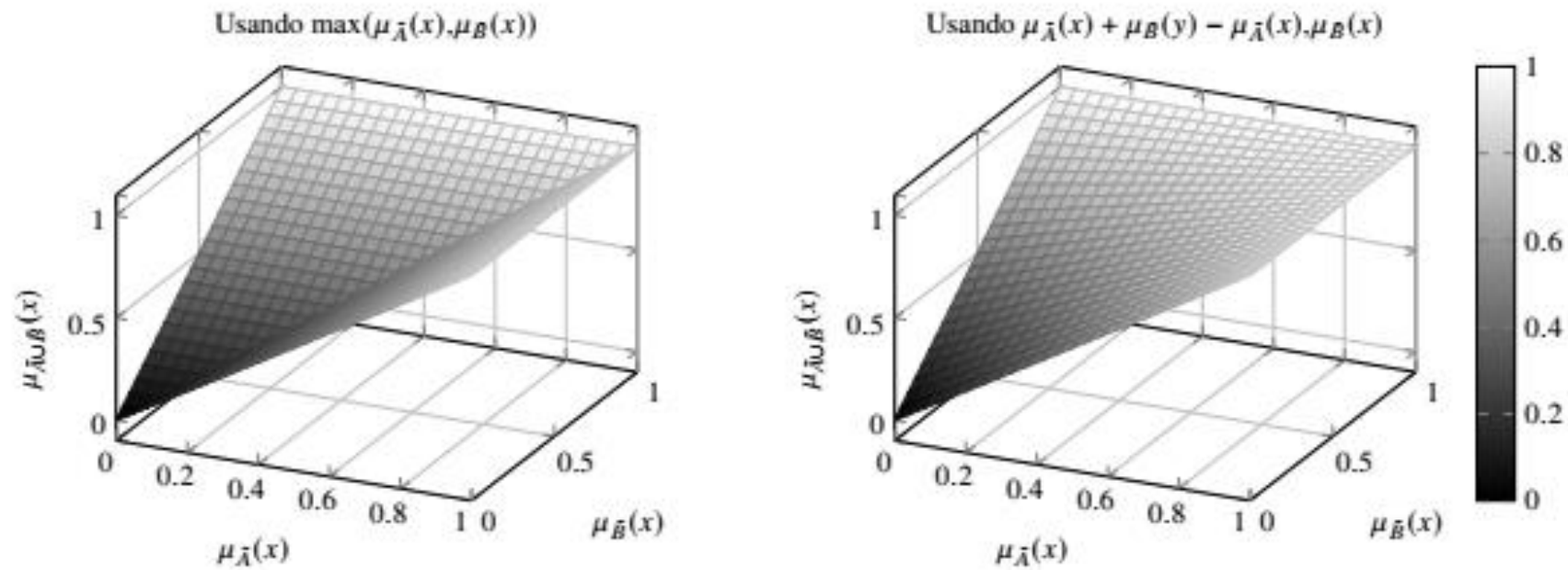
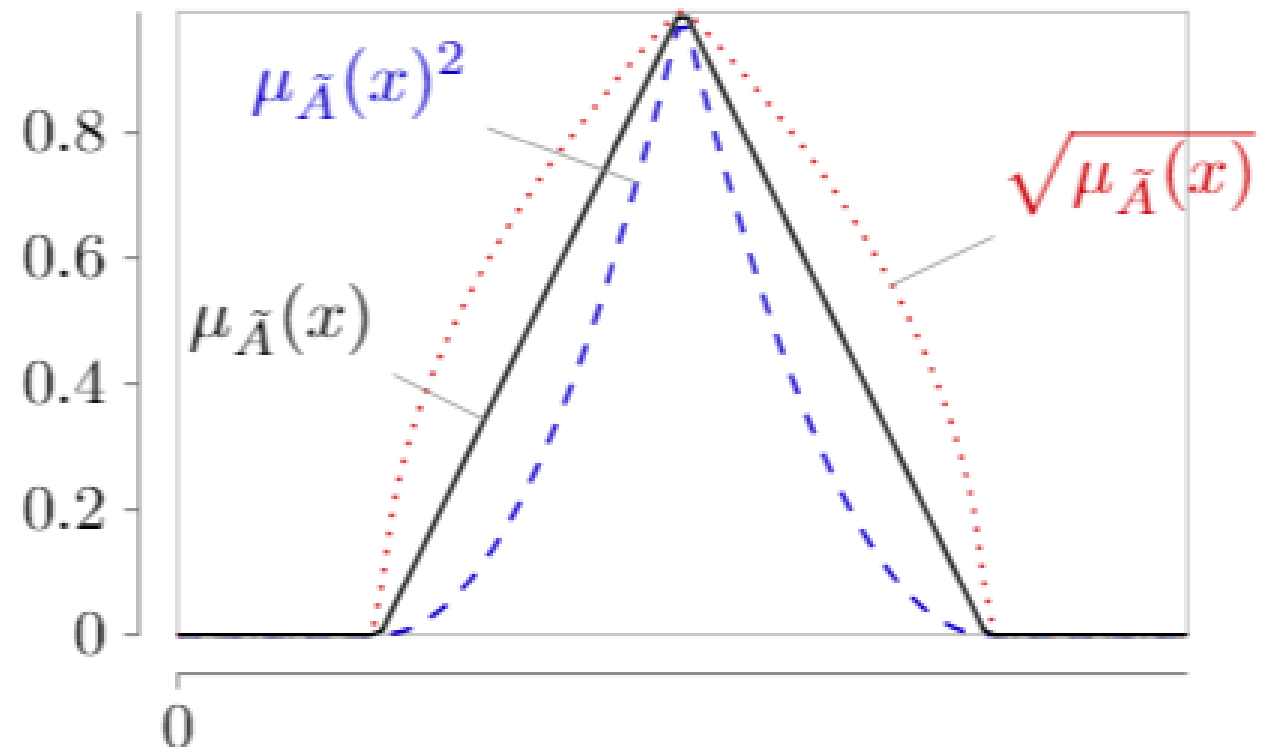


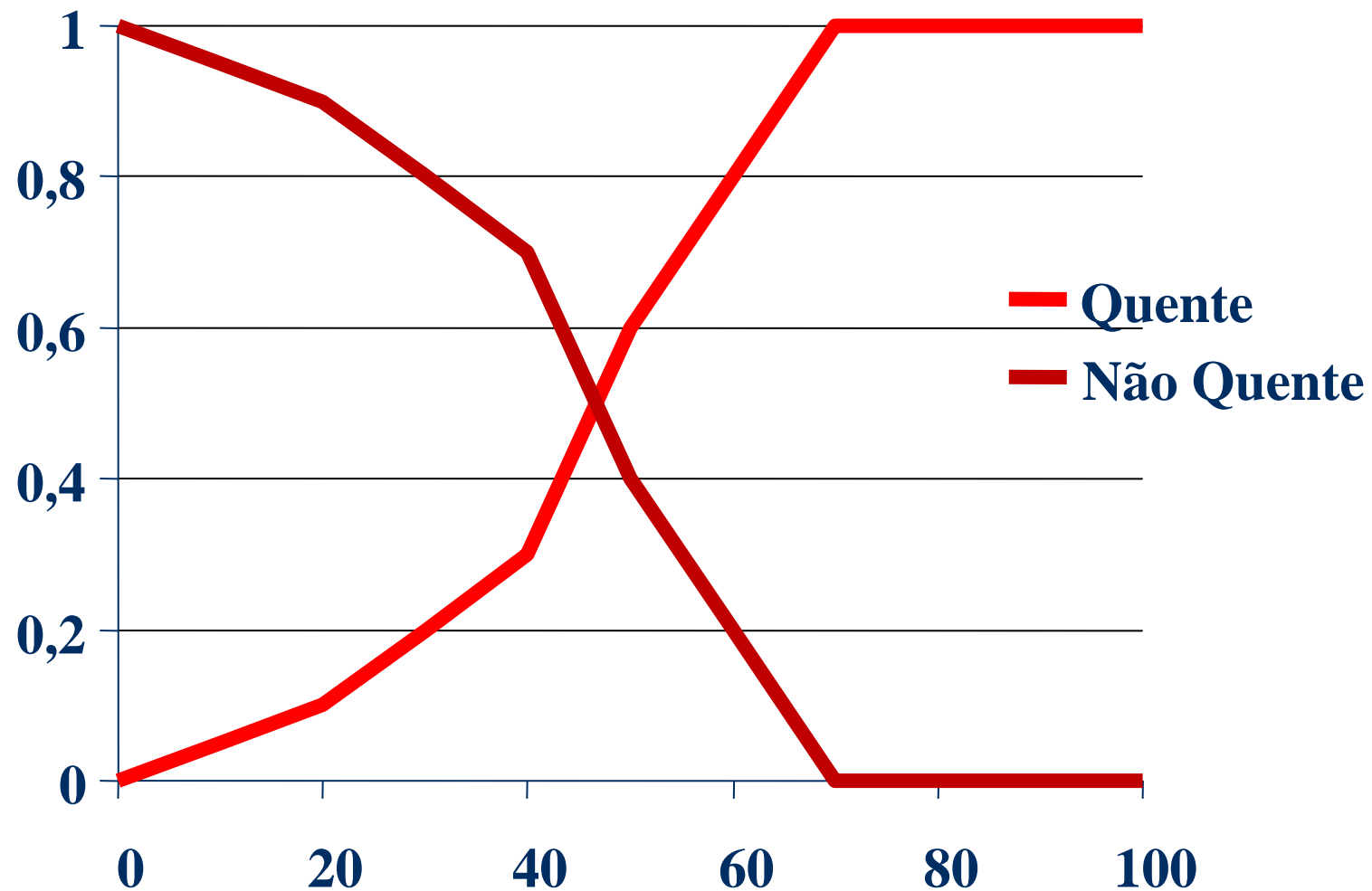
Figura 2.11: Gráficos para dois operadores possíveis para a união.

# Modificadores (Hedges)

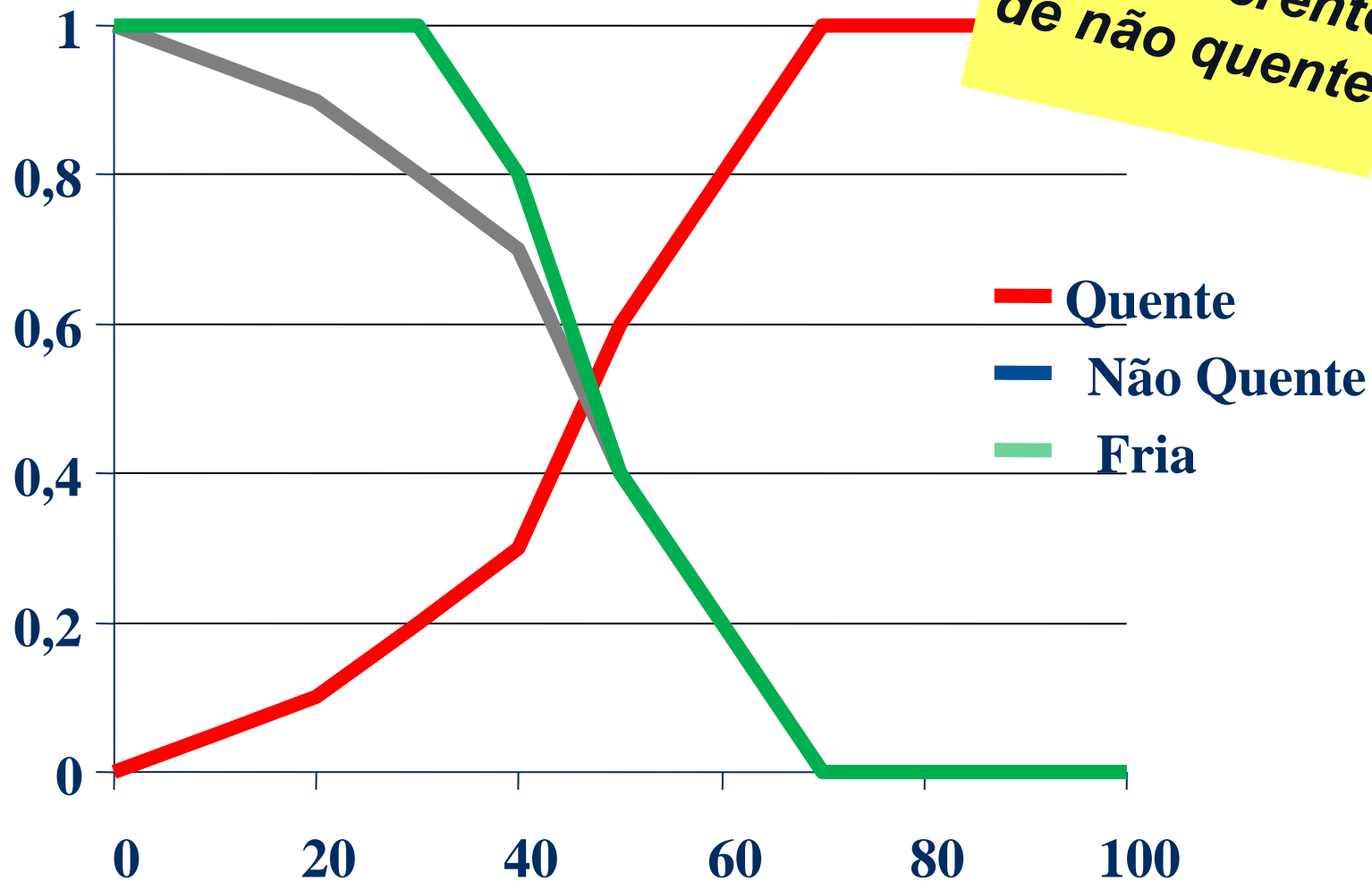
- Concentração (muito)
  - $m(\text{muito } A) = m(A)^2$
- Dilatação (algo)
  - $m(\text{muito } A) = m(A)^{1/2}$
- Qualquer outro termo que possa ser agregado ao conceito e que possua uma interpretação
  - Lembrar que o grau de pertinência é entre 0 e 1



# A água não está Quente



# A água está fria



Conceito:  
Frio é diferente  
de não quente

# Notações

- Como funções ou gráficos

- Quando finito:

$$\mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n$$

- Se grande, contável, infinito

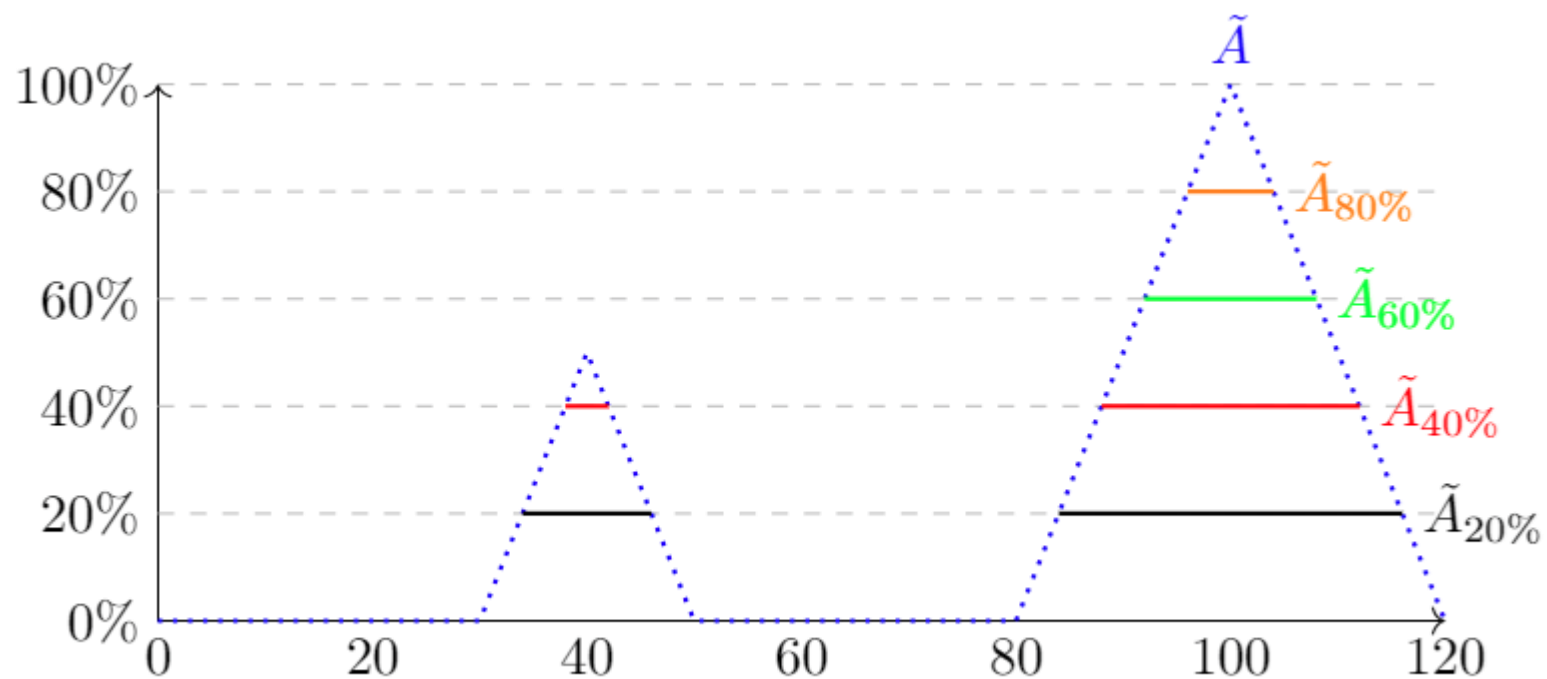
$$A = \sum_{i=1}^n \mu_A(x_i) / x_i \quad A = \int_x \mu_A(x_i) / x_i$$



# Propriedades

- Altura: é o valor supremo
- Suporte: pertinência maior que zero
- Corte-alfa: pertinência acima de um valor escolhido

# Corte Alfa

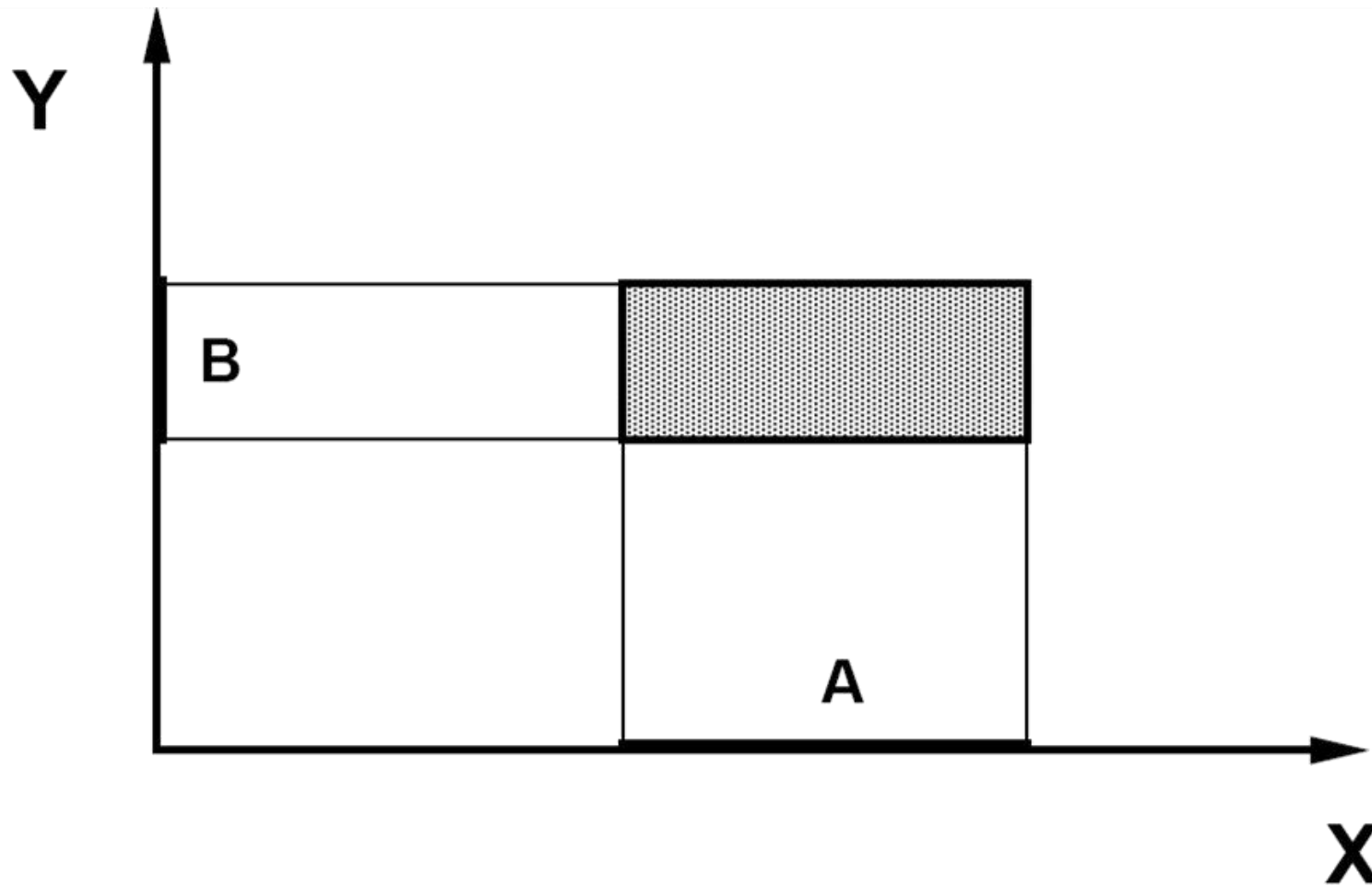


# Operações em Conjuntos Fuzzy

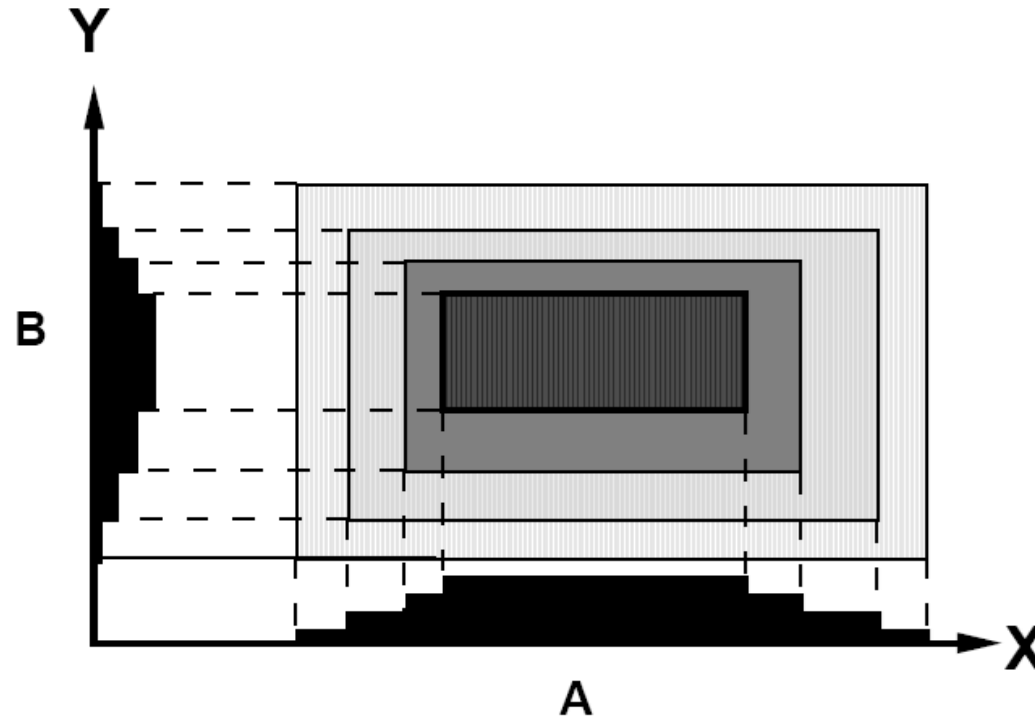
# Produto Cartesiano

- $A \times B$  é o conjunto de todos os pares ordenados onde o primeiro elemento pertence a  $A$  e o segundo a  $B$
- $A = \{ 1, 2, 3 \}$
- $B = \{ 1, 2 \}$
- $A \times B = \{ (1,1), (1,2), (2,1), (2,2), (3,1), (3,2) \}$

# Produto Cartesiano



# Produto Cartesiano de Conjuntos Fuzzy



- $A: X \rightarrow [0,1], B: Y \rightarrow [0,1]$
- $A \times B: X \times Y \rightarrow [0,1]: (x,y) \rightarrow \min(A(x), B(y))$

# Cardinalidade

$$|A| = \sum_{x \in X} \mu_A(x)$$

$$|A| = \int_x \mu_A(x) dx$$

# Operações

- As operações de interseção e união são mapeadas em classes de funções denominadas de *t-norma* e *t-conorma*.
- Usaremos a notação
  - $\mathcal{T}(a(x), b(x))$  *t-norma*
  - $\mathcal{U}(a(x), b(x))$  *t-conorma*
  - $\mathcal{C}(a(x))$  complemento genérico.



# Operações t-norma, t-conorma

- Os axiomas de 1 a 4 são necessários para caracterização dos operadores. Os axiomas 5 a 7 são por vezes desejáveis.
  1. Contorno
  2. Monotonicidade
  3. Comutatividade
  4. Associatividade
  5. Continuidade
  6. Sub-Idempotência
  7. Monotonicidade Estrita

$$I(a, 1) = a$$

# Operações t-norma, t-conorma

- Os axiomas de 1 a 4 são necessários para caracterização dos operadores. Os axiomas 5 a 7 são por vezes desejáveis.
  1. Contorno
  2. Monotonicidade
  3. Comutatividade
  4. Associatividade
  5. Continuidade
  6. Sub-Idempotência
  7. Monotonicidade Estrita

$$b \leq d$$
$$\Rightarrow$$
$$I(a, b) \leq I(a, d)$$

# Operações t-norma, t-conorma

- Os axiomas de 1 a 4 são necessários para caracterização dos operadores. Os axiomas 5 a 7 são por vezes desejáveis.
  1. Contorno
  2. Monotonicidade
  3. Comutatividade
  4. Associatividade
  5. Continuidade
  6. Sub-Idempotência
  7. Monotonicidade Estrita

$$I(a,b)=I(b,a)$$

# Operações t-norma, t-conorma

- Os axiomas de 1 a 4 são necessários para caracterização dos operadores. Os axiomas 5 a 7 são por vezes desejáveis.
  1. Contorno
  2. Monotonicidade
  3. Comutatividade
  4. Associatividade
  5. Continuidade
  6. Sub-Idempotência
  7. Monotonicidade Estrita

$$I(a, I(b, d)) = I(I(a, b), d)$$

# Operações t-norma, t-conorma

- Os axiomas de 1 a 4 são necessários para caracterização dos operadores. Os axiomas 5 a 7 são por vezes desejáveis.
  1. Contorno
  2. Monotonicidade
  3. Comutatividade
  4. Associatividade
  5. Continuidade
  6. Sub-Idempotência
  7. Monotonicidade Estrita

*Pequenas  
variações  
no argumento  
não causam  
grandes variações*

# Operações t-norma, t-conorma

- Os axiomas de 1 a 4 são necessários para caracterização dos operadores. Os axiomas 5 a 7 são por vezes desejáveis.
  1. Contorno
  2. Monotonicidade
  3. Comutatividade
  4. Associatividade
  5. Continuidade
  6. Sub-Idempotência
  7. Monotonicidade Estrita

$$I(a, a) \leq a$$

# Operações t-norma, t-conorma

- Os axiomas de 1 a 4 são necessários para caracterização dos operadores. Os axiomas 5 a 7 são por vezes desejáveis.
  1. Contorno
  2. Monotonicidade
  3. Comutatividade
  4. Associatividade
  5. Continuidade
  6. Sub-Idempotência
  7. Monotonicidade Estrita

$$a_1 < a_2 \text{ E } b_1 < b_2 \\ \Rightarrow \\ I(a_1, b_1) < I(a_2, b_2)$$

# Pares t-norma e t-conorma (1/2)

- Padrão:

$$\mathcal{T}(A(x), B(x)) = \min(\mu_A(x), \mu_B(x))$$

$$\mathcal{U}(A(x), B(x)) = \max(\mu_A(x), \mu_B(x))$$

$$\mathcal{C}(A(x)) = 1 - \mu_A(x)$$

- Produto e Soma Algébrica

$$\mathcal{T}(A(x), B(x)) = \mu_A(x) \cdot \mu_B(x)$$

$$\mathcal{U}(A(x), B(x)) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

Única t-norma  
idempotente



# Pares t-norma e t-conorma (2/2)

- **Diferença e Soma Limitada**

$$\mathcal{I}(\mathbf{A}(x), \mathbf{B}(x)) = \max(0, \mu_{\mathbf{A}}(x) + \mu_{\mathbf{B}}(x) - 1)$$

$$\mathcal{U}(\mathbf{A}(x), \mathbf{B}(x)) = \min(1, \mu_{\mathbf{A}}(x) + \mu_{\mathbf{B}}(x))$$

- **Interseção e União Robusta:**

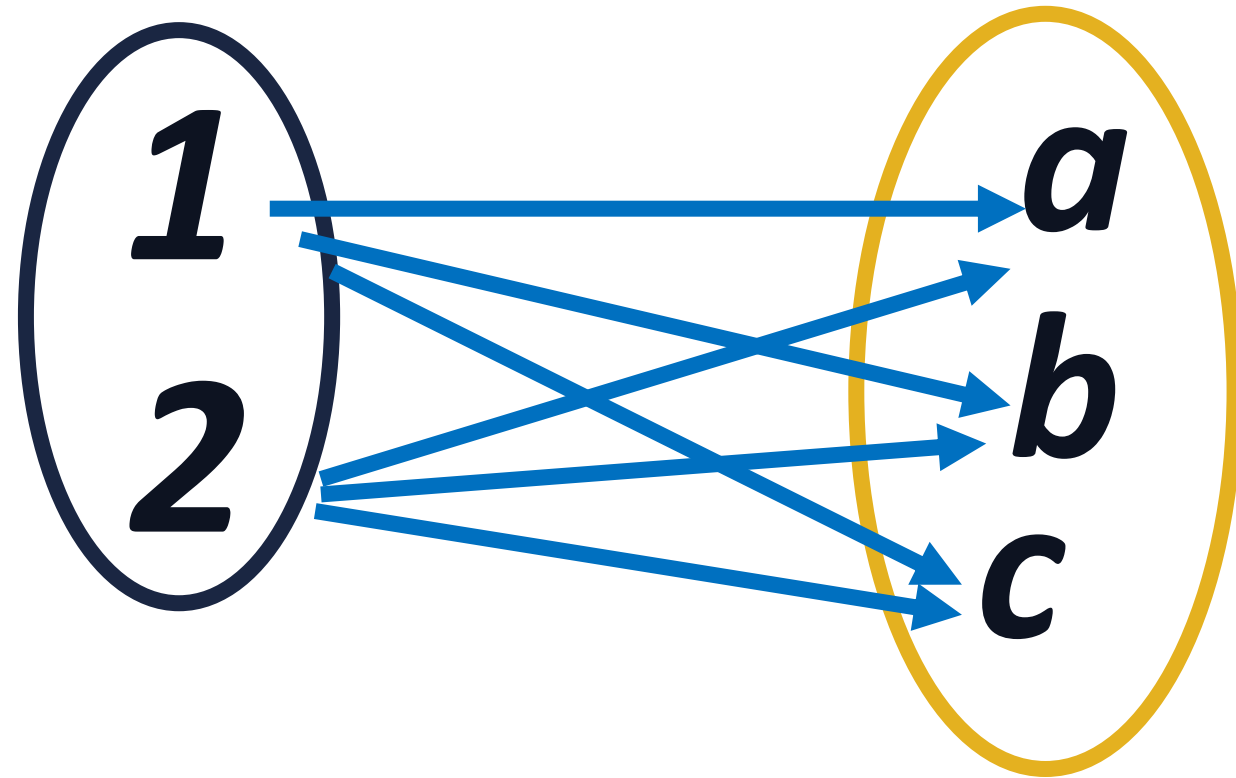
$$\mathcal{I}(\mathbf{A}(x), \mathbf{B}(x)) = \begin{cases} \mu_{\mathbf{A}}(x) & \text{quando } \mu_{\mathbf{B}}(x) = 1 \\ \mu_{\mathbf{B}}(x) & \text{quando } \mu_{\mathbf{A}}(x) = 1 \\ 0, & \text{de outro modo} \end{cases}$$

$$\mathcal{U}(\mathbf{A}(x), \mathbf{B}(x)) = \begin{cases} \mu_{\mathbf{A}}(x) & \text{quando } \mu_{\mathbf{B}}(x) = 0 \\ \mu_{\mathbf{B}}(x) & \text{quando } \mu_{\mathbf{A}}(x) = 0 \\ 1, & \text{de outro modo} \end{cases}$$

# Relações

- Uma relação entre conjuntos nítidos é um subconjunto do produto cartesiano desses conjuntos.
  - $A = \{1, 2\}$
  - $B = \{a, b, c\}$
  - $A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$
  - $A R B = \{(1, a), (1, b), (2, c)\}$
  - Existe uma função característica para a relação, já que ela é um conjunto

# Produto Cartesiano Nítido (único)



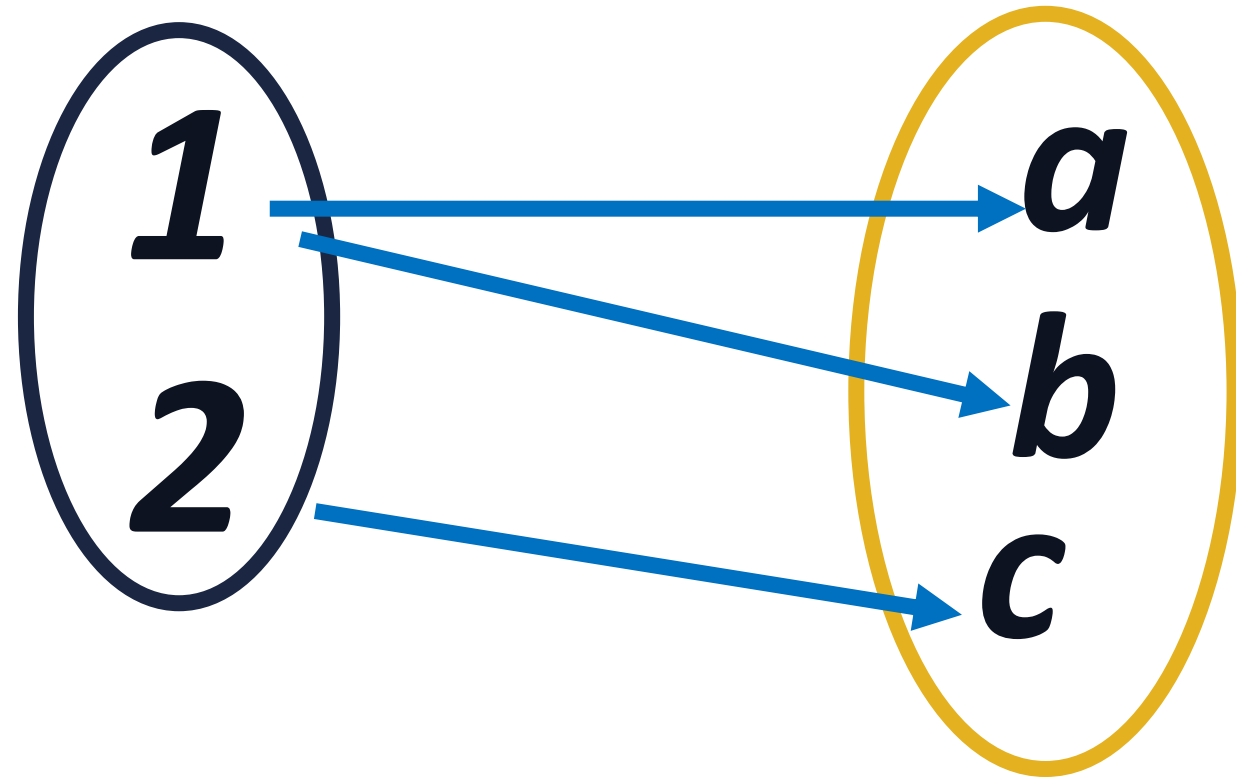
- Todos os pares onde o primeiro elemento é do primeiro conjunto e o segundo elemento é do segundo conjunto

$$A = \{1, 2\}$$

$$B = \{a, b, c\}$$

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$$

# Relação Nítida (um exemplo)



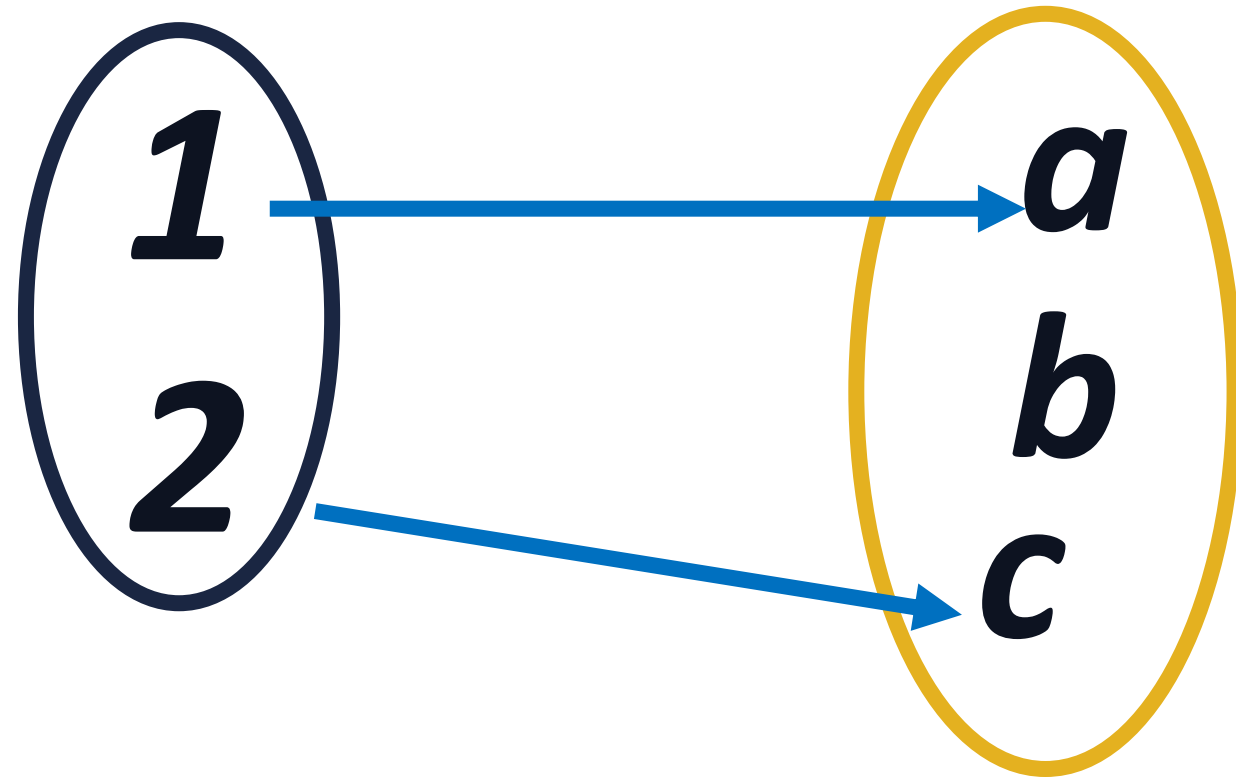
- Uma relação é um subconjunto qualquer do produto cartesiano

$$A = \{1, 2\}$$

$$B = \{a, b, c\}$$

$$R: A \rightarrow B = \{(1, a), (1, b), (2, c)\}$$

# Função Nítida (um exemplo)



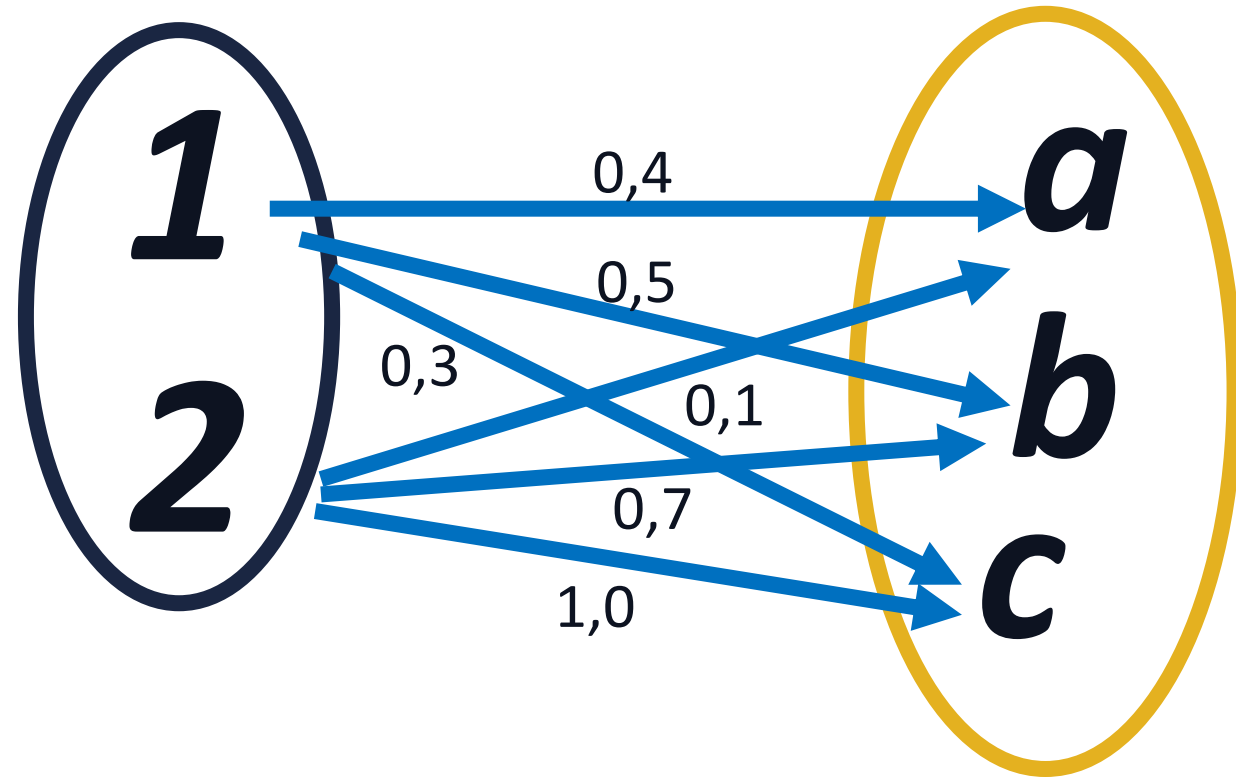
- Em uma função, um elemento do domínio só pode ser mapeado em um único elemento da imagem

$$A = \{1, 2\}$$

$$B = \{a, b, c\}$$

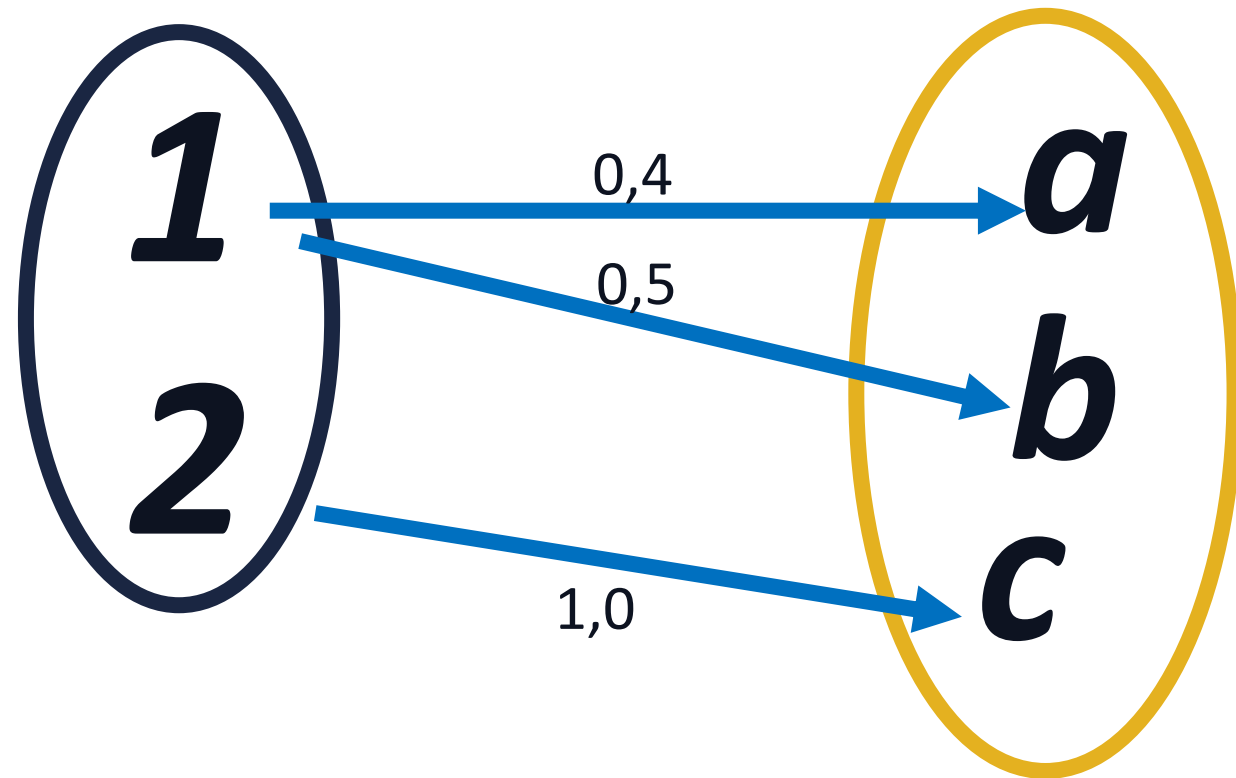
$$F: A \rightarrow B = \{(1, a), (2, c)\}$$

# Produto Cartesiano Fuzzy (um exemplo)

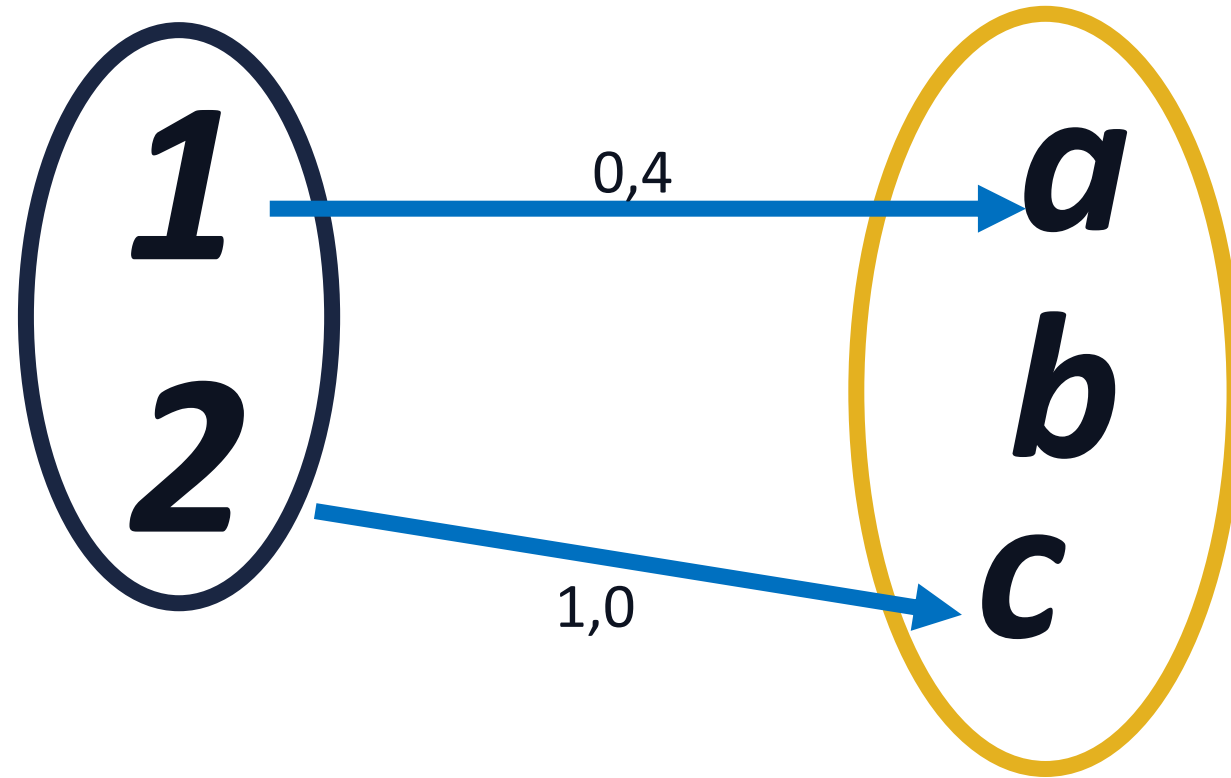


- O grau de pertinência pode variar
  - E até ser 0 (zero)

# Relação Fuzzy (um exemplo)



# Função Fuzzy (um exemplo)



- Algumas interpretações de funções fuzzy permitem que um elemento do domínio seja mapeado em mais de um elemento da imagem!



# Princípio da Extensão

# O Problema da Extensão

- Dada uma fórmula  $f(x)$  e um conjunto fuzzy  $\mu_A(x)$ , como calcular a função de pertinência de  $f(A)$ ?
- $f_\alpha(A) = f(A(\alpha))$ .

# Princípio da Extensão

- Os sistemas e modelos matemáticos estão baseados em relações que mapeiam um conjunto de variáveis de um universo em outro.

- $f: X \rightarrow Y$ , onde,

$X$  e  $Y$  representam dois universos quaisquer.

- Seja  $x \in X$  e  $y = f(x)$ ,  $y \in Y$ . A função característica que descreve a relação  $R$  (nítida) é

$$\chi_R(x, y) = \begin{cases} 1, & y = f(x) \\ 0, & y \neq f(x) \end{cases}$$

# Princípio da Extensão

- Devemos agora considerar que tenhamos subconjuntos de  $X$  que serão mapeados em subconjuntos de  $Y$ , dessa forma:

$$f: P(X) \rightarrow P(Y),$$

onde  $P(\cdot)$  representa o conjunto potência. Um subconjunto nebuloso  $\mathbf{A} \subset X$  é mapeado num subconjunto  $\mathbf{B} \subset Y$  da forma:

$$\mathbf{B} = f(\mathbf{A}) = \{y \mid \forall x \in \mathbf{A}, y = f(x)\}.$$

# Princípio da Extensão

Em termos mais gerais, devemos supor que mais de um elemento de  $Y$  seja mapeado a partir de um mesmo elemento de  $X$ , dessa forma, a função característica de  $\mathbf{B}$  seria,

$$\chi_{\mathbf{B}}(y) = \bigvee_{y = f(x)} \chi_{\mathbf{A}}(x)$$

onde  $\bigvee_{y = f(x)}$  representa o valor máximo para todos os valores de  $x$  que levem a  $y$ . A função característica em conjuntos precisos apenas indica se um valor pertence ou não a um conjunto.

# Princípio da Extensão

Zadeh(1975) elaborou uma extensão para  $\chi_B(y)$  de forma a lidar com valores intermediários de pertinência:

$$\mu_B(y) = \bigvee_{y = f(x)} \mu_A(x) = \max_{y = f(x)} \mu_A(x) \text{ onde,}$$

$x \in \mathbf{A}$  e  $y \in \mathbf{B}$  são agora elementos de conjuntos nebulosos e a função  $f: F(X) \rightarrow F(Y)$ , de conjuntos nebulosos de  $X$  em conjuntos nebulosos de  $Y$ .

Supondo ainda que a função  $f$  possa ser aplicada sobre um conjunto de  $n$  universos  $x_1, x_2, \dots, x_n$ , tal que

$$f: F(x_1, x_2, \dots, x_n) \rightarrow F(Y), \text{ onde } y = f(x_1, x_2, \dots, x_n)$$

# Princípio da Extensão

- A função de pertinência  $\mu_B(y)$  de  $B = f(A_1, A_2, \dots, A_n)$  será dada por:

$$\mu_B(y) = \max_{y = f(x_1, x_2, \dots, x_n)} \{ \min [\mu_{A_1}, \mu_{A_2}, \dots, \mu_{A_n}] \}$$

- O grau de pertinência de um valor do conjunto resposta ao conjunto resultado é o grau de pertinência máximo entre todas as aplicações da função que chegam aquele resultado
- O grau de pertinência de valor de uma aplicação da função é o grau de pertinência mínimo dos argumentos da função

# Exemplo

- Conjuntos fuzzy finitos para facilitar o exemplo
- $A = 3$
- $B = 2$
- $A + B = 5$
- Como é um “+” Fuzzy?
- Pelo princípio da extensão sabemos que existe

- $\tilde{A}$  é um  $A$  fuzzy
- $\tilde{B}$  é um  $B$  fuzzy

$$\tilde{A} = \frac{.1}{1} + \frac{.2}{2} + \frac{1}{3} + \frac{.1}{4}$$

$$\tilde{B} = \frac{.3}{1} + \frac{1}{2} + \frac{.5}{3}$$



# Exemplo – Números A e B

A	$\mu(A)$	B	$\mu(B)$	A+B	$\mu(A+B)$
1		1			
1		2			
1		3			
2		1			
2		2			
2		3			
3		1			
3		2			
3		3			
4		1			
4		2			
4		3			

# Exemplo – Graus de Pertinência para $\tilde{A}$ e $\tilde{B}$

A	$\mu(A)$	B	$\mu(B)$	A+B	$\mu(A+B)$
1	.1	1	.3		
1	.1	2	1		
1	.1	3	.5		
2	.2	1	.3		
2	.2	2	1		
2	.2	3	.5		
3	1	1	.3		
3	1	2	1		
3	1	3	.5		
4	.1	1	.3		
4	.1	2	1		
4	.1	3	.5		

# Exemplo – Soma dos Números

A	$\mu(A)$	B	$\mu(B)$	A+B	$\mu(A+B)$
1	.1	1	.3	2	
1	.1	2	1	3	
1	.1	3	.5	4	
2	.2	1	.3	3	
2	.2	2	1	4	
2	.2	3	.5	5	
3	1	1	.3	4	
3	1	2	1	5	
3	1	3	.5	6	
4	.1	1	.3	5	
4	.1	2	1	6	
4	.1	3	.5	7	

# Exemplo – Grau de Pertinência das Somas Usam o Mínimo

A	$\mu(A)$	B	$\mu(B)$	A+B	$\mu(A+B)$
1	.1	1	.3	2	.1
1	.1	2	1	3	.1
1	.1	3	.5	4	.1
2	.2	1	.3	3	.2
2	.2	2	1	4	.2
2	.2	3	.5	5	.2
3	1	1	.3	4	.3
3	1	2	1	5	1
3	1	3	.5	6	.5
4	.1	1	.3	5	.1
4	.1	2	1	6	.1
4	.1	3	.5	7	.1

# Exemplo – Múltiplos Graus de Pertinência para cada Resultado Possível

A	$\mu(A)$	B	$\mu(B)$	A+B	$\mu(A+B)$
1	.1	1	.3	2	.1
1	.1	2	1	3	.1
1	.1	3	.5	4	.1
2	.2	1	.3	3	.2
2	.2	2	1	4	.2
2	.2	3	.5	5	.2
3	1	1	.3	4	.3
3	1	2	1	5	1
3	1	3	.5	6	.5
4	.1	1	.3	5	.1
4	.1	2	1	6	.1
4	.1	3	.5	7	.1

# Exemplo – Escolhemos o Máximo para o Grau de Pertinência de Cada Resultado

A	$\mu(A)$	B	$\mu(B)$	A+B	$\mu(A+B)$
1	.1	1	.3	2	.1
1	.1	2	1	3	.1
1	.1	3	.5	4	.1
2	.2	1	.3	3	.2
2	.2	2	1	4	.2
2	.2	3	.5	5	.2
3	1	1	.3	4	.3
3	1	2	1	5	1
3	1	3	.5	6	.5
4	.1	1	.3	5	.1
4	.1	2	1	6	.1
4	.1	3	.5	7	.1

# Exemplo – Resultado Final

A	$\mu(A)$	B	$\mu(B)$	A+B	$\mu(A+B)$
1	.1	1	.3	2	.1
1	.1	2	1	3	.1
1	.1	3	.5	4	.1
2	.2	1	.3	3	.2
2	.2	2	1	4	.2
2	.2	3	.5	5	.2
3	1	1	.3	4	.3
3	1	2	1	5	.1
3	1	3	.5	6	.5
4	.1	1	.3	5	.1
4	.1	2	1	6	.1
4	.1	3	.5	7	.1

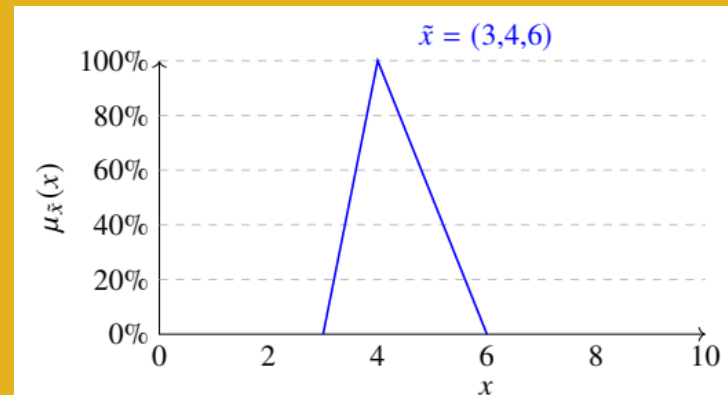
$$A + B = \frac{.1}{2} + \frac{.2}{3} + \frac{.3}{4} + \frac{1}{5} + \frac{.5}{6} + \frac{.1}{1}$$

# Atenção

- O princípio da extensão é uma prova construtiva que garante que há uma função fuzzy para cada função nítida, mas ele não define uma única função fuzzy ou exige que ela seja usada
- O designer do sistema pode projetar outra função fuzzy que atenda as suas necessidades e mantenha as propriedades da função original



# Aritmética Nebulosa



# Números Fuzzy

- Quando estamos falando sobre números no mundo real, muitas vezes não sabemos com precisão o valor desse número
  - O carro custa cerca de R\$15000,00.
  - A casa é daqui a uns 100m.
  - Compre uns 2 quilos de carne

# Valores Aceitáveis

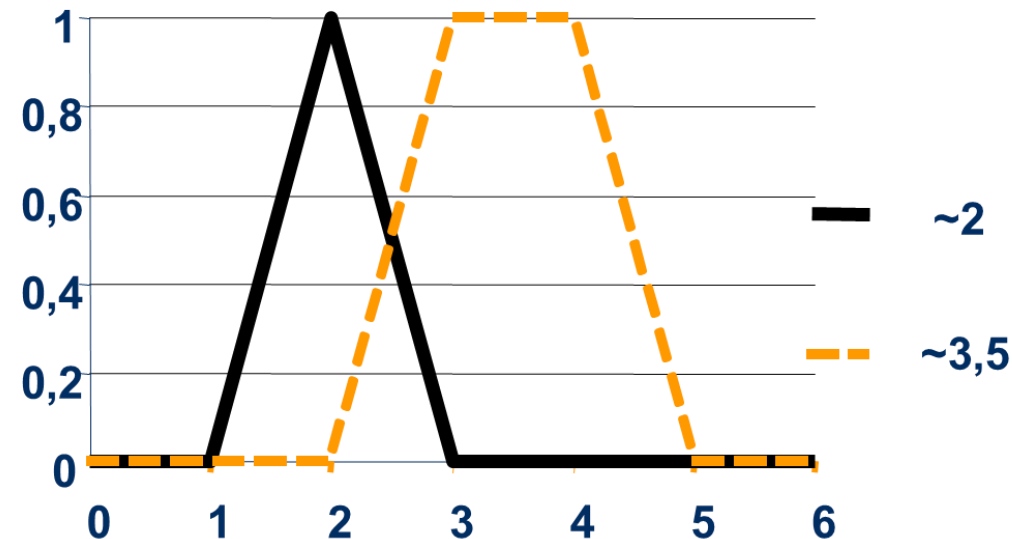
- Que valores são aceitáveis quando utilizamos esses termos?
  - Carro: entre 14000 e 16000? Entre 14900 e 15500?
  - Distância: entre 50 e 150 m?
  - Carne: entre 1,9 e 2,1 quilos?

# Números Fuzzy

- Estamos então trabalhando com números que podem ser descritos de uma forma “Fuzzy”
- Não são distribuições de probabilidade!
  - Apesar de podermos também modelar alguns casos dessa forma
    - quantos quilos temos aqui?

# Números Fuzzy

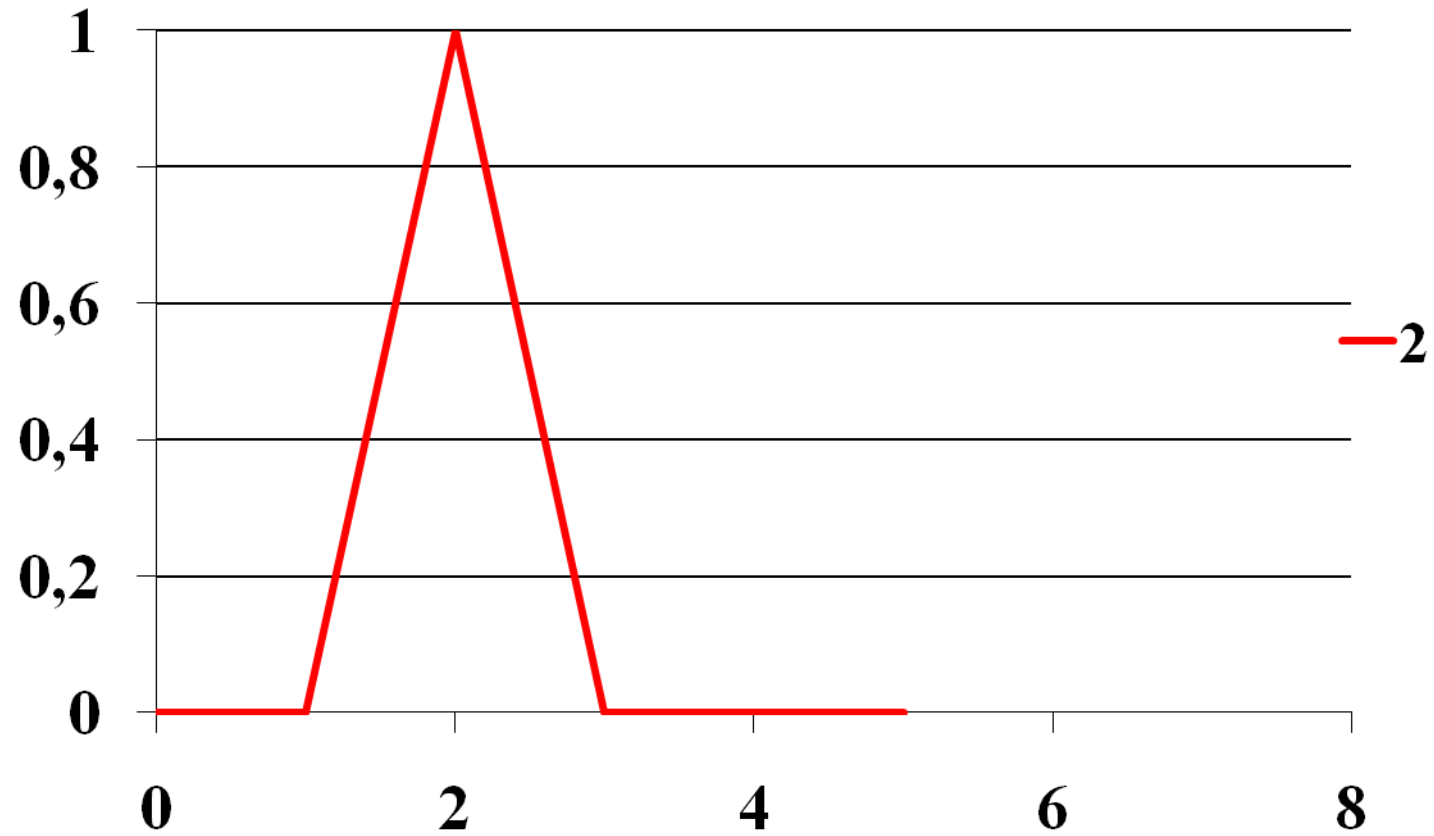
- São conjuntos Fuzzy, representando, de alguma forma que interesse a solução de um problema, uma incerteza em um valor numérico



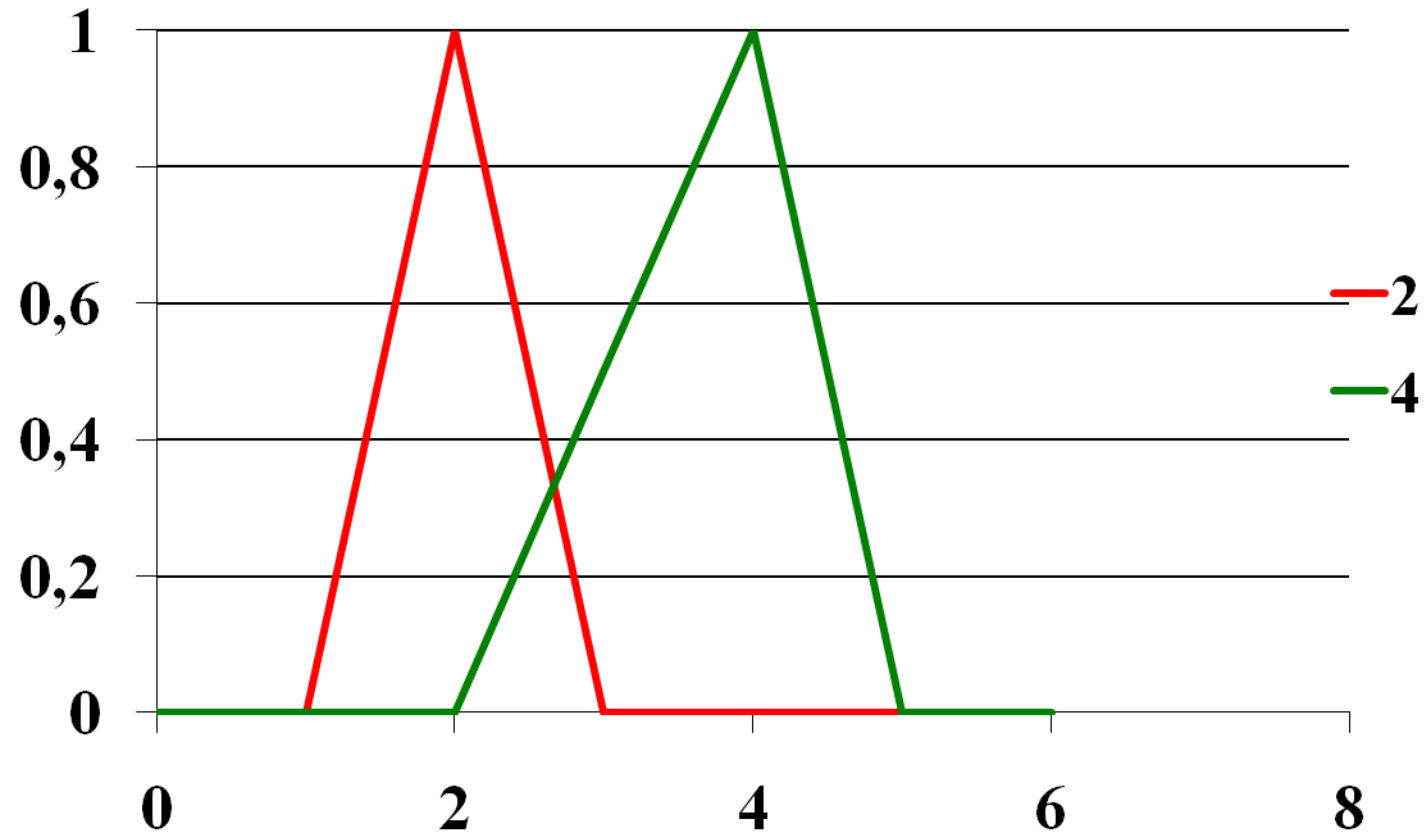
# Número Nebuloso

- Normal (altura = 1)
- Todo corte alfa precisa ser um intervalo fechado
- O suporte precisa ser limitado

# Número Fuzzy



# Número Fuzzy





# Operação Fuzzy

- Regra Simples (forma mais simples de fazer) para números triangulares
  - faça a operação normal com os valores do ápice (que são os valores nítidos)
  - some o tamanho das duas bases
  - divida por dois
  - crie o triângulo estendendo o número Fuzzy para cada lado pelo último valor

*Não segue o princípio da extensão*

*Não é um problema*

# Aritmética Fuzzy

- Também pode ser baseada na aritmética intervalar
- Seja um número Fuzzy triangular na forma
  - $(e, c, d)$
- Onde
  - $e$  é o valor mais à esquerda
  - $c$  é o valor central
  - $d$  é o valor mais à direita

# Operações Intervalares

- **Soma intervalar**

- $(e1, c1, d1) + (e2, c2, d2) = (e1+e2, c1+c2, d1+d2)$
- $(1, 2, 3) + (2, 3, 4) = (3, 5, 7)$

- **Pseudo inverso aditivo intervalar**

- $-(e1, c1, d1) = (-d1, -c1, -e1)$
- $-(2, 3, 4) = (-4, -3, -2)$

- **Subtração intervalar**

- $(e1, c1, d1) - (e2, c2, d2) = (e1, c1, d1) + (- (e2, c2, d2)) =$
- $(e1-d2, c1-c2, d1-e2)$
- $(1, 2, 3) - (2, 3, 4) = (-3, -1, 1)$

# Operações Intervalares

- **Produto Intervalar**

- $(e1, c1, d1) \times (e2, c2, d2) =$
- $(\min(e1 \times e2, e1 \times d2, d1 \times e2, d1 \times d2), c1 \times c2, \max(e1 \times e2, e1 \times d2, d1 \times e2, d1 \times d2))$
- $(1, 2, 3) \times (2, 3, 4) = (2, 6, 12)$

- **Pseudo Inverso Multiplicativo Intervalar**

- $(e1, c1, d1)^{-1} = (d1^{-1}, c1^{-1}, e1^{-1})$
- $(2, 3, 4)^{-1} = (0.25, 0.33, 0.5)$

- **Divisão Intervalar**

- $(e1, c1, d1) \div (e2, c2, d2) = (e1, c1, d1) \times (e2, c2, d2)^{-1} =$
- $(\min(e1 \div e2, e1 \div d2, d1 \div e2, d1 \div d2), c1 \div c2, \max(e1 \div e2, e1 \div d2, d1 \div e2, d1 \div d2))$
- $(1, 2, 3) \div (2, 3, 4) = (0.25, 0.66, 1.5)$

# Exemplos

Número $\tilde{n}$	$n_e$	$n_c$	$n_d$
$\tilde{x}$	1	2	3
$\tilde{y}$	2	3	4
$\tilde{x} + \tilde{y}$	3	5	7
$-\tilde{y}$	-4	-3	-2
$\tilde{x} - \tilde{y}$	-3	-1	1
$\tilde{x} \times \tilde{y}$	2	6	12
$\tilde{y}^{-1}$	0,25	0,33	0,5
$\tilde{x} \div \tilde{y}$	0,25	0,66	1,5

# Algoritmo DSW

- Igual para soma e subtração
- Mais correto para Multiplicação e Divisão

**Entrada:**  $\tilde{A}, \tilde{B}, n$

**Saída:**  $\tilde{C}$

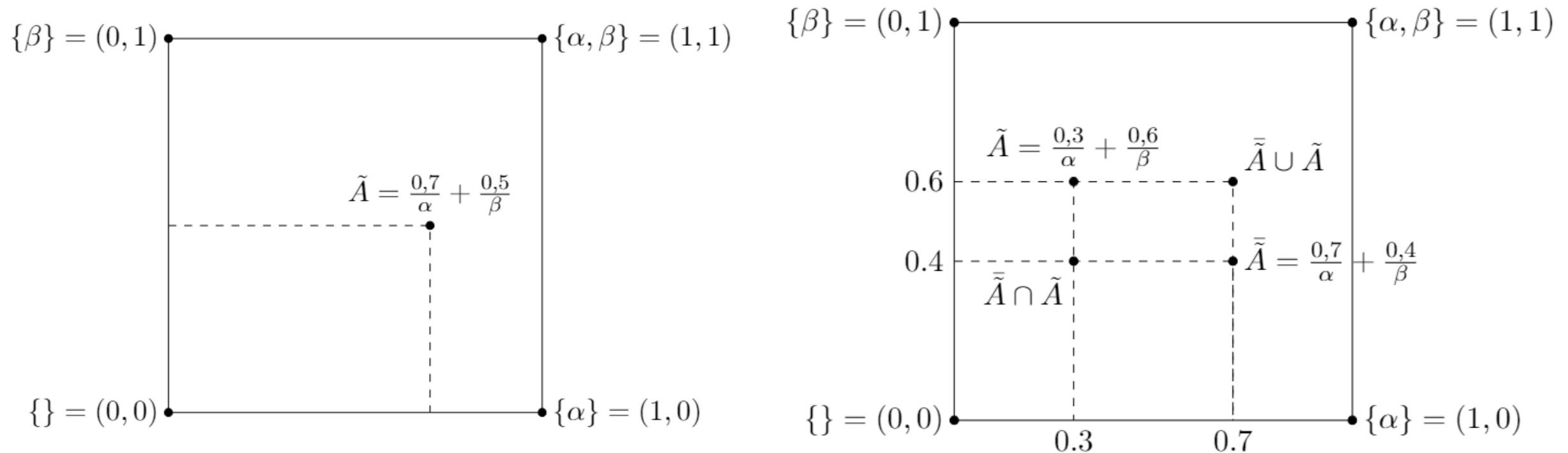
```
1  $S_0 \leftarrow \emptyset$ ;  
2 para  $\alpha \leftarrow 0$  até 1 com passo  $1/n$  faça  
3   Calcule  $\tilde{A}_\alpha$ ;  
4   Calcule  $\tilde{B}_\alpha$ ;  
   /* Usando matemática intervalar */  
5   Calcule  $\tilde{C}_\alpha = \tilde{A}_\alpha \times \tilde{B}_\alpha$ ;  
6    $S \leftarrow S \cup \{\tilde{C}_\alpha\}$   
7 fim  
8 Reconstrua  $\tilde{C}$  a partir dos cortes- $\alpha$  em  $S$ 
```

**Algoritmo 3.1:** Algoritmo DSW, adaptado de (Dong, Shah e Wongt, 1985)

**Quão Nebuloso é um  
Sistema Nebuloso?**

# Entropia

- Quão Fuzzy é um conjunto Fuzzy?
- Pense em um modelo geométrico





# Entropia

- Os vértices são não-Fuzzy, logo tem entropia zero
- O centro tem entropia máxima
- A função de entropia deve variar
- $E(A) = \frac{\text{menor distância a um vértice}}{\text{maior distância a um vértice}}$

# Entropia

$$E(A) = \frac{\ell^1(A, A_{\text{próximo}})}{\ell^1(A, A_{\text{distante}})}$$

$$E(A) = \frac{M(A \cap A^c)}{M(A \cup A^c)}$$

$$E(A) = S(A \cup A^c, A \cap A^c)$$

# Lógica Nebulosa

# Lógica e Lógica Fuzzy

- Lógica é a ciência dos princípios formais normativos do raciocínio (Webster)
- A Lógica Fuzzy se preocupa com os princípios do raciocínio aproximado, com o caso preciso visto como um caso limite
- Lógica Fuzzy objetiva modelar modos de raciocínio imprecisos, que têm um papel importante na forma humana de pensar

# Considerações Iniciais

- Apesar de poder ser vista como uma extensão da lógica multivalorada, seus usos e objetivos são diferentes
- O rigor matemático não tem um papel tão importante como nos sistemas lógicos clássicos
- Tudo é uma questão de grau

# Fuzzy x Clássico

- Clássico
  - Falso x verdadeiro
  - ou intervalo (lógicas multivaloradas), finito ou infinito (conjuntos de números)
- Fuzzy
  - Subconjuntos Fuzzy do intervalo! (conjuntos Fuzzy de números, cada número pertence com um grau a esse subconjunto)

# Operadores

$\wedge$	F	V
F	F	F
V	F	V

$\Rightarrow$	F	V
F	V	V
V	F	V

$\neg$	
F	V
V	F

$\vee$	F	V
F	F	V
V	V	V

$\Leftrightarrow$	F	V
F	V	F
V	F	V

# Modus Ponens

$P \Rightarrow Q$

$P$

---

$Q$



# Modus Tollens

$$P \Rightarrow Q$$
$$\neg Q$$

---

$$\neg P$$

# Operações Fuzzy

- Conjunção (E)
  - $\mu(A \text{ E } B) = \text{Mín}(\mu(A), \mu(B))$
- Disjunção (OU)
  - $\mu(A \text{ OU } B) = \text{Máx}(\mu(A), \mu(B))$
- Negação (NÃO)
  - $\mu(\text{NÃO } A) = 1 - \mu(A)$

**Existem outros modelos mais complexos**

# Modus Ponens Generalizado

$P \Rightarrow Q$

$P$        $\sim$

---

? (Q)       $\sim$

# P $\Rightarrow$ Q

- É uma relação
- Podemos entender de várias formas
  - $(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q)$
  - $(P \Rightarrow Q) \Leftrightarrow (\neg P \vee (P \wedge Q))$
- Cada forma implica na escolha de uma função

# A $\Rightarrow$ B

Tabela 4.14: Várias funções que podem ser usadas para calcular a implicação fuzzy (Klir e Yuan, 1995)

Nome	Função	Classe
Kleene-Dienes	$\max(\mu_B(y), (1 - \mu_A(x)))$	S,QL
Reichenbach	$1 - \mu_A(x) + \mu_A(x) \times \mu_B(y)$	S
Łukasiewicz	$\min(1, 1 - \mu_A(x) + \mu_B(y))$	S,R
Gödel	$\begin{cases} 1, & \text{if } \mu_A(x) \leq \mu_B(y) \\ \mu_B(y), & \text{em outro caso} \end{cases}$	R
Gaines	$\begin{cases} 1, & \text{if } \mu_A(x) \leq \mu_B(y) \\ \frac{\mu_B(y)}{\mu_A(x)}, & \text{em outro caso} \end{cases}$	R
Estrita Padrão	$\begin{cases} 1, & \text{if } \mu_A(x) \leq \mu_B(y) \\ 0, & \text{em outro caso} \end{cases}$	R
Mamdani	$\min(\mu_A(x), \mu_B(y))$	T
Larsen	$\mu_A(x) \times \mu_B(y)$	T
Zadeh (inicial)	$\max(\min(\mu_A(x), \mu_B(y)), (1 - \mu_A(x)))$	T
Yager	$\mu_B(y)^{\mu_A(x)}$	

# Implicação Kleene-Dies

Tabela 4.6: Tabela verdade para  $A \Rightarrow B = \bar{A} \vee B$  da implicação

$A$	$B$	$A \Rightarrow B$	$\bar{A}$	$\bar{A} \vee B$	$A \Rightarrow B \Leftrightarrow \bar{A} \vee B$
V	V	V	F	V	V
V	F	F	F	F	V
F	V	V	V	V	V
F	F	V	V	V	V

## Programa 4.1: Realizando operações lógicas em SciKit-Fuzzy

```
1 import numpy as np
2 import skfuzzy as fuzz
3 import matplotlib.pyplot as plt
4
5
6 x = np.arange(101)
7 num1 = fuzz.trimf(x,[10,40,70])
8 num2 = fuzz.trimf(x,[30,60,90])
9 x1, fand = fuzz.fuzzy_and(x, num1, x, num2)
10 x2, flor = fuzz.fuzzy_or(x, num1, x, num2)
11 fnot = fuzz.fuzzy_not(num1)
12
13 gnum1, = plt.plot(x, num1)
14 gnum2, = plt.plot(x, num2, linestyle=":")
15 plt.legend([gnum1, gnum2], ["Número_1", "Número_2"])
16 plt.show()
17
18 gand, = plt.plot(x1, fand)
19 plt.legend([gand], ["E_Fuzzy"])
20 plt.ylim(0,1)
21 plt.show()
22
23 gor, = plt.plot(x2, flor)
24 plt.legend([gor], ["Ou_Fuzzy"])
25 plt.show()
26
27 gnot, = plt.plot(x, fnot)
28 plt.legend([gnot], ["Negação_Fuzzy"])
29 plt.show()
```

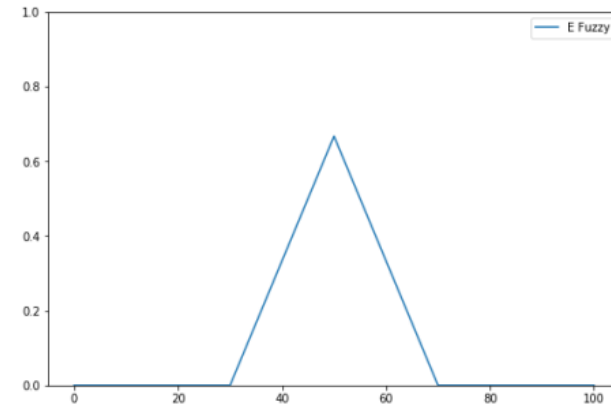
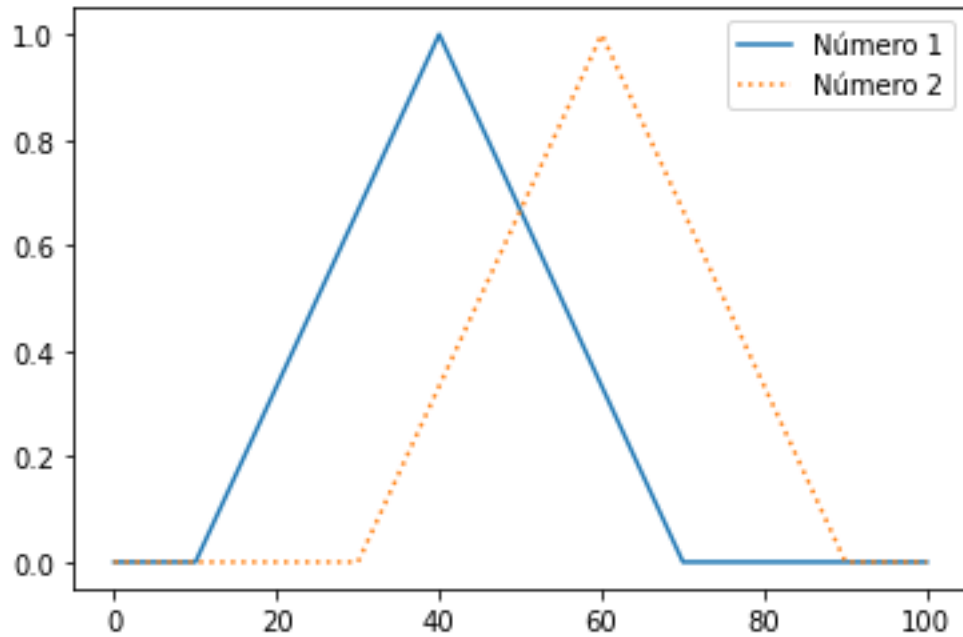


Figura 4.1: Gráfico da função **e**, gerado pelo Programa 4.1.

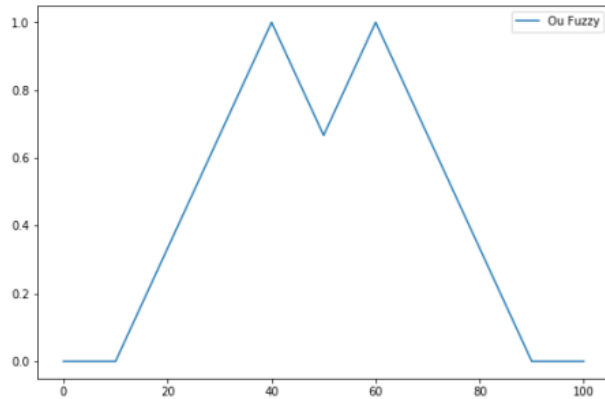


Figura 4.2: Gráfico da função **ou**, gerado pelo Programa 4.1. Não é mantida a convexidade.

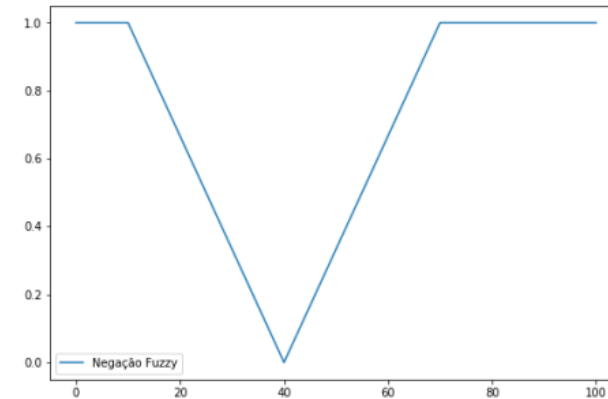


Figura 4.3: Gráfico da função de **negação**, gerado pelo Programa 4.1. Não é mantida a convexidade.



## Programa 4.2: Realizando o modus ponens em SciKit-Fuzzy

```
1 import numpy as np
2 import skfuzzy as fuzz
3 import matplotlib.pyplot as plt
4
5
6 x = np.arange(0,10,step=0.1)
7 a = fuzz.trimf(x,[1,2,3])
8 b = fuzz.trimf(x,[3,6,7])
9 alinhado = fuzz.trimf(x,[0,2,2.5])
10 r,mp = fuzz.modus_ponens(a,b,alinhado)
11
12 gnum1, = plt.plot(x,a)
13 gnum2, = plt.plot(x,b,linestyle=":")
14 gnum3, = plt.plot(x,alinhado,linestyle="--")
15 gnum4, = plt.plot(x,mp)
16 plt.legend([gnum1,gnum2,gnum3,gnum4],["A","B","A'","B'"])
17 plt.show()
```

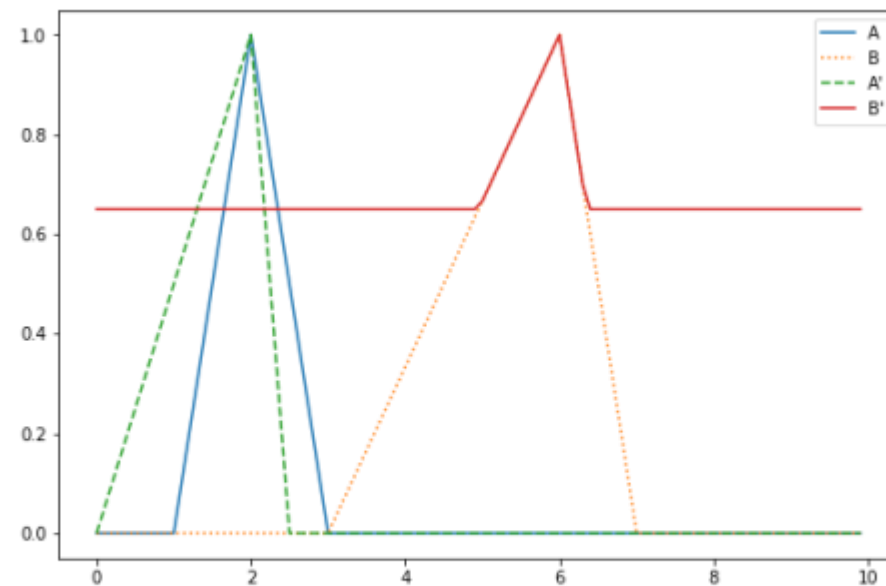
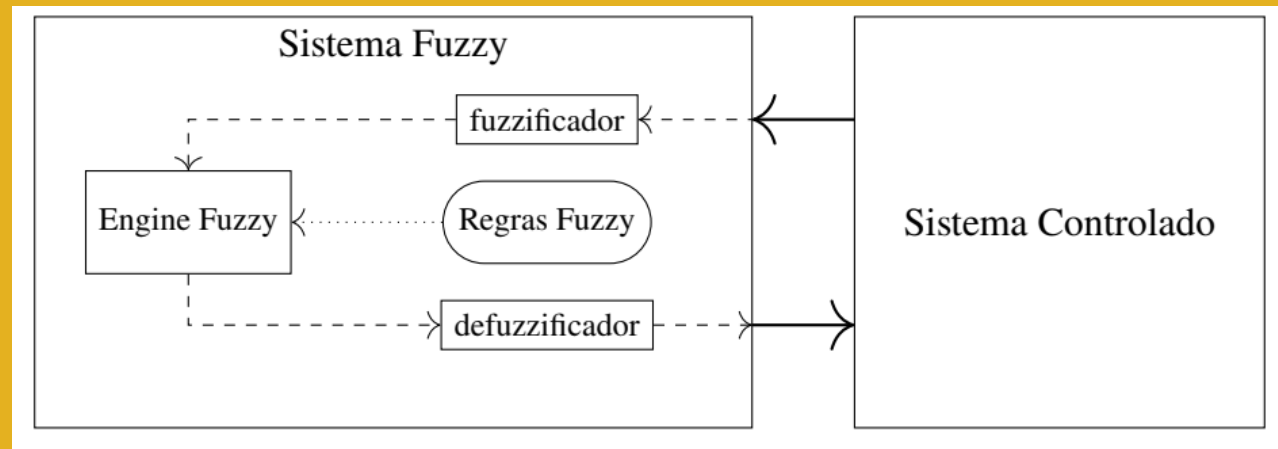


Figura 4.4: Gráfico da função **modus\_ponens**, gerado pelo Programa 4.2.

# Sistemas de Controle Fuzzy



# Regras Fuzzy

- São regras baseadas em proposições Fuzzy
  - SE X É ALTO ENTÃO Y É BAIXO
    - onde ALTO e BAIXO são descritos por funções de pertinência
- Variáveis e termos linguísticos

# Variáveis linguísticas

- Um conceito que pode assumir um conjunto de valores.
- Exemplo: tamanho de código.

# Termos linguísticos

- Valores linguísticos (palavras) que representam um conjunto nebuloso definido sobre a variável linguística. Exemplo: tamanho pode ser mínimo, pequeno, médio e grande. São, na verdade, funções da medida no intervalo  $[0,1]$ .

# Exemplo de regras

## Regra 1:

Se a temperatura está fria, o motor pára.

## Regra 2:

Se a temperatura está fresca, a velocidade do motor diminui.

## Regra 3:

Se a temperatura está normal, a velocidade do motor é média.

## Regra 4:

Se a temperatura está morna, a velocidade do motor é rápida.

## Regra 5:

Se a temperatura está quente, a velocidade do motor é máxima

# Controle Fuzzy

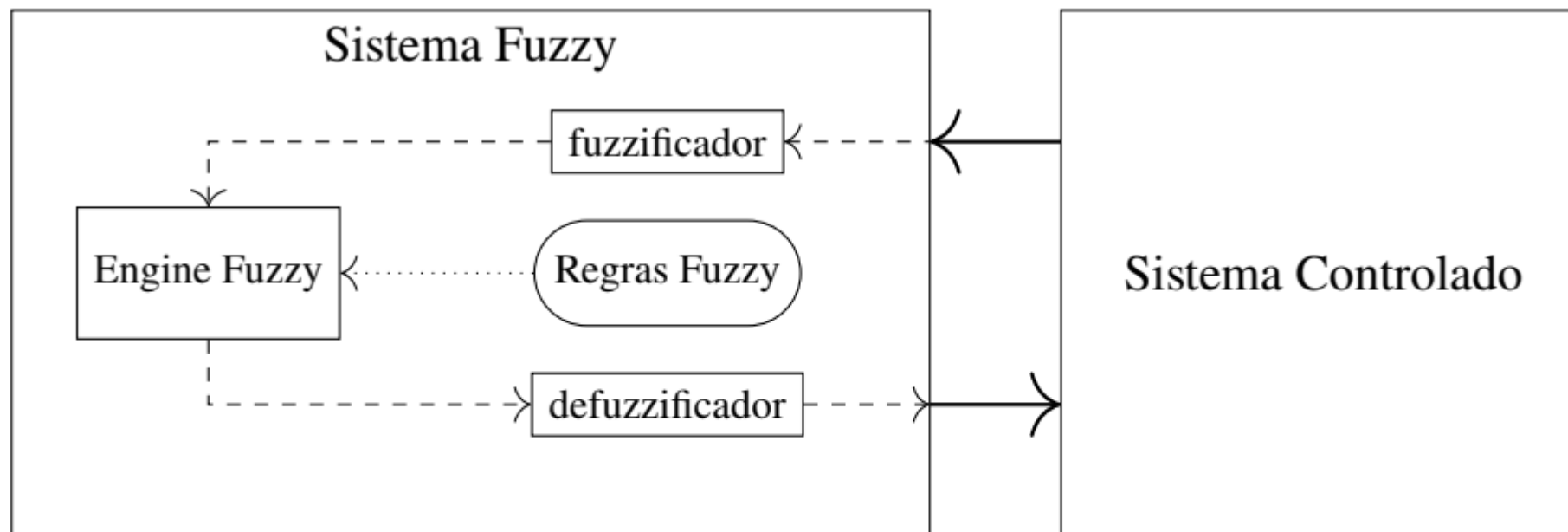


Figura 5.1: Sistema de controle fuzzy

# Um SCF é um Conjunto de Aproximações

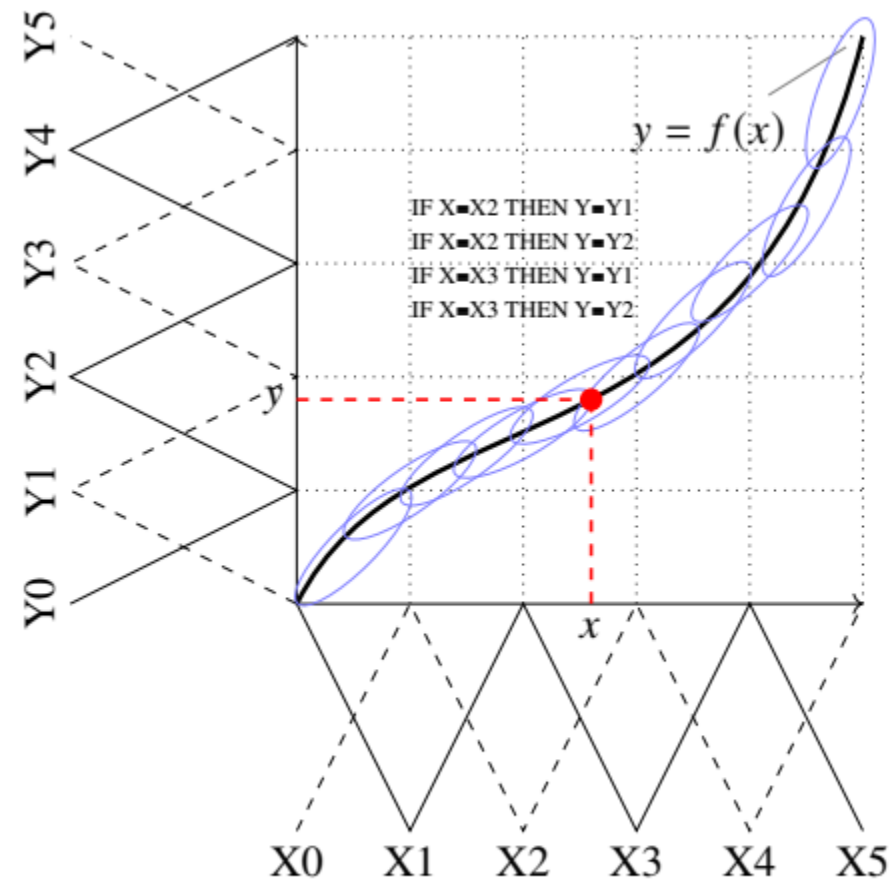
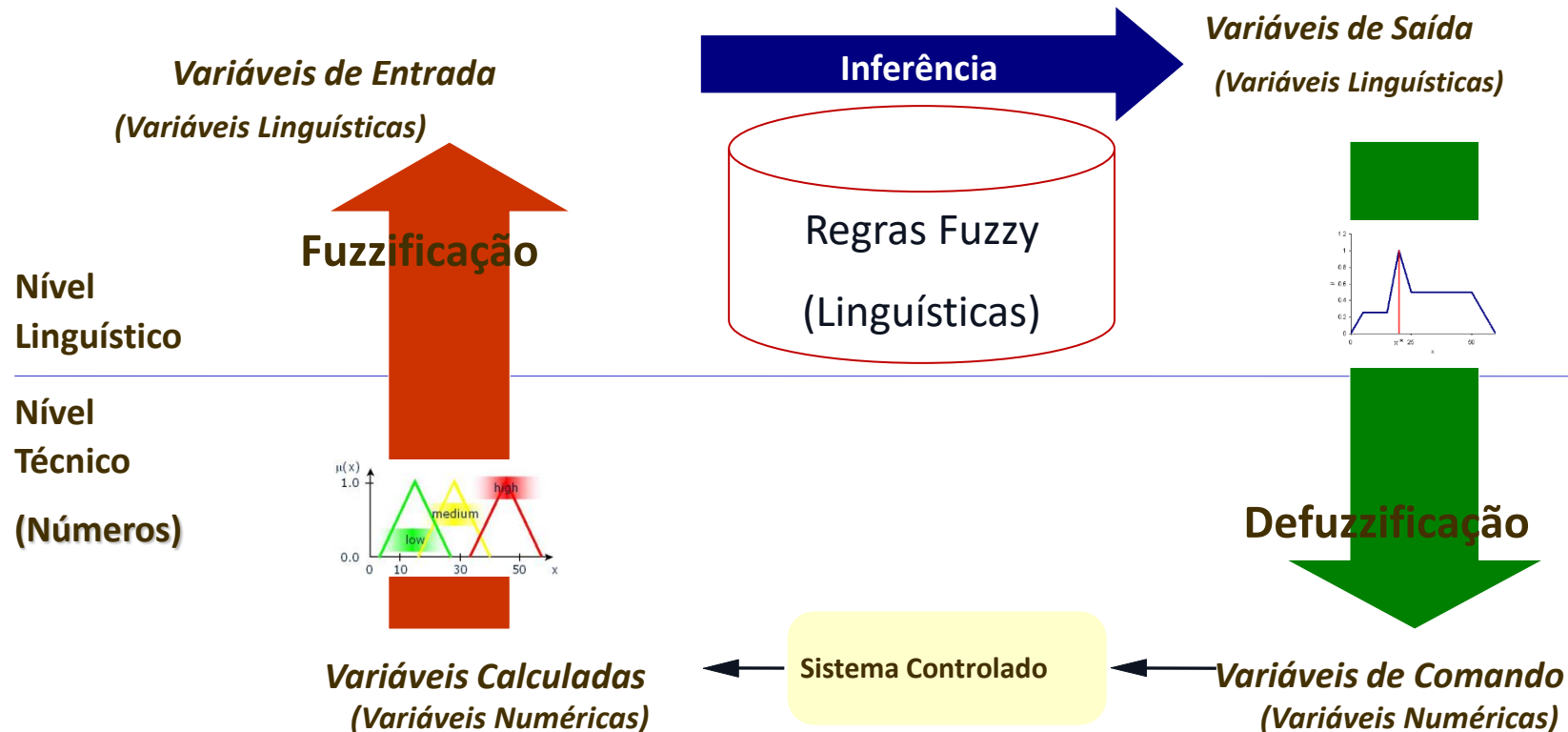


Figura 5.2: Regras fuzzy funcionam como especificação de pedaços das funções sendo agregadas (Cox, 1995)



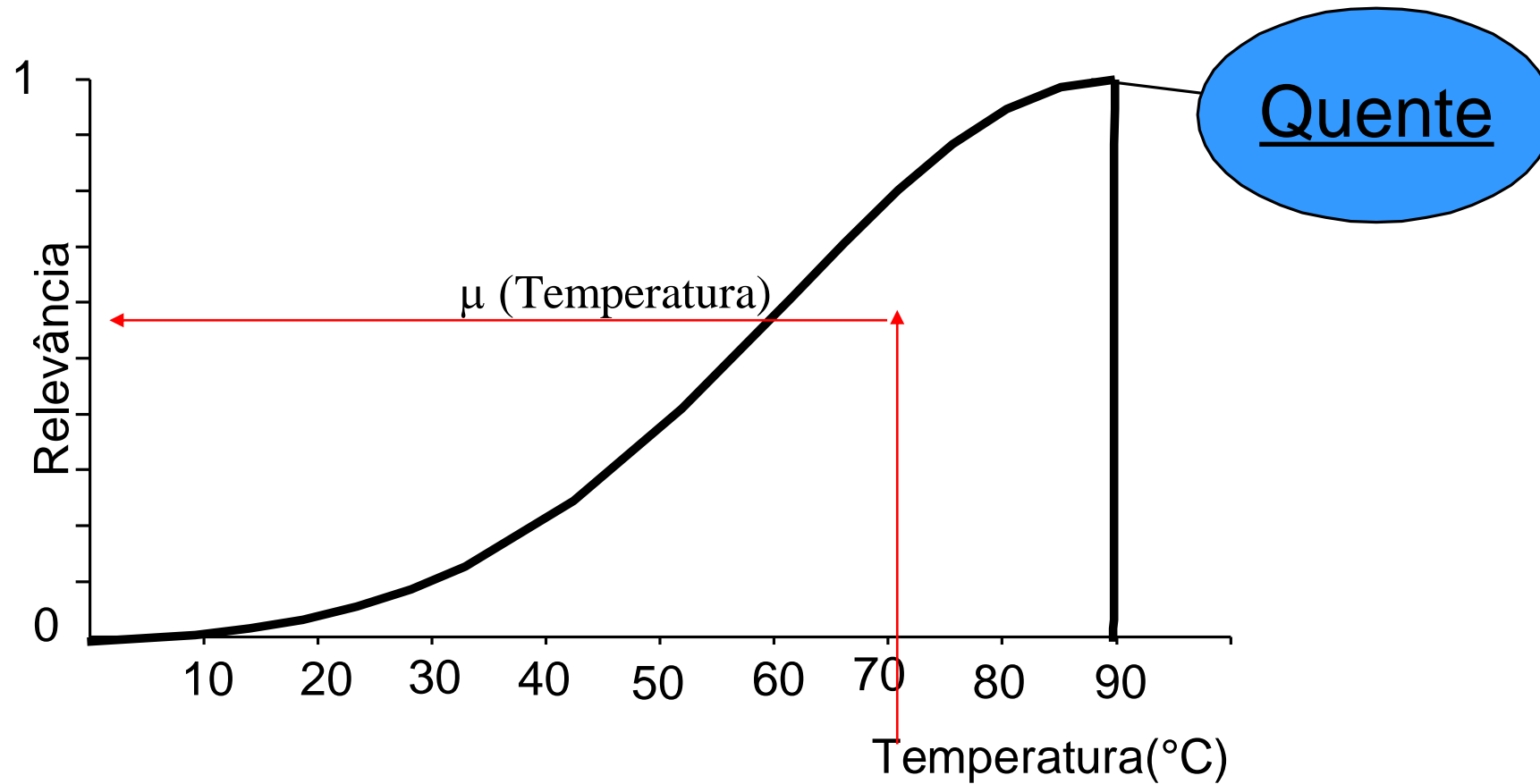
# Ciclo Fuzzy



# Fuzzificação

- Transformação do valor de entrada nos valores correspondentes em cada conjunto Fuzzy (termo linguístico)
- Uma forma básica: usar a função de pertinência

# Fuzzificação



# Inferência

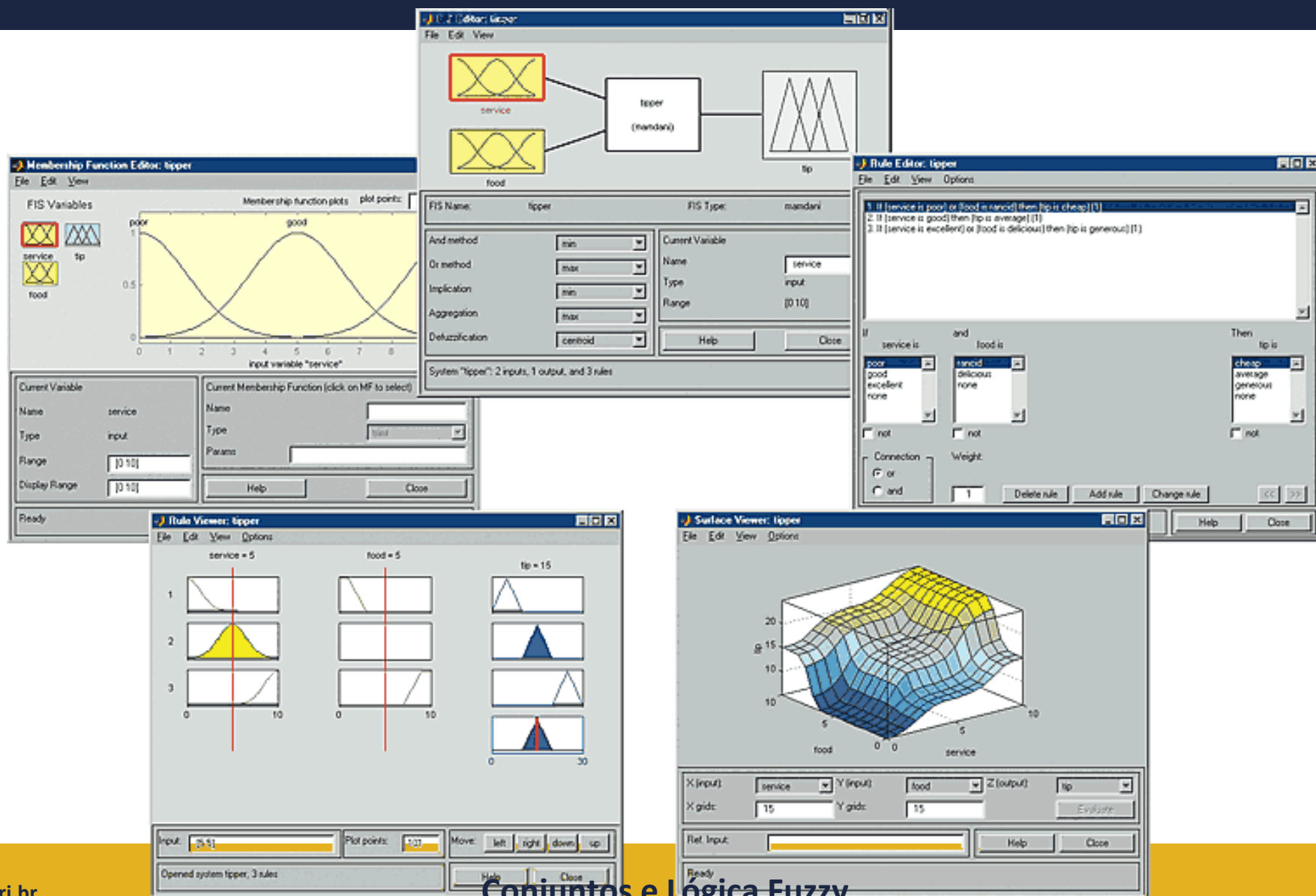
- Estabelecimento das regras
  - Teoricamente, deveria usar uma das regras de inferência na lógica
  - Na prática, a inferência de Mamdani (min) é a mais usada
  - mas “min” vale quando os casos do antecedente são verdade!

# Defuzzificação

- Transformação do resultado final em um valor único, nítido
- Várias formas, com diferentes métodos e aplicações
  - Valor que gera o grau de pertinência máximo
  - Centróide do máximo
  - Centróide das funções

# **Exemplo: Controle Fuzzy+Matlab Toolbox**

# Sistema de Regras



# Passos para Controle Fuzzy

- Identificar Entradas
  - identificar faixa de valores (variáveis)
  - identificar termos
  - definir curva dos termos
- Identificar Saídas
  - identificar faixa de valores
  - identificar termos
  - definir curvas dos termos
- Construir Base de Regras
  - Projetar (Matriz) e Codificar
  - Transformá-las em Fuzzy



# Matlab fuzzyLogicDesigner

The screenshot displays the MATLAB Fuzzy Logic Designer interface for a Mamdani fuzzy inference system. The main workspace shows two input membership functions for 'Velocidade(Km/h)' and 'Distancia(m)', both represented by overlapping bell-shaped curves. These inputs feed into a central block labeled 'Modelo Bicicleta Matlab io3 (mamdani)'. The output is a membership function for 'PercPressao', shown as three overlapping triangular curves. Below the workspace, the FIS Name is 'Modelo Bicicleta Matlab io3' and the FIS Type is 'mamdani'. The inference settings are: And method: min, Or method: max, Implication: min, Aggregation: max, and Defuzzification: centroid. The 'Current Variable' section shows 'Velocidade(Km/h)' as an input variable with a range of [0 80].

Fuzzy Logic Designer: Modelo Bicicleta Matlab io3

File Edit View

Velocidade(Km/h)

Distancia(m)

Modelo Bicicleta Matlab io3  
(mamdani)

PercPressao

FIS Name: Modelo Bicicleta Matlab io3 FIS Type: mamdani

And method: min

Or method: max

Implication: min

Aggregation: max

Defuzzification: centroid

Current Variable

Name: Velocidade(Km/h)

Type: input

Range: [0 80]

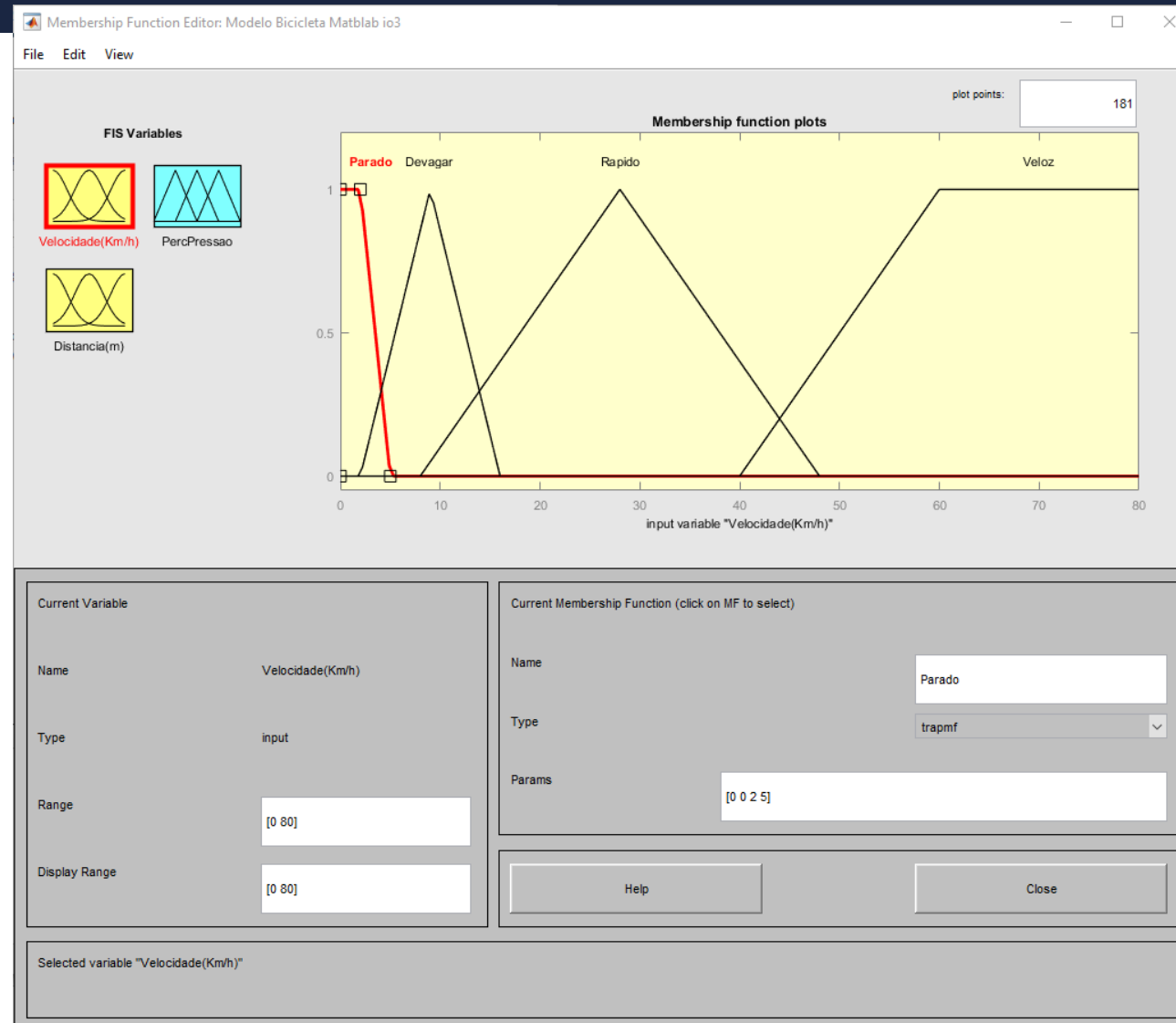
Help Close

Updating Membership Function Editor

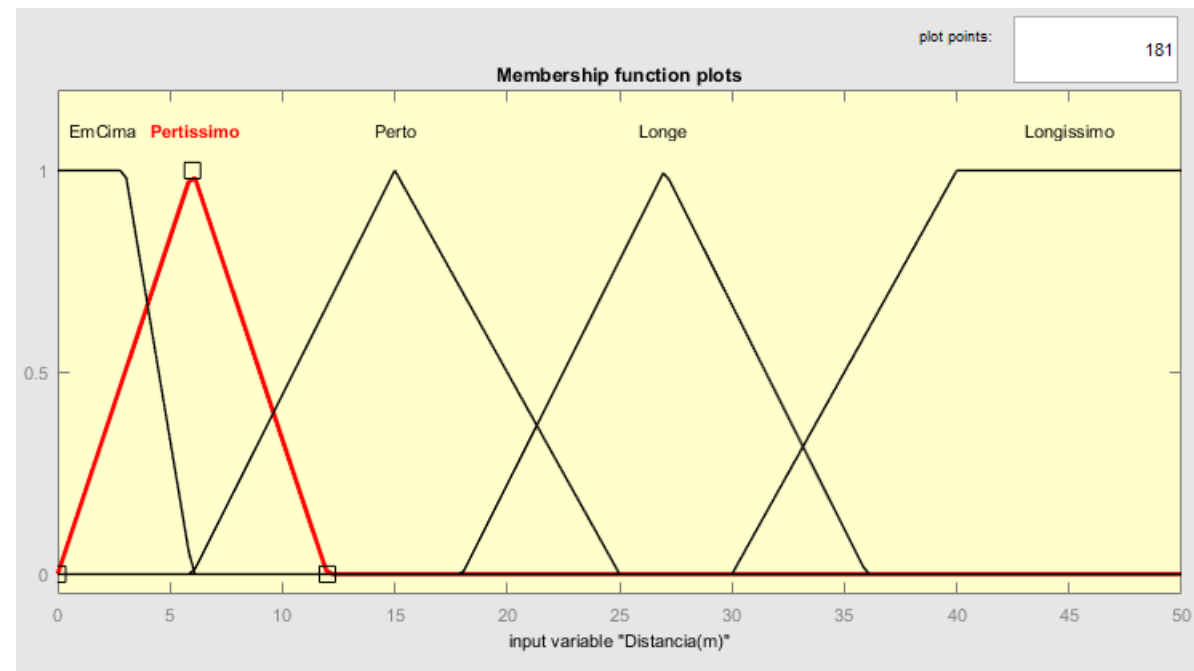
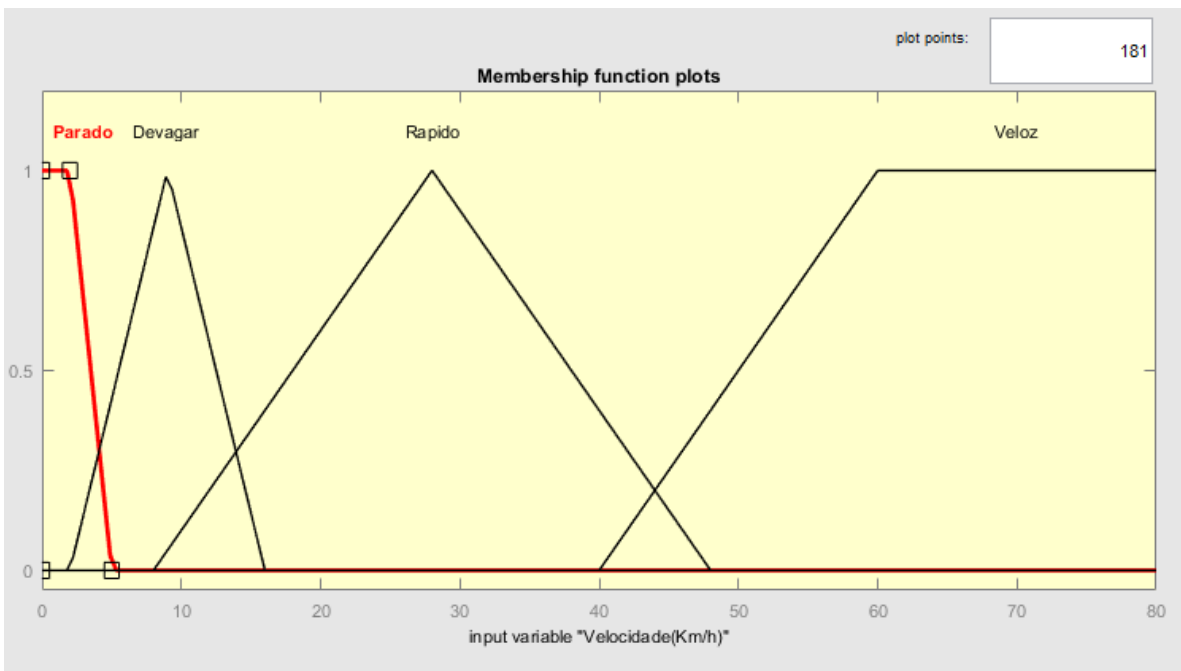
# Controle de Frenagem

- Controle de Freio de Bicicleta
  - Entrada 1: velocidade
  - Entrada 2: distância de obstáculo
  - Saída 1: Pressão no Freio

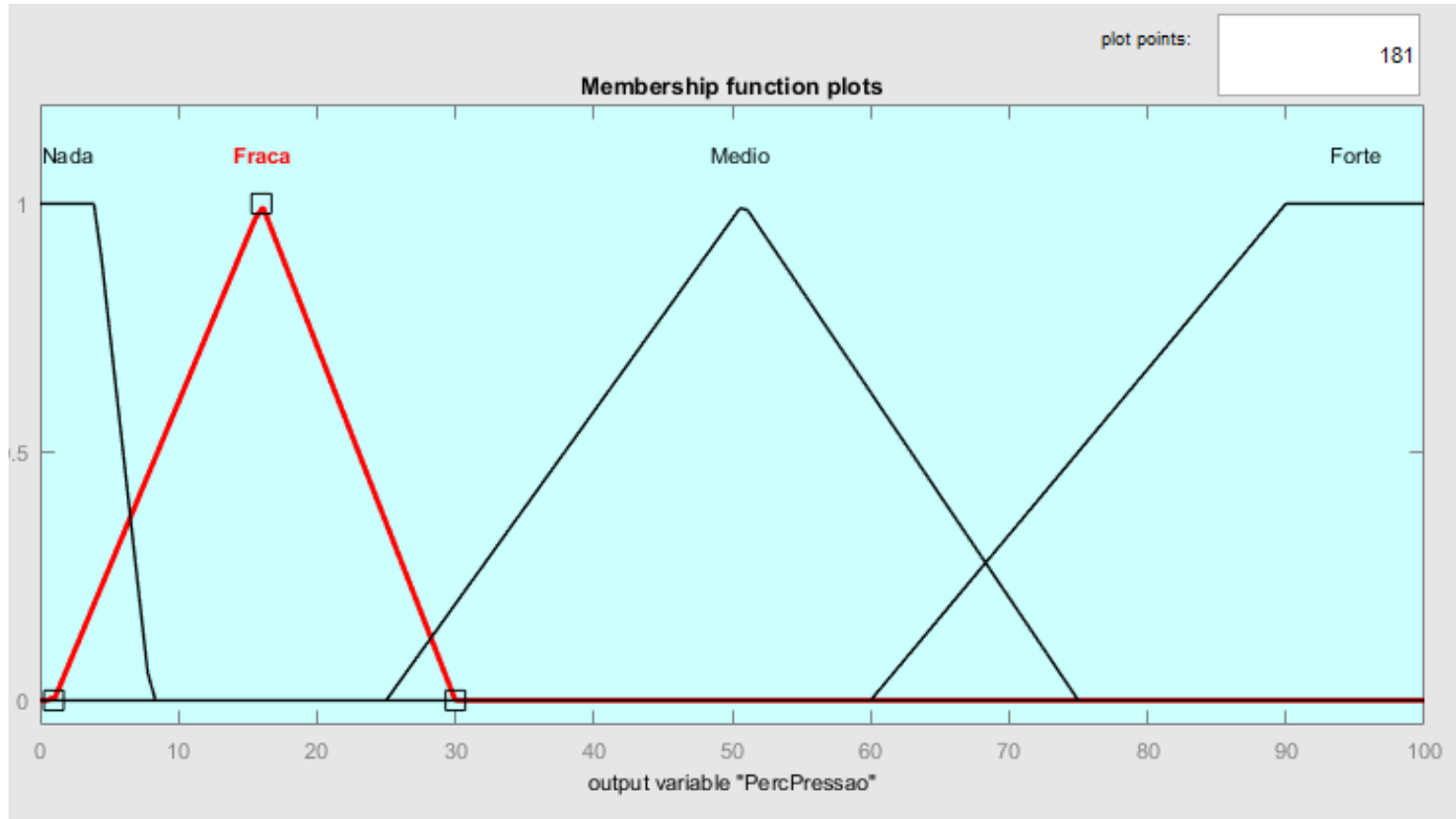
# Entradas e Saídas



# Variáveis e Termos - Entrada



# Variáveis e Termos - Saída



# Regras

- Se distância pequena e velocidade alto então freio forte
- Se distância grande e velocidade grande então freio fraco
- etc...
- Todas as regras podem ser vistas de uma vez só em um gráfico ou em uma tabela

# Tabela das Regras

	Parado	Devagar	Rápido	Veloz
Em cima	Nada	Médio	Forte	Forte
Pertíssimo	Nada	Fraca	Forte	Forte
Perto	Nada	Fraca	Forte	Forte
Longe	Nada	Nada	Médio	Médio
Longíssimo	Nada	Nada	Fraca	Fraca

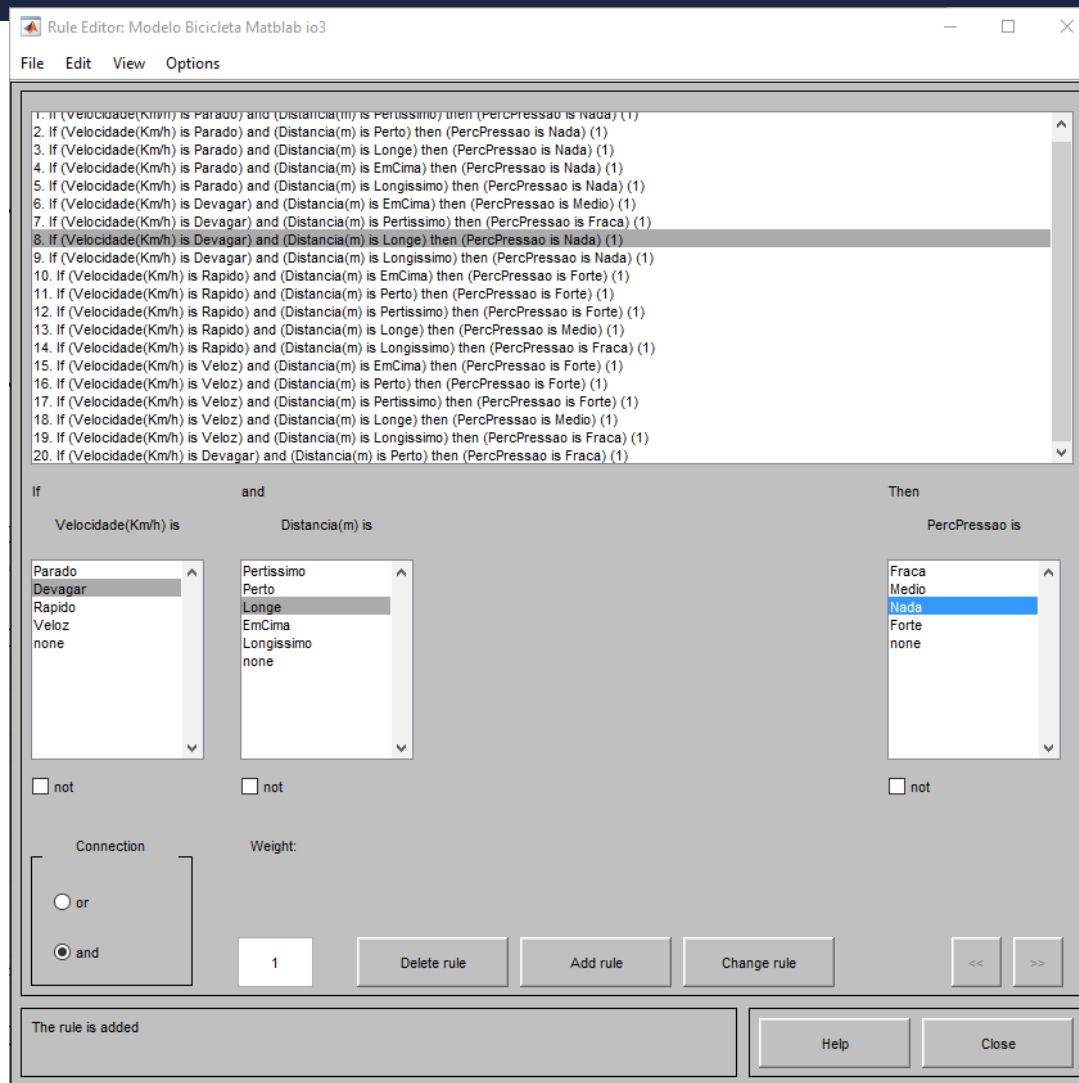
Tabela 5.1: Regras do sistema exemplo em formato tabular

# Regras no Matlab

1. If (Velocidade(Km/h) is Parado) and (Distancia(m) is Pertissimo) then (PercPressao is Nada) (1)
2. If (Velocidade(Km/h) is Parado) and (Distancia(m) is Perto) then (PercPressao is Nada) (1)
3. If (Velocidade(Km/h) is Parado) and (Distancia(m) is Longe) then (PercPressao is Nada) (1)
4. If (Velocidade(Km/h) is Parado) and (Distancia(m) is EmCima) then (PercPressao is Nada) (1)
5. If (Velocidade(Km/h) is Parado) and (Distancia(m) is Longissimo) then (PercPressao is Nada) (1)
6. If (Velocidade(Km/h) is Devagar) and (Distancia(m) is EmCima) then (PercPressao is Medio) (1)
7. If (Velocidade(Km/h) is Devagar) and (Distancia(m) is Pertissimo) then (PercPressao is Fraca) (1)
8. If (Velocidade(Km/h) is Devagar) and (Distancia(m) is Longe) then (PercPressao is Nada) (1)
9. If (Velocidade(Km/h) is Devagar) and (Distancia(m) is Longissimo) then (PercPressao is Nada) (1)
10. If (Velocidade(Km/h) is Rapido) and (Distancia(m) is EmCima) then (PercPressao is Forte) (1)
11. If (Velocidade(Km/h) is Rapido) and (Distancia(m) is Perto) then (PercPressao is Forte) (1)
12. If (Velocidade(Km/h) is Rapido) and (Distancia(m) is Pertissimo) then (PercPressao is Forte) (1)
13. If (Velocidade(Km/h) is Rapido) and (Distancia(m) is Longe) then (PercPressao is Medio) (1)
14. If (Velocidade(Km/h) is Rapido) and (Distancia(m) is Longissimo) then (PercPressao is Fraca) (1)
15. If (Velocidade(Km/h) is Veloz) and (Distancia(m) is EmCima) then (PercPressao is Forte) (1)
16. If (Velocidade(Km/h) is Veloz) and (Distancia(m) is Perto) then (PercPressao is Forte) (1)
17. If (Velocidade(Km/h) is Veloz) and (Distancia(m) is Pertissimo) then (PercPressao is Forte) (1)
18. If (Velocidade(Km/h) is Veloz) and (Distancia(m) is Longe) then (PercPressao is Medio) (1)
19. If (Velocidade(Km/h) is Veloz) and (Distancia(m) is Longissimo) then (PercPressao is Fraca) (1)
20. If (Velocidade(Km/h) is Devagar) and (Distancia(m) is Perto) then (PercPressao is Fraca) (1)



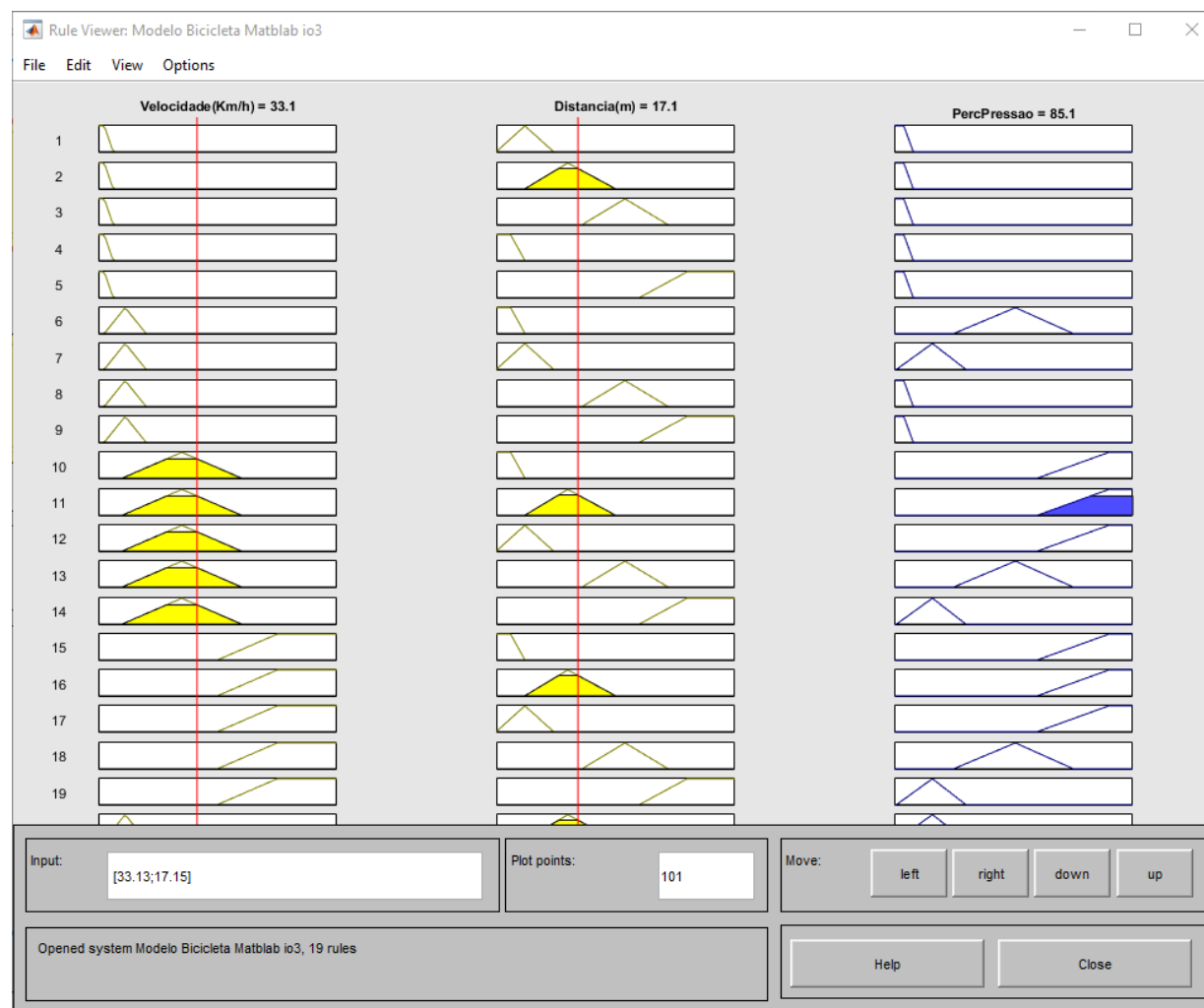
# Regras no Matlab - Interface



# Solução

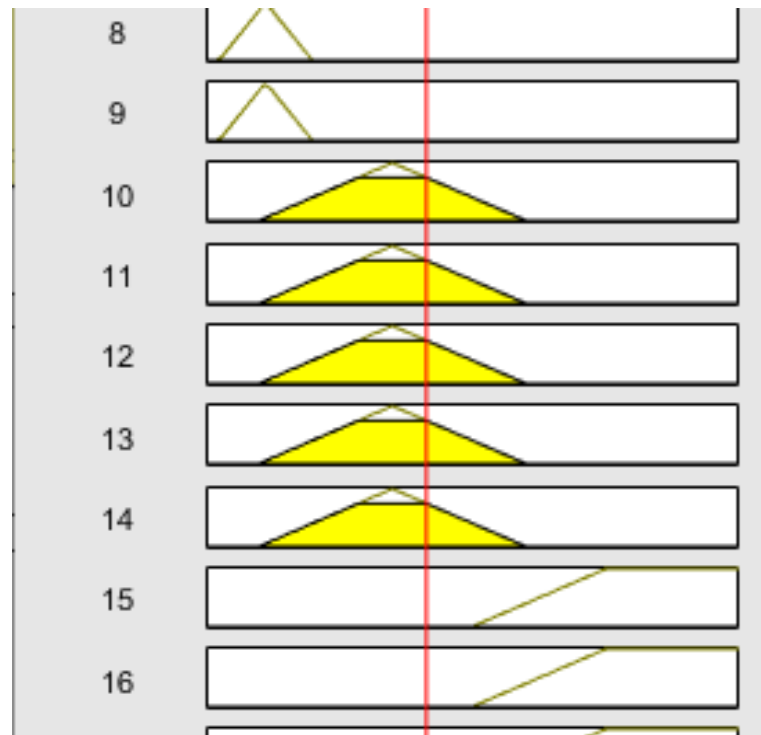
- Identifica entradas
  - 7 mph: Slow e Pretty Fast, 25 pés: Near e Real Close
- 4 saídas
- Para cada regra, calcula a conjunção
  - $\text{Slow}(.7) \text{ e } \text{Near}(.5) = .5$
- Calcula a saída final
  - OU
  - CENTRÓIDE
    - Corta cada função de saída no limite
    - Calcula o centróide
    - Calcula a média ponderada dos centróides

# Vendo a Solução - Dinâmica

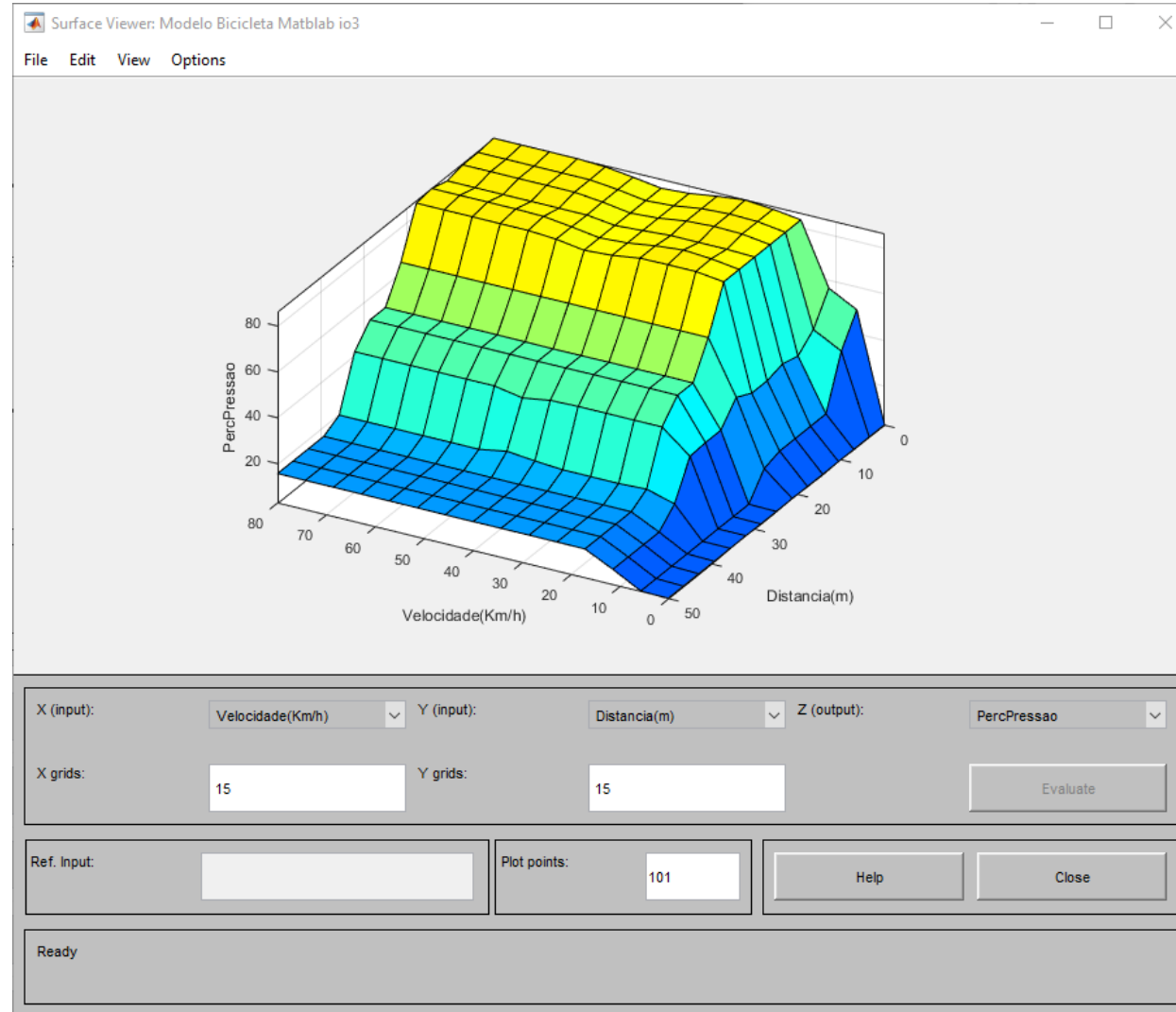


# Atenção!

- A interseção entre as curvas das funções de pertinência implica em que várias regras estarão ativas ao mesmo tempo
  - Isso é o fuzzy!



# Vendo a Solução - Completa

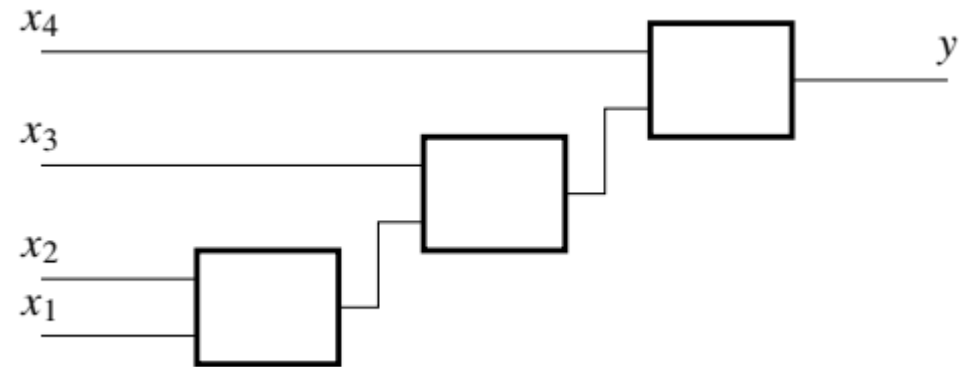
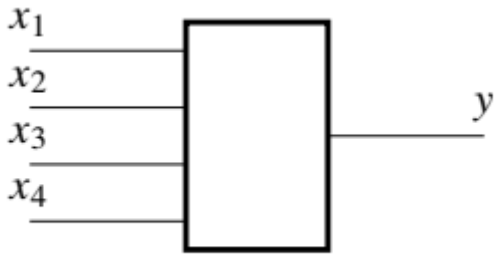


# Solução

- A solução é na verdade uma função não linear?
  - Teoricamente, pode ser qualquer função.
- O processo é simplificado, mas o resultado é o mesmo de sempre?
  - É uma função. O processo simplifica a busca pela função
  - Não é aleatório!

# Mais de 2 Entradas?

- Dependendo da ferramenta...



# **Desenvolvimento de Sistemas Fuzzy**



# Desenvolvimento de Sistemas Fuzzy

- Determinar se a escolha de um sistema fuzzy é mais correta
  - regras heurísticas ou formas aproximadas
  - modelo matemático complexo
- Identificar as entradas e saídas do sistemas, com os seus intervalos de valores
- Definir as funções de pertinência para cada parâmetro de entrada ou de saída

# Desenvolvimento de Sistemas Fuzzy

- Construir uma base de regras
  - quantas são necessárias
  - quando parar de adicionar
- Verificação do conjunto de saídas na base de regras; validação da correspondência entre a saída e a entrada de acordo com a base de regras;

# Desenvolvimento baseado em Especialistas Humanos

- Aplicado basicamente a sistemas de controle
- Baseia-se nas resposta de um indivíduo a respeito de um determinado sistema que o tenha controlado por um longo período de tempo.
- Se a pressão está muito baixa,
- eu aumento o fluxo de combustível
- Tipos: Automação de funções de controle,
  - Construção de sistema consultor,
  - Desenvolvimento de sistema de monitoração

# Desenvolvimento baseado em Tentativa-e-erro

- Aplicado a sistemas de controle, com enfoque no tempo
- Duas fases:
  - geração de sistema de controle inicial (aproximação)
  - refinamento do sistema inicial (produto final)
- Requisitos básicos: modelo matemático do sistema ou um modelo de dados
- Utiliza ferramentas gráficas para apoio na determinação dos pontos críticos de entrada

# Problemas e Limitações

- Estabilidade
- Capacidade de aprendizado
- Dificuldade em determinar ou sintonizar funções de pertinência com regras Fuzzy
- Má concepção do termo Fuzzy como impreciso ou imperfeito
- Testes extensivos para verificação e validação

# Gorgeta em FCL (1/5) – Entrada e Saída

Programa 5.2: Início do programa que calcula o percentual de gorjeta em FCL, com as criações das variáveis (Cingolani e Alcalá-Fdez, 2013)

```
1 FUNCTION_BLOCK tipper
2
3 VAR_INPUT
4     service : REAL;
5     food : REAL;
6 END_VAR
7
8 VAR_OUTPUT
9     tip : REAL;
10 END_VAR
```

# Gorgeta em FCL (2/5) - Serviço

Programa 5.3: Especificação dos termos linguísticos de entrada para serviço em FCL (Cingolani e Alcalá-Fdez, 2013)

```
1
2 FUZZIFY service
3     TERM poor := (0, 1) (4, 0) ;
4     TERM good := (1, 0) (4,1) (6,1) (9,0);
5     TERM excellent := (6, 0) (9, 1);
6 END_FUZZIFY
```

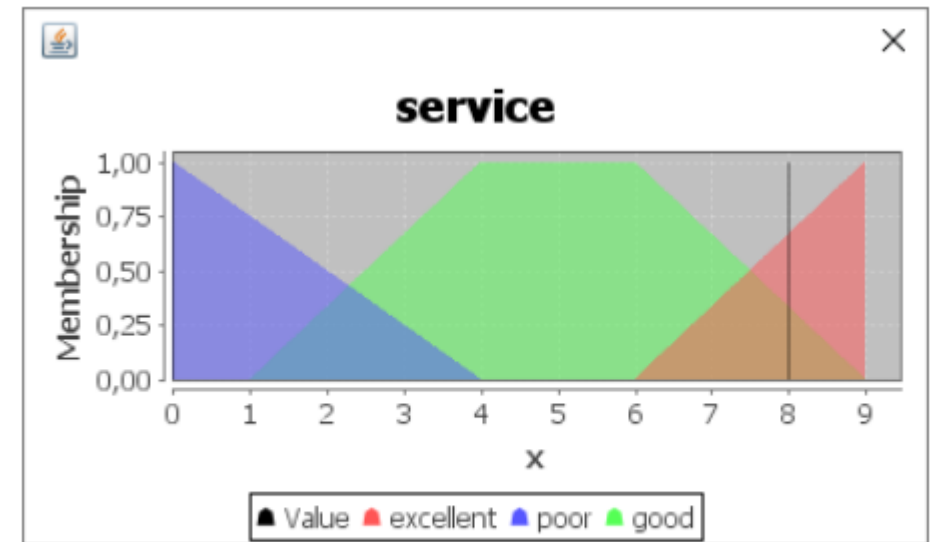


Figura 5.16: Funções de pertinência para serviço

# Gorgeta em FCL (3/5) - Comida

Programa 5.4: Especificação dos termos linguísticos de entrada para comida em FCL (Cingolani e Alcalá-Fdez, 2013)

```
1 FUZZIFY food
2     TERM rancid := (0, 1) (1, 1) (3,0) ;
3     TERM delicious := (7,0) (9,1);
4 END_FUZZIFY
```

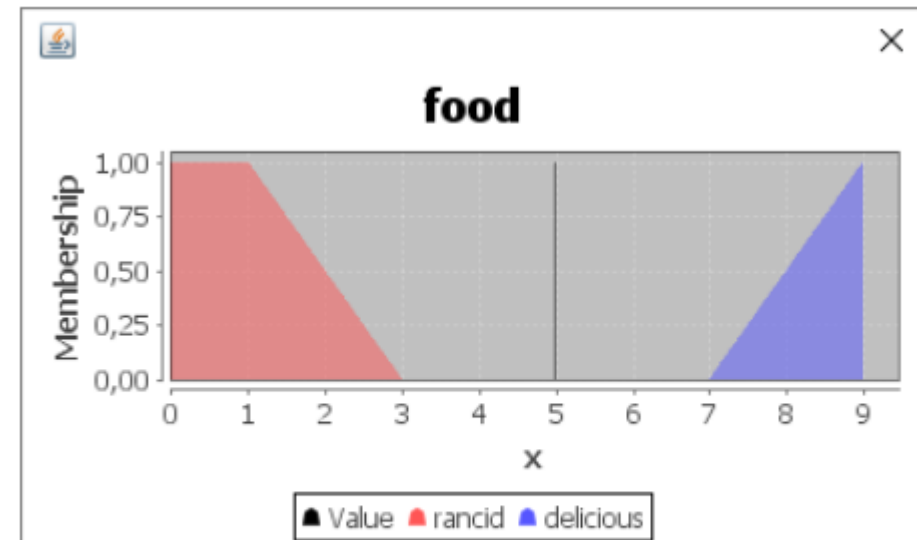


Figura 5.17: Funções de pertinência para comida



# Gorjeta em FCL (4/5) – Cálculo da Saída

Programa 5.5: Especificação do termo linguístico de saída e da regra de defuzzificação, além de um resultado default (Cingolani e Alcalá-Fdez, 2013)

```
1
2 DEFUZZIFY tip
3   TERM cheap := (0,0) (5,1) (10,0);
4   TERM average := (10,0) (15,1) (20,0);
5   TERM generous := (20,0) (25,1) (30,0);
6   // Use 'Center Of Gravity' defuzzification method
7   METHOD : COG;
8   // Default value is 0 (if no rule activates defuzzifier)
9   DEFAULT := 0;
10  END_DEFUZZIFY
```

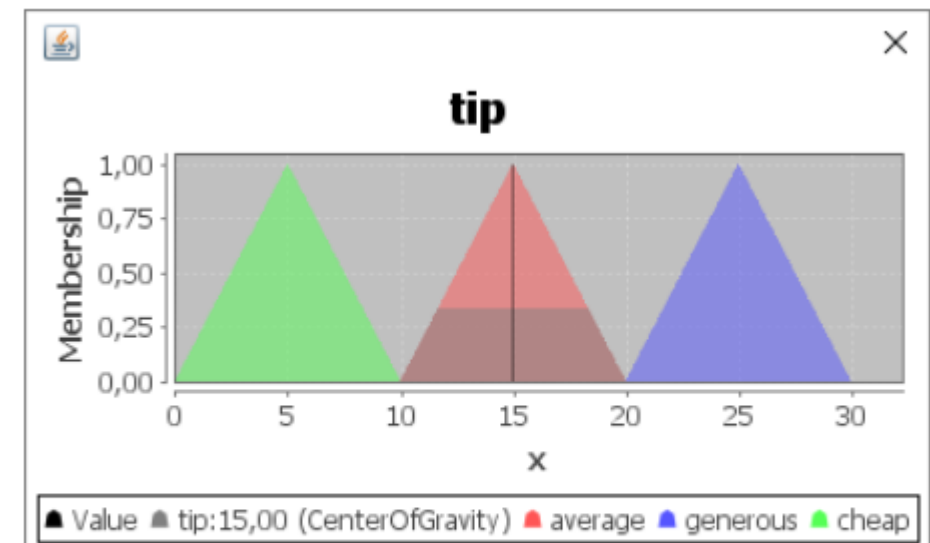


Figura 5.18: Funções de pertinência para gorjeta

# Gorgeta em FCL (5/5) – Regras

Programa 5.6: Especificação das regras e das funções usadas na sua resolução numérica, e fim do programa (Cingolani e Alcalá-Fdez, 2013)

```
1
2 RULEBLOCK No1
3     AND : MIN;
4     ACT : MIN;
5     ACCU : MAX;
6
7     RULE 1 : IF service IS poor OR food IS rancid THEN tip IS cheap;
8     RULE 2 : IF service IS good THEN tip IS average;
9     RULE 3 : IF service IS excellent AND food IS delicious THEN tip is
           ↪ generous;
10 END_RULEBLOCK
11
12 END_FUNCTION_BLOCK
```

# Programando um chip M68HC12

Programa 5.7: Programa simples de controle fuzzy para o M68HC12 (Freescale Semiconductor, 2006)

```
1 FUZZIFY    LDX #INPUT_MFS      ;Aponta para as funções de pertinência
2           LDY #FUZ_INS        ;Aponta para a tabela de entrada
3           LDAA CURRENT_INS    ;Pega o primeiro valor de entrada
4           LDAB #7              ;7 termos por entrada
5 GRAD_LOOP  MEM                ;Avalia um dos termos
6           DBNE B,GRAD_LOOP    ;Para todos os 7 termos (loop)
7           LDAA CURRENT_INS+   ;Pega o segundo valor de entrada
8           LDAB #7              ;7 termos por entrada
9 GRAD_LOOP1 MEM                ;Avalia um dos termos
10          DBNE B,GRAD_LOOP1   ;Para todos os sete termos
11          LDAB #7              ;7 termos na saída também
12 RULE_EVAL CLR 1,Y+           ;Limpa os registros usados
13          DBNE b,RULE_EVAL    ;Limpa todas as saídas fuzzy
14          LDX #RULE_START     ;Aponta para a primeira regra
15          LDY #FUZ_INS        ;Aponta para as entradas e saídas
16          LDAA #$FF           ;Inicializa Acumulador A (e limpa V-bit)
17          REV                  ;Processa a lista de regras
18 DEFUZ     LDY #FUZ_OUT        ;Aponta para a saída fuzzy
19          LDX #SGLTN_POS      ;Aponta para as respostas simples
20          LDAB #7              ;7 termos de saída
21          WAV                  ;Calcula as somas wtd av
22          DIV                  ;Divisão Final para wtd av
23          TFR Y,D              ;Move resultado para Acumulador AB
24          STAB COG_OUT        ;Salva na saída
```

# Como construir o sistema

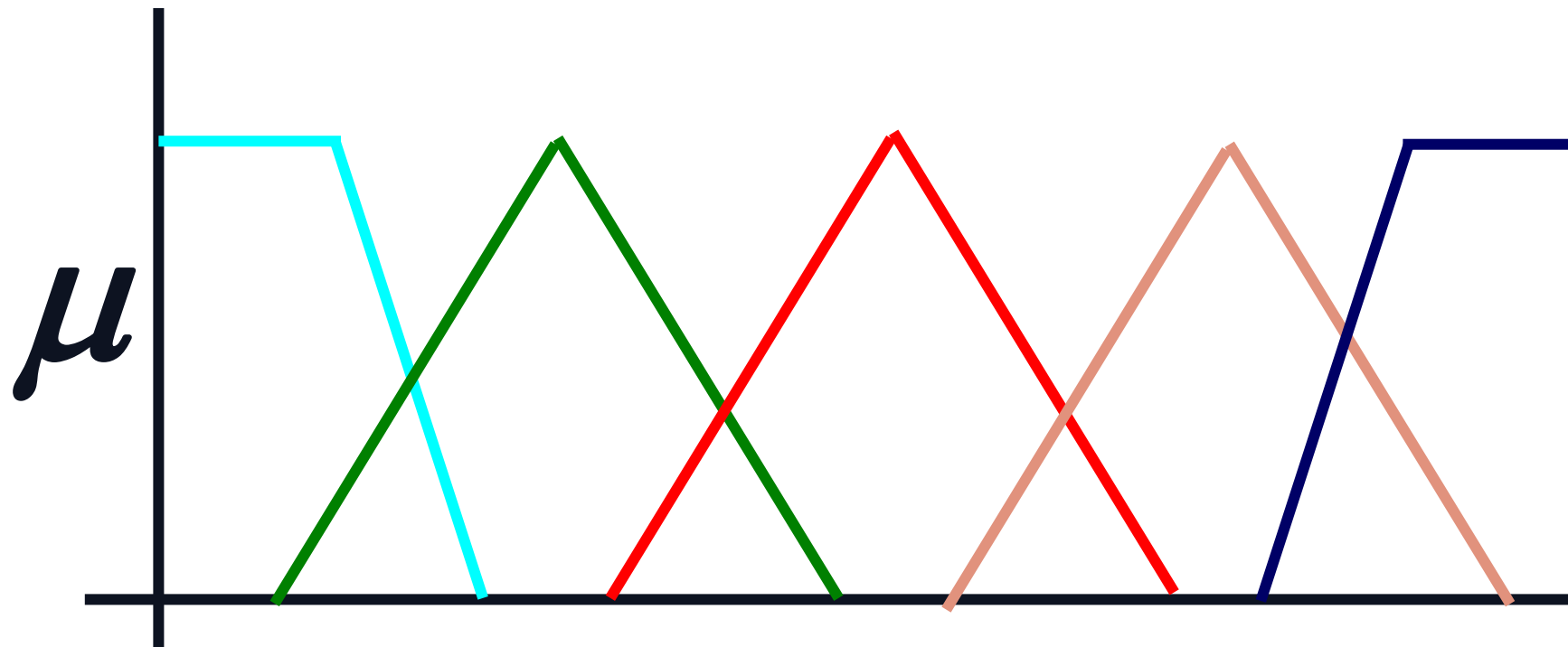
- Duas são as formas de construir sistemas Fuzzy baseados em regras
  - a partir do conhecimento especialista
  - a partir de um conjunto de dados de exemplo
- Tanto para a função de pertinência quanto para as regras

# **Determinando Funções de Pertinência**

# Det. Funções de Pertinência

- Representam uma interpretação subjetiva de um valor
- Para cada valor de entrada ou saída do sistema (Variável Linguística) determinamos um conjunto de termos que dão uma noção de valor para essa entrada (Termo Linguístico)

# Funções de Pertinência



Domínio Mensurável

# Métodos

- Intuição
- Método Horizontal
- Método Vertical
- Comparação par a par
- Inferência
- Métodos automáticos (...)



# Intuição

- O projetista escolhe as funções
- É importante o contexto
  - Temperatura
    - Água para comida
    - De uma fornalha
    - Do ar em uma sala

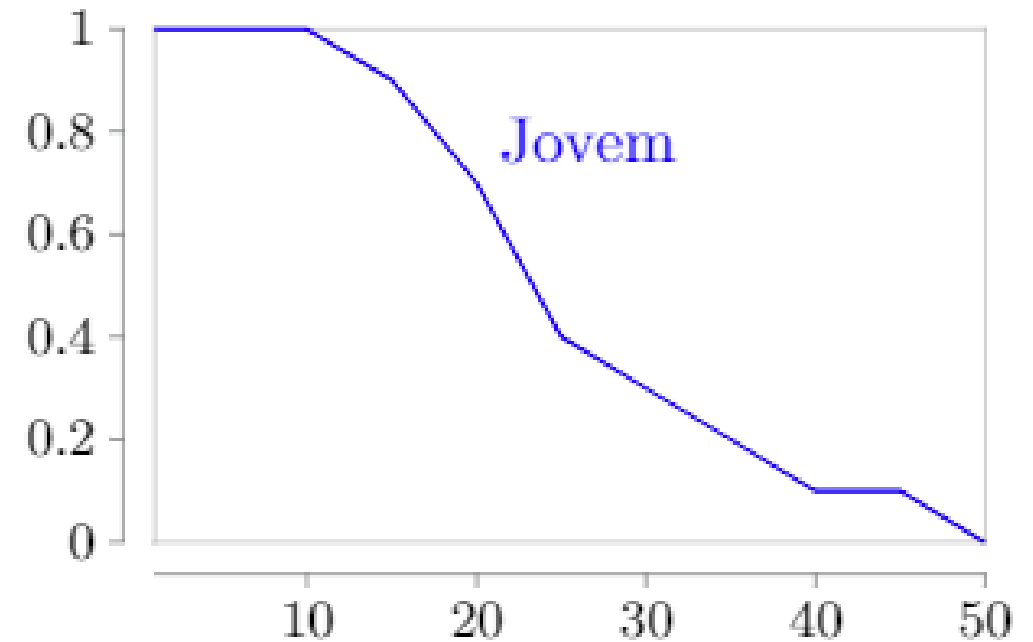


# Método Horizontal

- Baseado em entrevistas com especialistas
- Para um conjunto de valores os especialistas devem dizer se eles são ou não compatíveis com um termo
- O valor estimado é a razão entre o número de respostas sim e o número de perguntas

# Método Horizontal

Pesq.	Idades										
	1	5	10	15	20	25	30	35	40	45	50
A	S	S	S	S	N	N	N	N	N	N	N
B	S	S	S	S	N	N	N	N	N	N	N
C	S	S	S	N	N	N	N	N	N	N	N
D	S	S	S	S	S	S	S	S	N	N	N
E	S	S	S	S	S	N	N	N	N	N	N
F	S	S	S	S	S	N	N	N	N	N	N
G	S	S	S	S	S	S	S	N	N	N	N
H	S	S	S	S	S	S	N	N	N	N	N
I	S	S	S	S	S	N	N	N	N	N	N
J	S	S	S	S	S	S	S	S	S	S	N
$ S(x) $	10	10	10	9	7	4	3	2	1	1	0



# Método Vertical

- Cada conjunto é levantado por meio de cortes-alfa
  - o conjunto nítido que contém todos os elementos cujo grau de pertinência é maior que um valor alfa

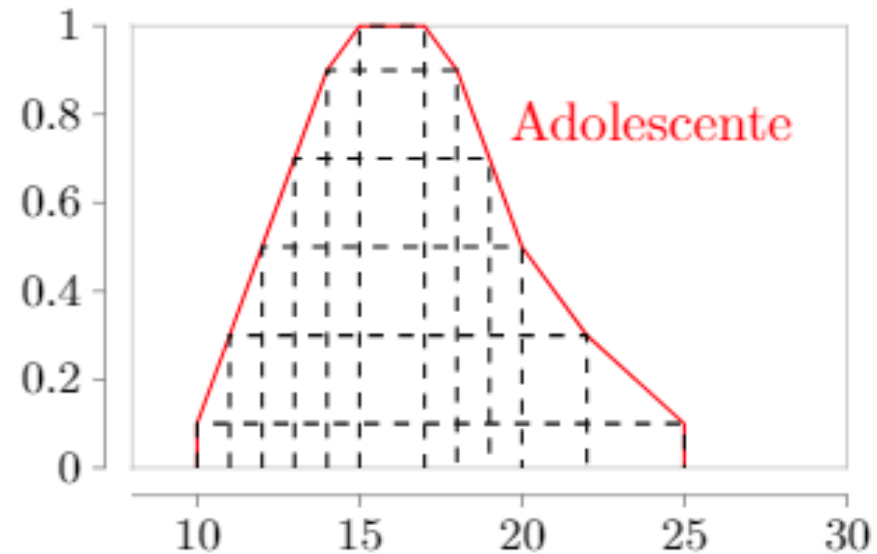
# Método Vertical

- A pergunta é
  - Que elementos são  $x\%$  compatíveis (ou mais) que o conceito dado?
- Para um especialista
- Para vários, necessita de uma operação de “média”

# Adolescentes?

- Pergunte a um pesquisador qual a idade máxima e mínima compatível  $x\%$  com o conceito de adolescente

Pesq.	mín.	máx
10%	10	25
30%	11	22
50%	12	20
70%	13	19
90%	14	18
100%	15	17



# Comparação par a par

- É construída uma matriz comparando par a par em níveis de preferência
- Essa matriz não precisa ser “simétrica”
- A partir de operações algébricas nessa matriz, desenvolve-se a função de pertinência
- Número finito de elementos

# Inferência

- Baseado em conhecimento prévio construímos funções de pertinência
- Exemplo (Triângulos)
  - I : Isósceles aproximados
  - R : Retos aproximados
  - E : Equiláteros aproximados
  - O : outros
  - IR : Isósceles e retos



# Triângulos

- $A \geq B \geq C, A+B+C=180^\circ$
- I:  $\mu_I(A,B,C)=1-\min(A-B,B-C)/60^\circ$
- R:  $\mu_R(A,B,C)=1-|A-90^\circ|/90^\circ$
- IR :  $I \cap R = \min(\mu_I(A,B,C),\mu_R(A,B,C))$
- E :  $1-|A-C|/180$
- O =  $\neg(I \cup R \cup E) = 1-\max(\dots)$

# Métodos automáticos (...)

- Redes neurais
- Algoritmos genéticos
- Semelhantes ao métodos de aprendizado de regras

# **Determinando Conjuntos de Regras**

# Determinando Regras

- Novamente, podemos usar especialistas ou dados de exemplo
- A busca de conhecimento especialista é conhecida como “Engenharia do Conhecimento” e pode ser um trabalho de muito simples até muito complicado

# Determinando Regras

- Quando temos um conjunto de dados, podemos usar esse conjunto para aprender as regras
- É uma tarefa de “aprendizado de máquina”
- É uma tarefa semelhante a dos estatísticos

# O que temos

- Um conjunto de dados
  - uma tabela
- Temos que separar entre dados de entrada e dados de saída
- “Aprendemos” em função de uma saída de cada vez
  - questões ligadas a realimentação

# Tipos de métodos

- Reduccionistas

- baseado na idéia de análise de cada caso separadamente como uma evidência
  - Wang e Mendel

- Holísticos

- baseado na idéia de todos os casos serem usados de uma só vez como evidência
  - Algoritmos Genéticos

# Método de Wang e Mendel



# Que informação está disponível?

- Supomos que
- Não há um modelo matemático
  - Model-free design problem
- Há um humano que controla o sistema com sucesso
  - Temos a experiência
  - Temos exemplos de entrada-saída gravados do controle realizado pelo humano

# Representamos experiência

- Experiência e Conhecimento normalmente são representados na forma de regras SE-ENTÃO

# Situação Normal

- Ao expressar seu conhecimento, humanos perdem alguma informação
- Os pares de entrada e saída normalmente não cobrem todas as possíveis situações

# Dados disponíveis

- Variáveis de entrada  $x_1$  e  $x_2$
- Variável de saída  $y$
- Pontos no formato  $(x_1, x_2, y)$
- Ponto
  - $P_i = (x_1^i, x_2^i, y^i)$

# O método tem 5 passos

- Dividir os espaços de entrada e saída em regiões fuzzy
- Gerar as regras fuzzy a partir dos dados
- Dar um grau para cada regra gerada, para resolver conflitos de regras
- Criar uma base de regra combinadas entre regras geradas e regras fornecidas por humanos
- Determinar o mapeamento entre entrada e saída

# Os Termos Linguísticos são Arbitrários

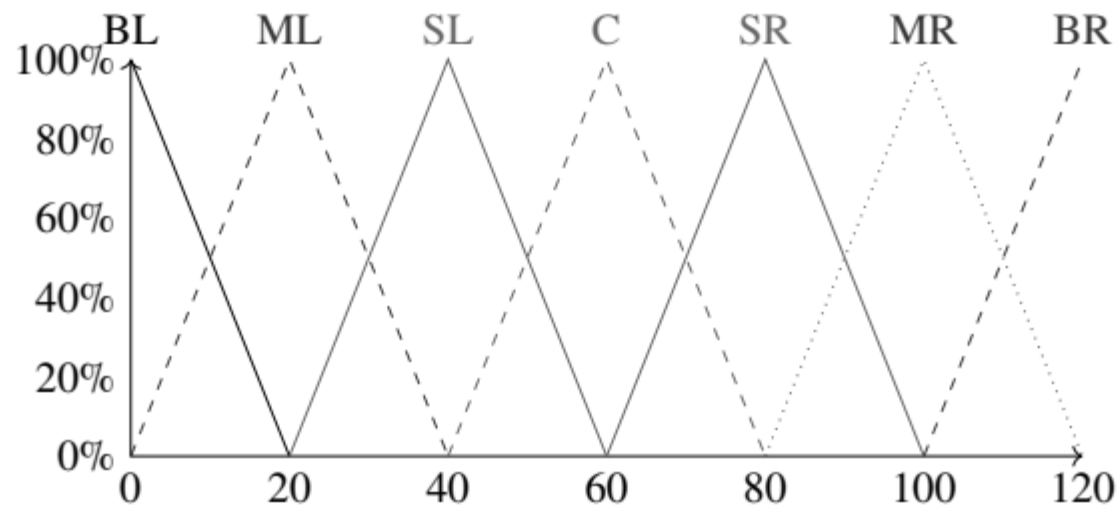


Figura 11.1: Exemplo de predefinição dos termos linguísticos para o método de Wang e Mendel (L. Wang e Mendel, 1992)

# Dividir os espaços de entrada e saída em regiões fuzzy

- Divida o intervalo de domínio em  $2N+1$  regiões
  - $N$  pode ser diferente para cada variável
  - O intervalo central é nomeado CE
  - Os intervalos menores que CE são nomeados  $SK$  ( $1 \leq K \leq N$ )
  - Os intervalos maiores que são nomeados  $BK$  ( $1 \leq K \leq N$ )
- O algoritmo original propunha formas triangulares

# Gerando regras a partir dos dados numéricos

- Dado um ponto

$$P_i = (x_1^i, x_2^i, y^i)$$

- *Determine o grau de pertinência de cada valor*

- $P_1 = (x_1^1, x_2^1, y^1)$

- $\mu_{B_1}(x_1^1) = 0.8$

- $\mu_{B_2}(x_1^1) = 0.2$



# Escolha a região de maior valor

- $P_1 = (x_1^1, x_2^1, y^1)$

- $\mu_{B_1}(x_1^1) = 0.8$

- $\mu_{B_2}(x_1^1) = 0.2$

- *0.0 nas outras*

- $\mu_{B_1}(x_1^1) = 0.8$

# Escolha a região de maior valor

- $P_1 = (x_1^1, x_2^1, y^1)$ 
  - $\mu_{S_2}(x_2^1) = 0.4$
  - $\mu_{S_1}(x_2^1) = 0.7$
  - $0.0$  nas outras
  
- $\mu_{S_1}(x_2^1) = 0.7$

# Escolha a região de maior valor

- $P_1 = (x_1^1, x_2^1, y^1)$

- $\mu_{CE}(y^1) = 0.9$

- $\mu_{B_1}(y^1) = 0.2$

- *0.0 nas outras*

- $\mu_{CE}(y^1) = 0.9$

# Cada ponto define uma regra

$$(x_1^{(1)}, x_2^{(1)}; y^{(1)}) \Rightarrow [x_1^{(1)}(0.8 \text{ in } B1, \text{ max}), x_2^{(1)}(0.7 \text{ in } S1, \text{ max}); \\ y^{(1)}(0.9 \text{ in } CE, \text{ max})] \Rightarrow \text{Rule 1:}$$

IF  $x_1$  is  $B1$  and  $x_2$  is  $S1$ , THEN  $y$  is  $CE$ ;

$$(x_1^{(2)}, x_2^{(2)}; y^{(2)}) \Rightarrow [x_1^{(2)}(0.6 \text{ in } B1, \text{ max}), x_2^{(2)}(1 \text{ in } CE, \text{ max}); \\ y^{(2)}(0.7 \text{ in } B1, \text{ max})] \Rightarrow \text{Rule 2:}$$

IF  $x_1$  is  $B1$  and  $x_2$  is  $CE$ , THEN  $y$  is  $B1$ .

# As regras montam uma matriz

- $B1 \wedge S1 \Rightarrow CE$

- $B1 \wedge CE \Rightarrow B1$

	S2	S1	CE	B1	B2
B3					
B2					
B1		CE	B1		
CE					
S1					
S2					
S3					

# É possível que haja conflitos

- Duas regras com o mesmo antecedente

- Geram um conseqüente diferente

$X_1$	$X_2$	$Y$
B1	S1	CE
B1	CE	B1
B1	S2	CE
B3	S2	CE
B1	S1	B1
B2	S2	S1
B3	CE	S1

# Precisamos dar um peso para as regras

- O peso das regras é o produto dos graus de pertinência de cada membro da regra

# E se soubermos a importância do exemplo?

- Podemos associar a cada tupla um grau de pertinência mi
- Multiplicamos esse valor pelo valor anterior



# Preenchendo a “matriz”

- Escolha a regra com maior peso para preencher a casa da matriz
  - Por isso digo que essa abordagem é reducionista, já que ela considera apenas um valor como melhor exemplo e não vê a influência de todos os valores ou o resultado esperado
  - Ela só olha o comportamento do controlador e não do controlado
    - O fator  $m$  pode ser usado para indicar o comportamento do controlado
  - Em compensação, em um passo pelos dados se gera o modelo (matriz)

# Importância do Exemplo?

- Em casos reais podemos ter:
  - Exemplos com diferentes confiabilidades
  - Exemplos gerados por pessoas com diferentes experiências
  - Exemplos com diferentes resultados (qualidade do resultado)

# Criar uma base de regra combinada (dados+humanos)

- Dar pesos as regras citadas por humanos
- Aplicar o mesmo procedimento de escolha

# Determinar o mapeamento entre entrada e saída

- Para cada para  $(x_1, x_2)$ , o grau de pertinência do consequente é o produto
- A defuzificação é feita com o centróide

# Wang e Mendel

- Cria as funções de pertinência em função dos valores disponíveis
- Para cada regra criamos uma regra que a representa
- Para cada regra calculamos um índice de qualidade
- Quando duas ou mais regras tem o mesmo antecedente mas consequentes diferentes, escolhemos a com maior índice

# Fuzzy C-Means

# Fuzzy C-Means

- Algoritmo de agrupamento (Dunn, 1973)
- Minimiza uma função objetivo

$$J_m = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty$$

- Seguindo uma regra iterativa

$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^N \mu_{ij}^m \cdot x_i}{\sum_{i=1}^N \mu_{ij}^m}$$

# Fuzzy C-means – Exemplo

Atributo 1	Atributo 2
2	12
4	9
7	13
11	5
12	7
14	4

Tabela 11.5: Pontos usados no exemplo

Grupos	$\mu((x,y))$					
1	0.034600	0.060194	0.111226	0.979533	0.966514	0.968711
2	0.965400	0.939806	0.888774	0.020467	0.033486	0.031289

Tabela 11.6: Graus de pertinência



```
1  clc
2  clear all
3  pkg load fuzzy-logic-toolkit
4
5  #dados originais de
6  entrada = [2 12; 4 9; 7 13; 11 5; 12 7; 14 4];
7  clusters = 2;
8  [centros, particao_soft, history_funcao_objetivo] = ...
9      fcm (entrada, clusters)
10
11 # Marca os pontos como x azuis
12 figure ('NumberTitle', 'off', 'Name', 'FCM Demo 1');
13 for i = 1 : rows (entrada)
14     plot (entrada(i, 1), entrada(i, 2), 'LineWidth', 2, ...
15         'marker', 'x', 'color', 'b');
16     hold on;
17 endfor
18
19 # Marca os centros como asteriscos vermelhos
20 for i = 1 : clusters
21     plot (centros(i, 1), centros(i, 2), ...
22         'LineWidth', 4, 'marker', '*', 'color', 'r');
23     hold on;
24 endfor
25
26 # Melhora a figura
27 xlim ([0 15]);
28 ylim ([0 15]);
29 xlabel ('Atributo 1');
30 ylabel ('Atributo 2');
31 grid
32 hold
```

# Resultado

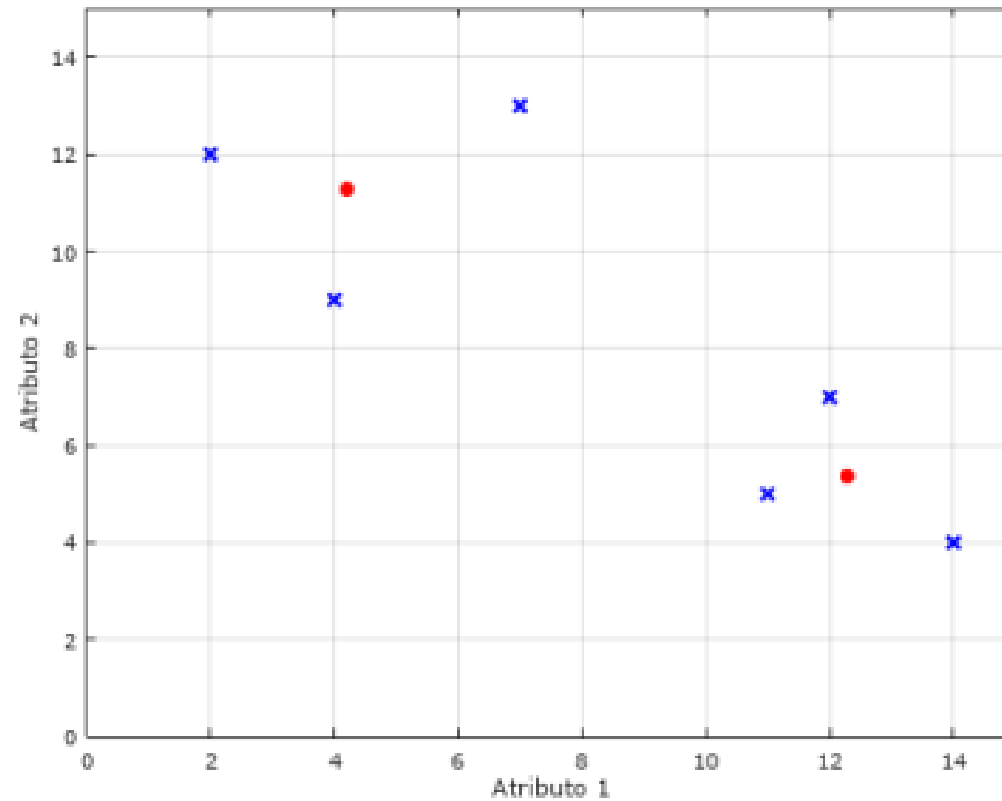
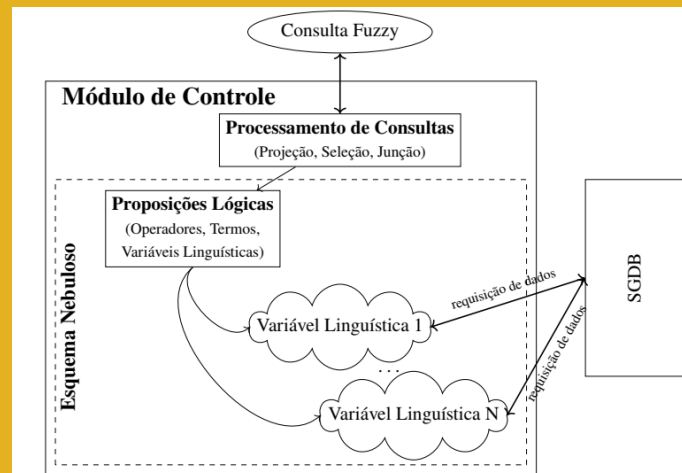
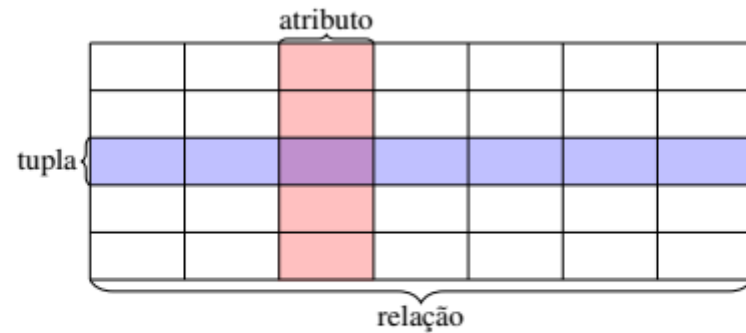


Figura 11.7: Resultado do programa 11.2

# Bancos de Dados Fuzzy



# Bancos de Dados Relacionais - Nítidos



Produtos		
id	nomeProduto	precoProduto
1	Pão	R\$1,00
2	Leite	R\$4,00
3	Queijo	R\$15,00
4	Presunto	R\$16,00
5	Ovo (dúzia)	R\$5,00

Tabela 10.1: Exemplo de uma tabela de preços de produto.

Figura 10.1: Modelo abstrato de uma relação como uma tabela



Figura 10.2: Um exemplo bastante pequeno de Banco de Dados Relacional

Programa 10.1: Comandos em SQL para criar as tabelas da Figura 10.2

```
CREATE TABLE IF NOT EXISTS 'Locais' (  
    'Local' TEXT,  
    'Terreno' TEXT,  
    'Desenvolvimento' TEXT,  
    PRIMARY KEY('Local')  
);  
  
CREATE TABLE IF NOT EXISTS 'Especialistas' (  
    'Nome' TEXT,  
    'Empresa' TEXT,  
    'Especialidade' TEXT,  
    PRIMARY KEY('Nome')  
);  
  
CREATE TABLE IF NOT EXISTS 'Avaliacoes' (  
    'Nome' TEXT,  
    'Local' TEXT,  
    'Custo' INTEGER NOT NULL,  
    PRIMARY KEY('Nome', 'Local'),  
    FOREIGN KEY('Nome') REFERENCES 'Especialistas'('Nome'),  
    FOREIGN KEY('Local') REFERENCES 'Locais'('Local')  
);
```

Programa 10.2: Sentença em SQL que recupera todas as avaliações de João sobre terrenos pantanosos

```
SELECT * FROM Locais, Avaliações  
WHERE Avaliações.Nome="João"  
AND Locais.Terreno="Pantanosos"  
AND Locais.Local=Avaliações.Local;
```

# Bancos de Dados Fuzzy

- Podemos ter dois tipos
  - dados nítidos, consultas Fuzzy
  - dados Fuzzy
- Vejamos um exemplo de uso de bancos de dados com consultas Fuzzy e capacidade de utilizar regras Fuzzy

# Projeto na Coppe/Sistemas

- Banco de Dados Fuzzy
- Avaliação de Qualidade de Software Fuzzy
  - Com produto em uso no mercado
- Confiabilidade de END de Ultra Som
- Alocação de Recursos Humanos

# Bando de Dados Fuzzy

- Tese de Mestrado de André Braga
- Ferramentas de Manipulação Nebulosa de Dados com Aplicação em Sistemas de Informação Geográfica
- Maio/1998



# CLOUDS (Braga, 1998)

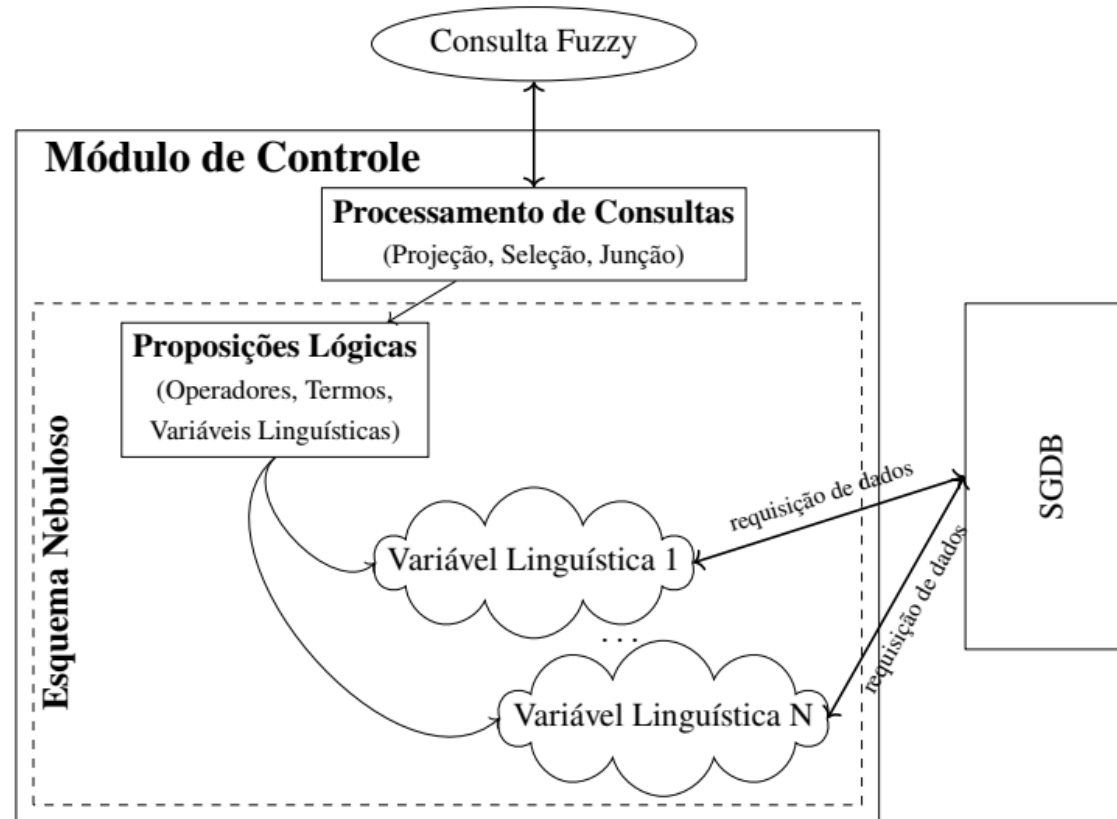


Figura 10.3: O mapeamento fuzzy em CLOUDS, adaptado de (Braga, 1998)

# Arquitetura de CLOUDS

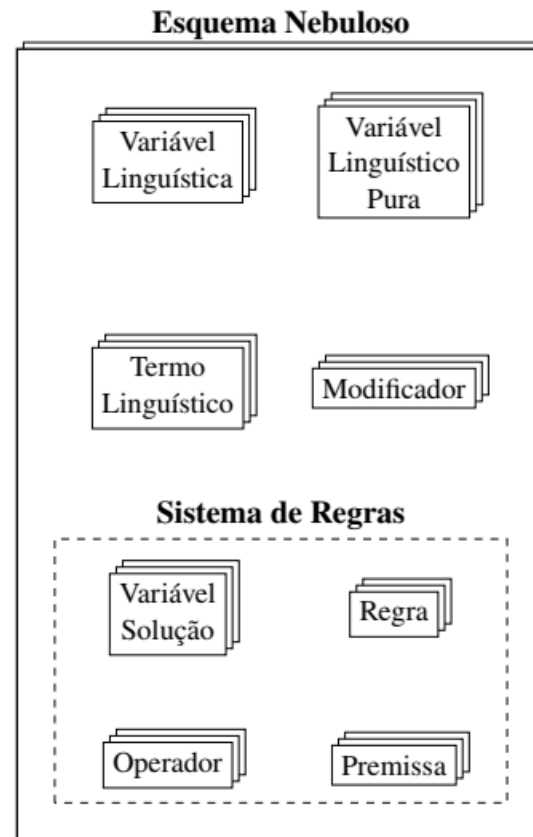
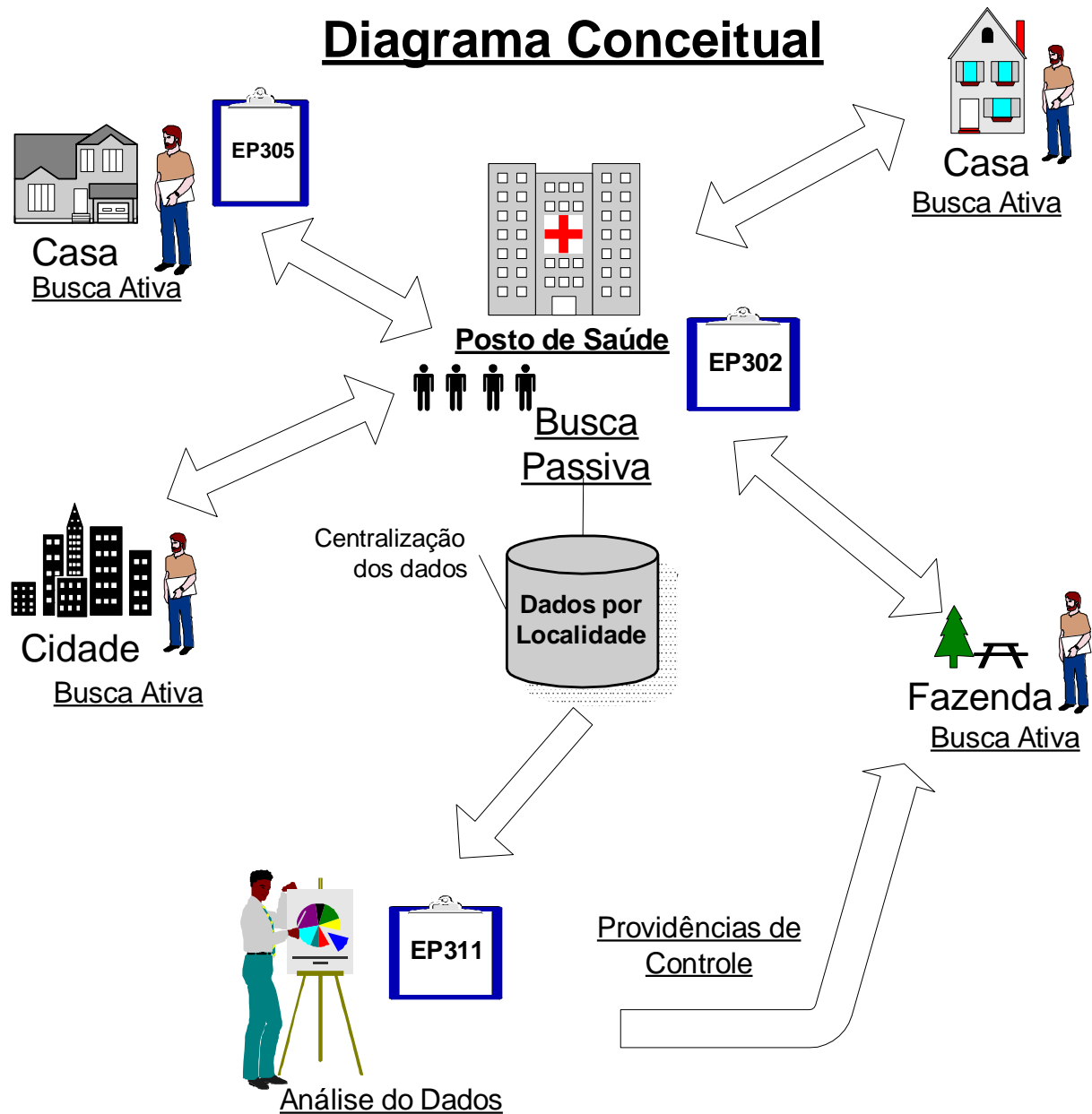


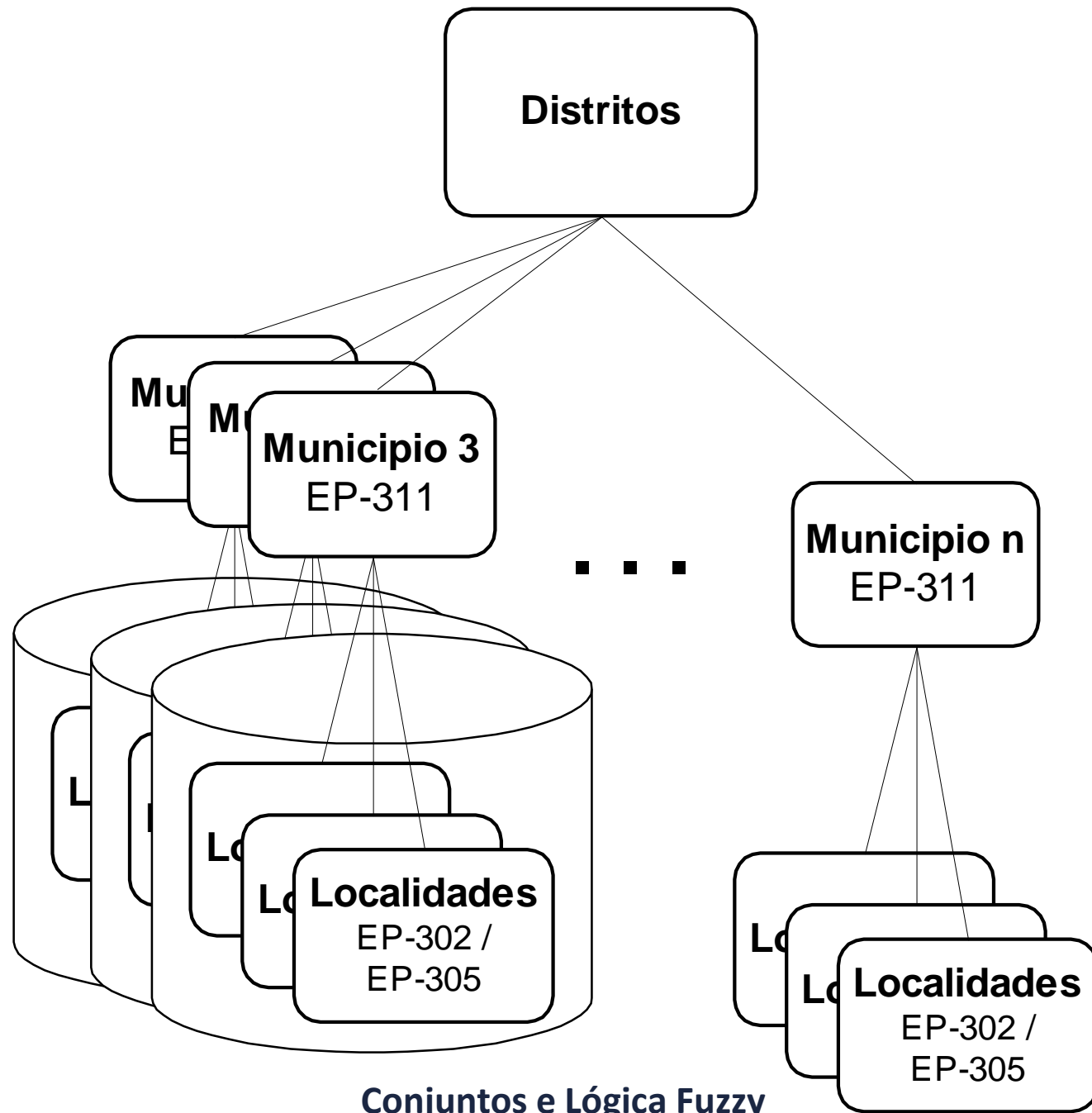
Figura 10.4: A arquitetura de CLOUDS, adaptada de (Braga, 1998)

# Uma aplicação

- Banco de dados geo-referenciado contendo dados sobre malária
- Software “nítido” com extensão nebulosa: SIGEPI (Coppe/Biomédica)
- Tese de mestrado de André Braga (Coppe/Sistemas)

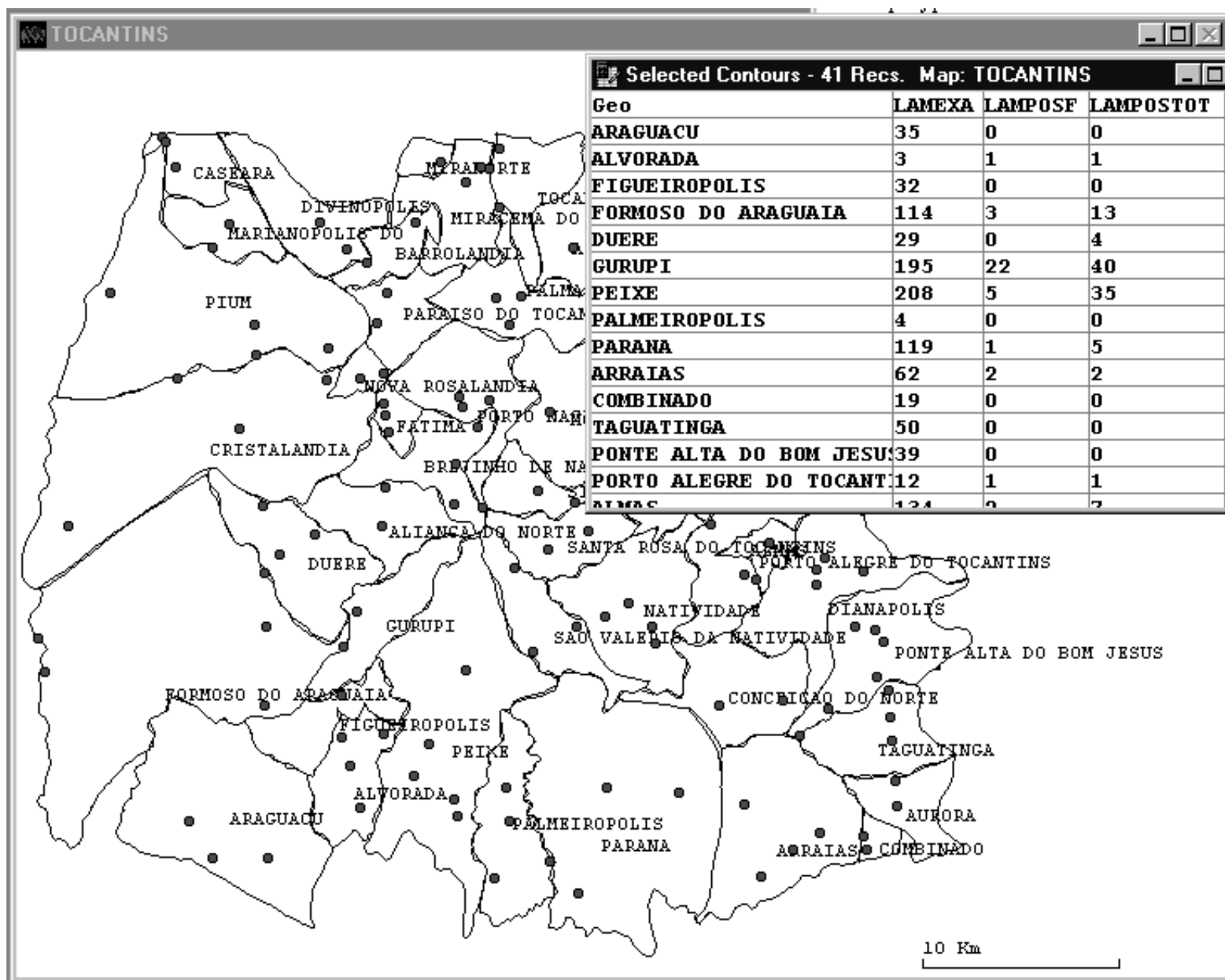
# Diagrama Conceitual





# Colunas da Tabela

- LAMEXA - Número de indivíduos examinados. A sigla refere-se a número de lâminas examinadas.
- LAMPOSF - Número de indivíduos que acusaram somente o Plasmodium falciparum no sangue.
- LAMPOSV - Número de indivíduos que acusaram somente o Plasmodium vivax no sangue.
- LAMPOSM - Número de indivíduos que acusaram somente o Plasmodium malariae no sangue.
- LAMPOSFV - Número de indivíduos que acusaram o Plasmodium falciparum e Plasmodium vivax no sangue.
- LAMPOSTOT - Número total de indivíduos que acusaram o Plasmodium no sangue.
- CASEXI - Número de casas existentes na área de pesquisa.
- CASVIS - Número de casas visitadas na área de controle



# Variáveis Criadas

- VISITAS - Variável criada pelo sistema para controle do aumento ou diminuição de visitas nas diversas áreas.
- RETORNO - Variável criada para indicar a possibilidade de retorno da doença devido a ação do Plasmodium vivax.



# Comandos

- Create Table Tocantins FROM tab1 SIG;
- Create Table Mapa AS mapa.dat SIG;
- Create Table Visitas AS Visitas.dat ARQTEXT;
- Create Table Epidemica AS epidem.dat ARQTEXT;
- Create Table Relatorio11 AS Relat11.dat ARQTEXT;

# Variáveis Lingüísticas

- Variáveis Lingüísticas
  - Descrevem algo que podemos avaliar (medir) no mundo real
    - idade
- Termos Lingüísticos
  - avaliam (ou medem) uma variável linguística
    - velho, novo

# Variáveis Linguísticas

- Modificadores
  - alteram a intensidade de um termo linguístico
    - muito, pouco, algo

# Criando Var. Lingüísticas

- **Create LingVar Integer *LaminasExaminadas* OVER *Tocantins* AS LAMEXA;**
- **Create LingVar Integer *LamPosFalciparum* OVER *Tocantins* AS LAMPOSF;**
- **Create LingVar Integer *LamPosVivax* OVER *Tocantins* AS LAMPOSV;**

# Criando Var. Lingüísticas

- Create LingVar Real *PerFalciparum* OVER *Tocantins*  
FROM LAMPOSF/LAMEXA;
- Create LingVar Real *PerVivax* OVER *Tocantins*  
FROM LAMPOSV/LAMEXA;
- Create LingVar Real *PerMalariae* OVER *Tocantins*  
FROM LAMPOSM/LAMEXA;

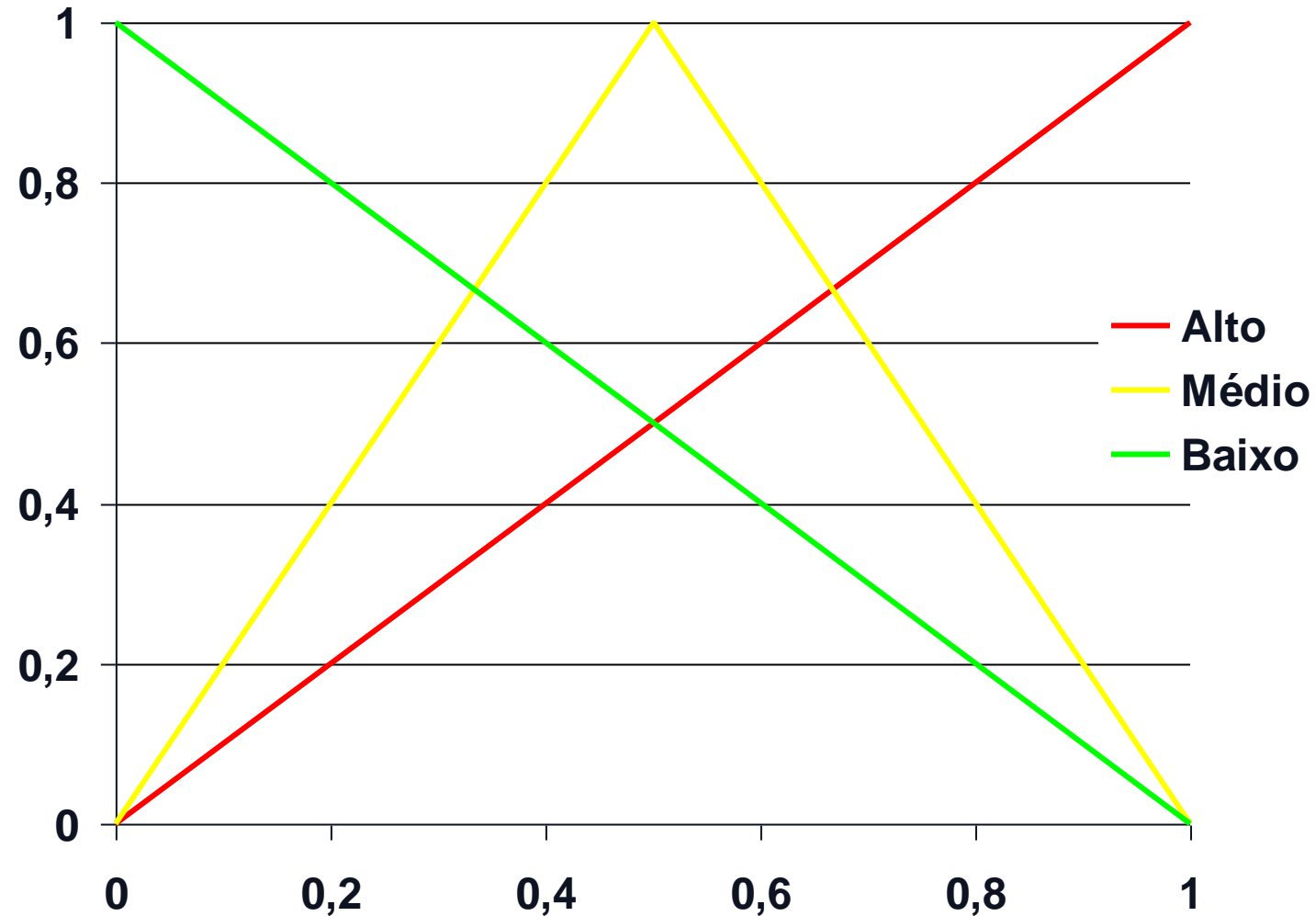
# Criando Variáveis Solução

- **Create SOLVAR Real *Visitas* DOMAIN 0, 1 OVER Tocantins AS VISITAS;**
- **Create SOLVAR Real *Retorno* DOMAIN 0, 1 OVER Tocantins AS VISITAS;**

# Os Termos Linguísticos

- **Create LingTerm** *Baixo* AS TRIANGLE (0.0, 0.0, 1.0) ;
- **Create LingTerm** *Médio* AS TRIANGLE (0.0, 0.5, 1.0) ;
- **Create LingTerm** *Alto* AS TRIANGLE (0.0, 1.0, 1.0) ;

# Alto/Médio/Baixo





# T.L. para Conclusões

- **Create LingTerm** *Permanecer* **AS TRIANGLE** (0.0, 0.0, 1.0) ;
- **Create LingTerm** *Aumentar* **AS TRIANGLE** (0.0, 1.0, 1.0) ;
- **Create LingTerm** *Improvável* **AS TRIANGLE** (0.0, 0.0, 1.0) ;
- **Create LingTerm** *Provável* **AS TRIANGLE** (0.0, 1.0, 1.0) ;

# Criando Modificadores

- **Create** hedge *Muito* **AS** concentrator 3.0;
- **USE** hedge *Muito* **IN** *Baixo*;
- **USE** hedge *Muito* **IN** *Medio*;
- **USE** hedge *Muito* **IN** *Alto*;

# Usando T.L. em V.L.

- USE Lingterm *Baixo* In *LaminasExaminadas*;
- USE Lingterm *Medio* In *LaminasExaminadas* ;
- USE Lingterm *Alto* In *LaminasExaminadas* ;
- USE Lingterm *Baixo* In *LamPosFalciparum* ;
- USE Lingterm *Medio* In *LamPosFalciparum* ;
- USE Lingterm *Alto* In *LamPosFalciparum* ;
- ...

# Criando Regras

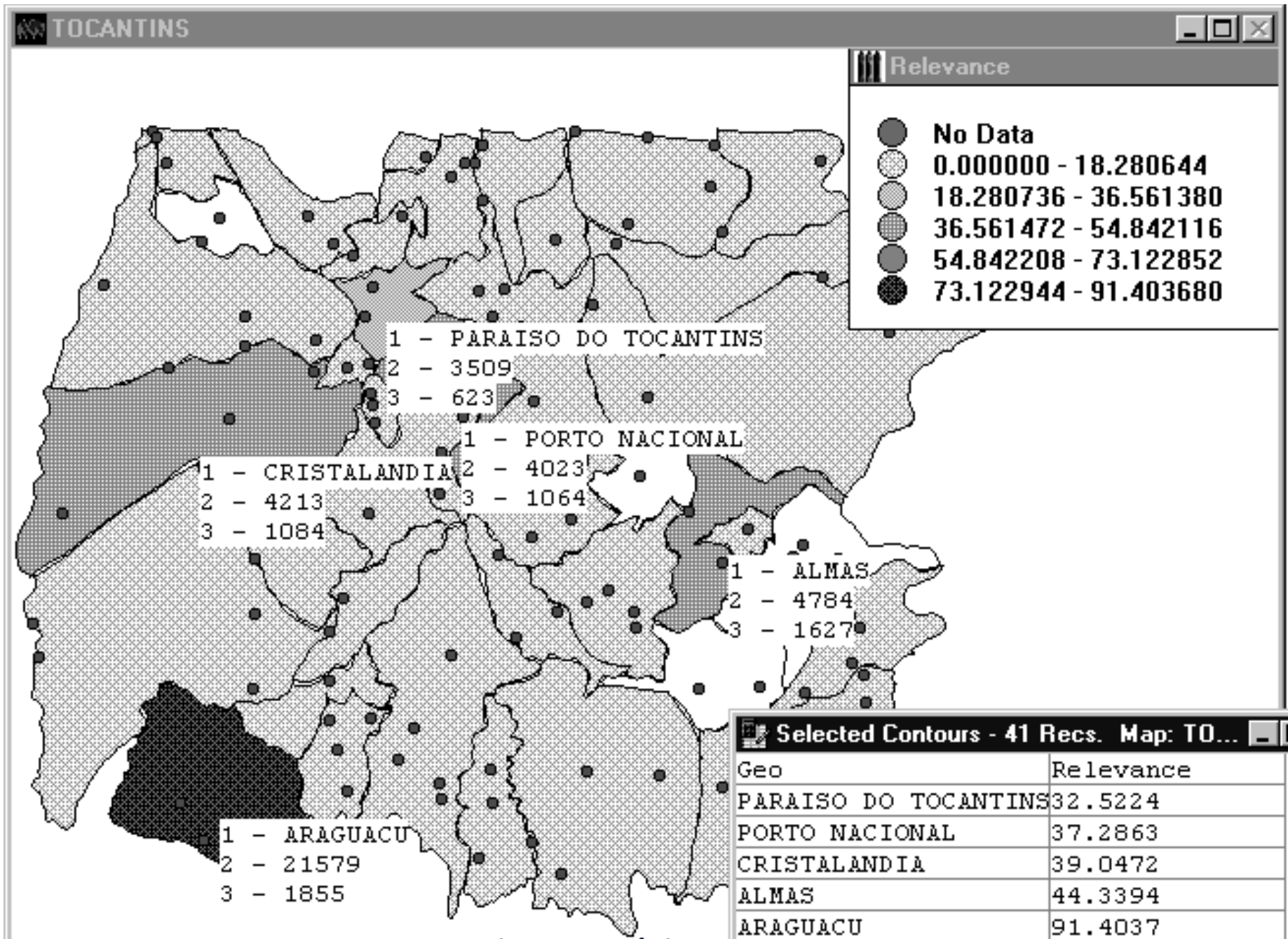
- **IF** *PerVisitadas* **IS** *Baixo* **THEN** *Visitas* **IS** *Aumentar*;
- **IF** *PerPositivasTotal* **IS** *Baixo* **THEN** *Visitas* **IS** *Permanecer*;

# Mais Regras

- **IF** *PerVivax* **IS** *Alto* **THEN** *Retorno* **IS** *Provável*;
- **IF** *PerVivax* **IS** *Baixo* **THEN** *Retorno* **IS** *Improvável*;

# Fazendo Consultas

- **SELECT** *CasasExistentes, PerVisitadas, Visitas* **FROM** *Tocantins* **TO** *Mapa* **WHERE**  $((\textit{Visitas} \text{ IS } \textit{Aumentar}) \text{ AND } ((\textit{CasasExistentes} \text{ IS } \textit{Medio}) \text{ OR } \textit{CasasExistentes} \text{ IS } \textit{Alto}))$  **WITH** *0.3*;



# Fuzzy Prolog



# Exemplo em SWI-Prolog

Programa 13.1: Um programa para calcular relações familiares

```
1 pai(manoel , david) . pai(manoel , flavia) . pai(antonio , manoel) .
2 pai(jose , antonio) . pai(leon , tania) . pai(pedro , luana) .
3 pai(carlos , leon) . pai(marcos , sara) . pai(pedro , regina) .
4 pai(leon , debora) . pai(paulo , christiana) . pai(luis , eduardo) .
5 pai(gaston , pedro) . pai(sergio , valeria) . pai(marcos , sara) .
6
7 mae(valeria , regina) . mae(regina , christiana) . mae(regina , eduardo) .
8 mae(sara , tania) . mae(luana , manoel) . mae(valeria , luana) .
9 mae(luisa , pedro) . mae(amanda , valeria) . mae(helena , sara) .
10 mae(sara , debora) . mae(veronica , leon) . mae(tania , david) . mae(tania , flavia) .
11
12
13 parente(X,Y) :- pai(X,Y) .
14 parente(X,Y) :- mae(X,Y) .
15
16 descendente(X,Y) :- parente(X,Y) .
17 descendente(X,Y) :- parente(Z,Y) , descendente(X,Z) .
18
19 irmao(X,Y) :- parente(Z,X) , parente(Z,Y) , X\=Y .
20 tio(X,Y) :- parente(Z,Y) , irmao(Z,X) .
21 primo(X,Y) :- parente(Z,Y) , tio(Z,X) .
```

```
🐉 SWI-Prolog (AMD64, Multi-threaded, version 7.6.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This
Please run ?- license. for legal details.

For online help and background, visit http://www.
For built-in help, use ?- help(Topic). or ?- aproj

?- tio(X,Y).
X = luana,
Y = christiana ;
X = luana,
Y = christiana ;
X = luana,
Y = eduardo ;
X = luana,
Y = eduardo ;
X = regina,
Y = manoel ;
X = regina,
Y = manoel ;
X = debora,
Y = david ;
X = debora,
Y = david ;
X = debora,
Y = flavia ;
X = debora,
Y = flavia.

?-
```

Figura 13.1: Uma execução `swi`

# Prolog – Fuzzy Simples

- Associar a todo fato um grau de verdade
- Em toda regra, calcular um novo grau de verdade, a partir do grau de verdade obtidos
- Colocar limites nos graus de verdade, para evitar buscar regras pouco possíveis

Programa 13.2: O uso mínimo de Lógica Fuzzy em um programa Prolog comum

```
1 alto(everest,100). alto(k2,100).
2 alto(montebranco,80). alto(kilimanjaro,70).
3 alto(paodeacucar,10). alto(picodaneblina,30).
4 alto(pico31demarco,30). alto(picodabandeira,25).
5 alto(corcovado,12).
6
7 longe(everest,100). longe(k2,100).
8 longe(montebranco,10). longe(kilimanjaro,80).
9 longe(paodeacucar,0). longe(picodaneblina,100).
10 longe(pico31demarco,100). longe(picodabandeira,35).
11 longe(corcovado,0).
12
13 risco(X,R) :- alto(X,R1),longe(X,R2),R is max(R1,R2),R > 20.
```

# Flopper

```
#####
#####  ##          #####  #####  #####  #####
##      ##          ##  ##  ##  ##  ##  ##  ##
**      **          **  **  **  **  **  **  **
****    **          **  **  ****    ****    ****
**      **          **  **  **          **  **
oo      oo          oo  oo  oo          oo  oo
oo      ooooooo    ooooooo  oo          ooooooo  oo  oo

** Fuzzy Logic Programming Environment for Research **
**                               v 2.0                               **
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

**** PROGRAM MENU ****
** parse --> Parse/load a fuzzy prolog file (.fpl) **
** save  --> Parse/load/save a fuzzy prolog file.  **
** load  --> Consult a prolog file (.pl).          **
** list  --> Displays the last loaded clauses.     **
** clean --> Clean the database                    **

**** LATTICE MENU ****
** lat   --> Load a Multi-Adjoint lattice          **
** show  --> Show current Multi-Adjoint lattice    **
** ismode --> Select kind of interpretive steps    **

**** GOAL MENU ****
** intro --> Introduce a new goal (between quotes). **
** run   --> Execute a goal completely             **
** depth --> Set the maximum level of execution trees **
** leaves --> Interpret a goal                    **
** tree  --> Generate a partial execution tree     **

**** TRANSFORMATION MENU ****
** pe    --> Partial evaluation                    **
** fu    --> Fold/Unfold Transformations           **
** red   --> Reductants Calculus                   **

-----

** stop  --> Stop the execution of the parser.    **
** quit  --> Exit to desktop.                     **

-----
```

Figura 13.2: O menu de FLOPPER

Programa 13.4: Um programa simples em FLOPER (Morcillo e Moreno, 2008).

- 1  $y(\text{pedro})$  with 0.4.
- 2  $y(\text{maria})$  with 0.8.
- 3  $h(\text{pedro})$  with 0.9.
- 4  $h(\text{maria})$  with 0.3.
- 5  $e(\text{pedro})$  with 0.5.
- 6  $e(\text{maria})$  with 0.95.
- 7  $c(X) < \text{prod}(h(X) | \text{prod } e(X)) \& \text{prod } y(X)$  with 1.

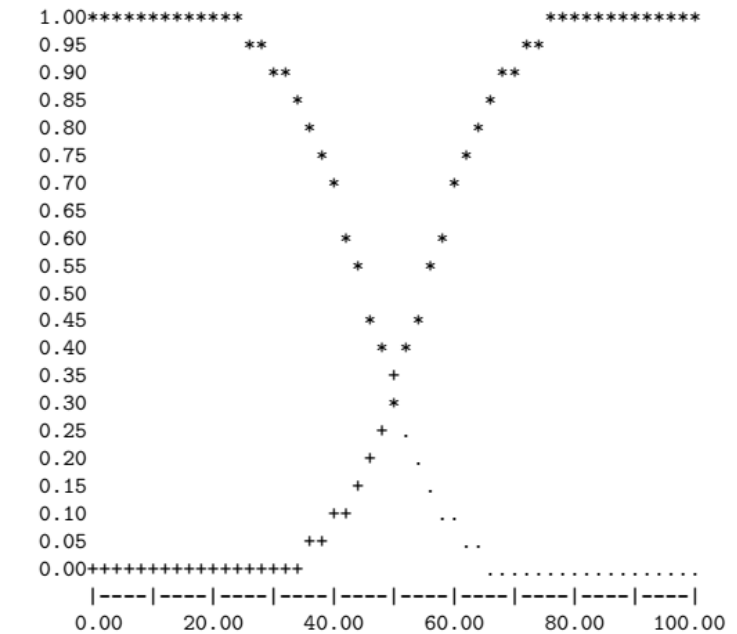
# Fuzzy Clips (1/2)

Programa 13.8: Desenhando funções de pertinência em FuzzyClips

```
1 (deftemplate temp
2   0 100 C
3   ((cold (z 20 70))
4     (hot (s 30 80))
5   )
6 )
7
8 (defrule do
9 =>
10 (plot-fuzzy-value t ".+*" nil nil
11   (create-fuzzy-value temp cold)
12   (create-fuzzy-value temp hot)
13   (fuzzy-union (create-fuzzy-value temp cold)
14                (create-fuzzy-value temp hot))
15 )
16 )
```

```
FuzzyCLIPS> (clear)
FuzzyCLIPS> (load c.fc)
Defining deftemplate: temp
Defining defrule: do +j
TRUE
FuzzyCLIPS> (reset)
FuzzyCLIPS> (run)
```

```
Fuzzy Value: temp
Linguistic Value: cold (.), hot (+), [ cold ] OR [ hot ] (*)
```



Universe of Discourse: From 0.00 to 100.00

FuzzyCLIPS>

# Fuzzy Clips (2/2)

Programa 13.9: Procurando pessoas muito altas em Fuzzy Clips

```
1 (deftemplate altura
2   100 210 centimetros
3   ((baixo (Z 110 150))
4    (medio (pi 35 170))
5    (alto (175 0) (210 1))
6  )
7 )
8
9 (deftemplate pessoa
10  (slot nome)
11  (slot alt (type FUZZY-VALUE altura))
12 )
13
14
15 (deffacts pessoas
16 (pessoa (nome mario) (alt alto))
17 (pessoa (nome carlos) (alt baixo))
18 (pessoa (nome julio) (alt medio))
19 )
20
21 (defrule medida
22   (pessoa (nome ?n) (alt ?h & very alto))
23   =>
24   (printout t ?n
25    "e muito alto , com altura aproximada de "
26     (maximum-defuzzify ?h) " "
27     (get-u-units ?h) crlf )
28 )
```

# Conclusões

# Usos de Fuzzy

- Problemas mal definidos
- Soluções baseadas em regras definidas por humano
- Tratamento de subjetividade
- Tratamento de qualquer forma de incerteza

# Vantagens de Fuzzy

- Abordagem gradativa na solução
- Mix matemática objetividade com regras e subjetividade
- Mix quantitativo com qualitativo
- Rigor matemático para temas normalmente tratados sem rigor
  - Similaridade



# Bibliografia

- Xexéo, G. (2022) Conjuntos e Lógica Fuzzy: Introdução a Teoria e Aplicações. Ciência Moderna, Rio de Janeiro
  - A bibliografia completa pode ser encontrada no livro



## Geraldo Xexéo

[xexeo@ufrj.br](mailto:xexeo@ufrj.br)



<http://xexeo.net>  
<http://xexeo.net/youtube>  
<http://line.cos.ufrj.br>  
<http://ludes.cos.ufrj.br>



Este obra está licenciado com uma Licença [Creative Commons Atribuição-NãoComercial-SemDerivações 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).