



JOSÉ LUIS COSTA

MODELAGEM E CONTROLE FUZZY

Santo André, 2017



UNIVERSIDADE FEDERAL DO ABC

CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO

JOSÉ LUIS COSTA

MODELAGEM E CONTROLE FUZZY

Orientador: Prof. Dr. Daniel Miranda Machado

Dissertação de mestrado apresentada ao Centro de
Matemática, Computação e Cognição para
obtenção do título de Mestre

ESTE EXEMPLAR CORRESPONDE A VERSÃO FINAL DA DISSERTAÇÃO
DEFENDIDA PELO ALUNO JOSÉ LUIS COSTA,
E ORIENTADA PELO PROF. DR. DANIEL MIRANDA MACHADO.

SANTO ANDRÉ, 2017

Sistema de Bibliotecas da Universidade Federal do ABC
Elaborada pelo Sistema de Geração de Ficha Catalográfica da UFABC
com os dados fornecidos pelo(a) autor(a).

Costa, José Luis
Modelagem e controle fuzzy / José Luis Costa. — 2017.

71 fls.

Orientador: Daniel Miranda Machado

Dissertação (Mestrado) — Universidade Federal do ABC, Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, Santo André, 2017.

1. Lógica Fuzzy. 2. Controle. 3. Modelagem. I. Machado, Daniel Miranda. II. Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, 2017. III. Título.

Este exemplar foi revisado e alterado em relação à versão original, de acordo com as observações levantadas pela banca no dia da defesa, sob responsabilidade única do autor e com a anuência de seu orientador.

Santo André, 28 de agosto de 2017.

Assinatura do autor: 

Assinatura do orientador: 



MINISTÉRIO DA EDUCAÇÃO
Fundação Universidade Federal do ABC
Programa de Pós-Graduação em Mestrado Profissional em Matemática
em Rede Nacional

Avenida dos Estados, 5001 – Bairro Santa Terezinha – Santo André – SP
CEP 09210-580 · Fone: (11) 4996-0017
profmat@ufabc.edu.br

FOLHA DE ASSINATURAS

Assinaturas dos membros da Banca Examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato José Luis Costa, realizada em 16 de agosto de 2017:

Prof.(a) Dr.(a) **Daniel Miranda Machado** (Universidade Federal do ABC) – Presidente

Prof.(a) Dr.(a) **Rodney Carlos Bassanezi** (Universidade Estadual de Campinas) – Membro Titular

Prof.(a) Dr.(a) **Vinicius Cifú Lopes** (Universidade Federal do ABC) – Membro Titular

Prof.(a) Dr.(a) **Cristian Favio Coletti** (Universidade Federal do ABC) – Membro Suplente

Prof.(a) Dr.(a) **Pedro Jose Catuogno** (Universidade Estadual de Campinas) – Membro Suplente

RESUMO

Este trabalho têm como objetivo demonstrar a utilização da modelagem em lógica fuzzy na tomada de decisões, em particular na escolha da taxa SELIC de modo antecipado, utilizando alguns dos parâmetros que servem para a decisão pelo COPOM (Comitê de política monetária), ao final poderemos verificar que o resultado real é compatível com o resultado fuzzy, podendo portanto este tipo de modelagem ser utilizada em outras situações. Também veremos a utilização da lógica fuzzy para controle de proximidades, quando temos uma incerteza de quanto um objeto estará perto de um obstáculo, como resultado veremos através de algumas execuções do programa que será mais adequado utilizar a lógica fuzzy do que a lógica clássica. Para essas aplicações utilizaremos as linguagens de programação R e Kturtle, sendo um diferencial pois são gratuitas e possibilitam a criação de outras aplicações. Teremos antes a demonstração da teoria da lógica fuzzy necessária para as aplicações propostas.

Palavras-chave: modelagem, fuzzy, controle

ABSTRACT

The purpose of this dissertation is to demonstrate the use of fuzzy logic modeling in decision making, particularly in the selection of the SELIC rate in advance, using some of the parameters that serve for the decision by the COPOM (Monetary Policy Committee). We verify that the actual result is compatible with the fuzzy result, so that this type of modeling can be used in other situations. We will also see the use of fuzzy logic for neighborhood control when we have an uncertainty about how much an object will be close to an obstacle, as a result we will see through some program executions that it will be more appropriate to use fuzzy logic than classical logic. For these applications we will use the programming languages R and Kturtle, being an advantage because they are free and allow the creation of other applications. We will first develop of the fuzzy logic theory required for the proposed applications.

Keywords: fuzzy, modeling, control

CONTEÚDO

Lista de Figuras	xiii
Lista de Tabelas	xv
INTRODUÇÃO	1
1 CONJUNTOS FUZZY	5
1.1 Incerteza	5
1.2 Pertinência	6
1.3 Subconjuntos Fuzzy	7
1.4 Operações com Conjuntos Fuzzy	8
1.5 Alpha-nível	12
1.6 O Princípio de Extensão de Zadeh	15
1.7 Números Fuzzy	17
1.7.1 Operações aritméticas com números fuzzy	20
2 LÓGICA FUZZY	25
2.1 Conectivos básicos da Lógica Classica	25
2.2 Conectivos básicos da Lógica Fuzzy	28
2.2.1 Operações t-norma e t-conorma	28
2.3 Raciocínio Aproximado e Variáveis Linguísticas	32
2.4 Modus Ponens e Modus Ponens Generalizado	33
2.5 Controle Fuzzy	37
2.6 Base de Regras Fuzzy	38
2.7 Controlador Fuzzy	39
2.8 O Método de Inferência de Mamdani	41
2.9 O Método de Inferência de Takagi-Sugeno-Kang	44
2.10 Métodos de Defuzzificação	47
2.10.1 Centro de gravidade, Centroide ou Centro de Área	47
2.10.2 Centro dos Máximos	48
2.10.3 Média dos Máximos	49
3 APLICAÇÕES	51
3.1 Taxa Selic	51

3.1.1	Programa para modelagem Fuzzy	52
3.1.2	Variáveis de entrada e saída	56
3.1.3	Funções de Pertinência	57
3.1.4	Regras	58
3.1.5	Inferência e defuzzificação	58
3.1.6	Conclusão	59
3.2	Uso da Lógica Fuzzy em proximidades	60
3.2.1	Programa para simulação de robô	60
3.2.2	Comentários	62
3.2.3	Uso da aplicação	63
3.2.4	Conclusão	67
	Bibliografia	71

LISTA DE FIGURAS

Figura 1	Lâmpada apagada e lâmpada acesa	6
Figura 2	Copos cheios, meio-cheios ou meio-vazios?	6
Figura 3	Exemplo função de pertinência $f(x) = e^{-(x-3)^2}$	7
Figura 4	Conjuntos Fuzzy	8
Figura 5	União Fuzzy	9
Figura 6	Intersecção Fuzzy	9
Figura 7	Operações com subconjuntos fuzzy: (a) união; (b) intersecção; (c) complemento. [7]	10
Figura 8	Altura baixa	11
Figura 9	Peso gordo	11
Figura 10	Tabela exemplo operações fuzzy	11
Figura 11	Imagem de um subconjunto fuzzy a partir do princípio de ex- tensão para uma função f [7]	16
Figura 12	Número Fuzzy Triangular	18
Figura 13	Número Fuzzy Trapezoidal	19
Figura 14	Número fuzzy em forma de sino [7]	19
Figura 15	Esquema geral de um controlador fuzzy. [7]	40
Figura 16	Saídas parciais no método de Mamdani. [7]	43
Figura 17	Saída final do controlador fuzzy de Mamdani. [7]	44
Figura 18	Saída do controlador fuzzy TSK para o exemplo [7]	46
Figura 19	Exemplo defuzzificação centroide	48
Figura 20	Defuzzificador centro de máximo. [7]	49
Figura 21	Entrada variável IPCA	56
Figura 22	Entrada variável Dólar	56
Figura 23	Entrada variável Desemprego	57

Figura 24	Saída variável SELIC	57
Figura 25	Funções de pertinência	58
Figura 26	Saída sistema fuzzy com valores de janeiro 2013	59
Figura 27	Comparação SELIC real e fuzzy	59
Figura 28	Função de aproximação	62
Figura 29	Percurso da tartaruga com desvio e calor final	63

LISTA DE TABELAS

Tabela 1	Conectivo e	25
Tabela 2	Conectivo ou	25
Tabela 3	Conectivo negação	26
Tabela 4	Conectivo implicação	26
Tabela 5	Forma Geral de uma base de regras fuzzy.	38
Tabela 6	Ilustração da obtenção da relação \mathcal{R}	41
Tabela 7	Base de regras controlador fuzzy com duas entradas e uma saída.	42
Tabela 8	Base de regras do método TSK.	44
Tabela 9	Base de duas regras do método TSK.	45
Tabela 10	Base de regras para o exemplo.	45
Tabela 11	Tabela de valores das entradas fuzzy.	68
Tabela 12	Regras de Inferência	69

AGRADECIMENTOS

Agradeço

A Deus em primeiro lugar, por ter me dado força para terminar esse curso.

A Minha família, pelo incentivo e apoio que me deram.

Aos professores da UFABC, pela dedicação e empenho, em especial ao meu orientador pela paciência e incentivo na realização deste trabalho.

Aos meus companheiros de curso, pelo companheirismo e apoio nos estudos fora da sala de aula, em especial agradeço a professora Marilene e engenheira Bete pela companhia nas viagens.

Ao PROFMAT, por ter dado oportunidade à realização deste sonho.

INTRODUÇÃO

Este trabalho tem como objetivo utilizar a lógica fuzzy através da modelagem para tomada de decisões em que temos situações onde as informações são incertas e também para controle de um robô evitando obstáculos. Mas antes o que é lógica?

Lógica vem do termo grego **logiké** que está relacionado com razão ou ciência do raciocínio, segundo o filósofo grego Aristóteles a lógica objetiva o estudo do pensamento e as regras e leis que o controlam. Podemos considerar que a lógica é o estudo de processos de inferência que a partir de antecessoras válidas seguindo regras levam a consequências verdadeiras, exemplo: na frase "Se fizer Sol vou para a praia", significa que se for verdade que o dia está com Sol com certeza eu irei para a praia.

Euler com círculos representando conjuntos e Venn com seus diagramas tornaram mais simples a lógica com sua representação geométrica. As propriedades dos conjuntos tem semelhanças com proposições lógicas, de forma a constituir a álgebra dos conjuntos, uma em particular, utilizando os princípios de Aristóteles é representada por apenas dois números, foi desenvolvida pelo matemático inglês Boole, que estudou e empregou ideias algébricas no domínio da lógica, sendo denominada de álgebra booleana ou lógica binária ou lógica a dois estados, exemplos: "sim" e "não", "ligado" e "desligado", este tipo de lógica é utilizada neste trabalho como "lógica clássica".

A lógica binária é largamente utilizada em computadores, pois é muito mais simples reconhecer apenas dois estados associados aos números **0** e **1**, do que reconhecer 10 estados diferentes para representar números decimais, para realizar as operações matemáticas existem os operadores **e**, **ou** e **negação** que podem ser implementados em circuitos lógicos.

Na implementação de certos tipos de problemas o uso da lógica clássica pode ser suficiente, mas temos situações com valores verdade imprecisos, como por exemplo o que pode significar "calor", para alguns acima de 25°C é quente, para outros pode ser acima de 30°C, poderíamos utilizar valores analógicos, que resultariam em equações matemáticas complexas que nem sempre teriam o resultado esperado. As situações incertas precisam de uma modelagem diferenciada, podemos afirmar com 100% que 50°C é quente, podendo atribuir 70% para 25°C, este tipo situação é considerado um modelo de lógica fuzzy. A lógica fuzzy é utilizada em situações em que temos um grau de certeza entre **0** e **1**.

O conceito de lógica fuzzy foi introduzido, em 1965, por Lotfi A. Zadeh, reconhecido como grande colaborador da lógica, do controle moderno e da inteligência artificial. Zadeh observou que os recursos disponíveis não eram capazes de automatizar atividades humanas relacionadas a problemas de natureza industrial, química ou biológica, que compreendessem situações ambíguas. Publicou um artigo resumindo os conceitos dos conjuntos fuzzy, revolucionando com a criação de sistemas fuzzy. Esta teoria propôs a extensão da lógica clássica através de uma função que operasse sobre o intervalo dos números reais $[0, 1]$.

Uma das principais aplicações da lógica fuzzy é em sistemas de controle, onde os termos linguísticos são transformados em implicações lógicas, como regras "Se ... então ...", que são utilizadas em controladores fuzzy, sendo que as informações e as regras são fornecidas por especialistas ou extraídas de dados numéricos, temos como exemplo de controladores: os controles fuzzy de plantas industriais, refinarias, processos biológicos e químicos, trocador de calor, máquina diesel, tratamento de água e sistema de operação automática de trens.

Na atualidade a lógica fuzzy é aplicada em sistemas de controle e de suporte à decisão, onde a descrição do problema não pode ser feita de forma precisa. Esta metodologia tem sido aplicada em análise de dados, construção de sistemas especialistas, controle e otimização, reconhecimento de padrões; portanto, desde controle de aeronaves a máquinas de lavar roupa, de análise de estoques de venda a modelos genéticos, de reconhecimento de escrita a ajuste de foco de câmeras de vídeo [12].

A tese está organizada do seguinte modo:

No capítulo 1 apresentamos a definição de lógica fuzzy, a comparação com a lógica clássica, operações com os subconjuntos fuzzy e definição de α -nível, veremos como transformar números reais em subconjuntos fuzzy e construção de funções de pertinência e seus diferentes formatos.

No capítulo 2 será dada a noção da lógica fuzzy, seus conectivos e tabelas verdades e transformações de uma sentença em lógica fuzzy através do Modus Ponens, com esta teoria poderemos construir um modelador que transforme entradas reais em valores fuzzy que serão manipulados segundo regras estabelecidas inferindo resultados na saída que poderão voltar a serem reais através do processo chamado defuzzificação.

Finalmente no capítulo 3 utilizaremos o método do capítulo 2 para desenvolver um sistema de modelagem fuzzy para a determinação da taxa SELIC e uma aplicação para uso didático com os alunos do percurso de um robô evitando obstáculos utilizando a lógica fuzzy.

Para melhor compreensão do assunto abordado tenho como sugestão a leitura dos manuais dos programas R e Kturtle, que podem ser acessados nos próprios programas, a leitura da bibliografia ao final da tese, assim como um conhecimento básico da lógica clássica.

CONJUNTOS FUZZY

Veremos neste capítulo o que são os conjuntos fuzzy, suas operações e o que é um número fuzzy, os conceitos servirão de base para efetuarmos as operações necessárias a serem utilizadas em um controlador fuzzy, a teoria desse capítulo têm como referência o livro [7].

1.1 INCERTEZA

O que é incerteza? Segundo alguns dicionários é falta de certeza, estado do que é incerto, hesitação, dúvida. Na nossa vida temos algumas certezas como uma lâmpada está acesa ou apagada 1, um objeto está dentro de uma caixa ou fora dela, numa partida de tênis ou você vence ou perde, podemos associar em matemática esses dois estados aos algarismos 0 e 1, esta associação facilita em muito o uso nos computadores, pois os mesmos têm mais facilidade de distinguir melhor apenas esses dois estados, sendo que para o controle dos computadores é largamente utilizada a lógica clássica.

Mas e para os casos em que existem dúvidas? Como altura de uma pessoa, pois se uma pessoa de 1,70 metros vivendo numa aldeia de pigmeus pode ser considerada alta, mas vivendo na Holanda pode ser considerada baixa; o que pode ser considerado quente? 20°, 25°, 30°, 35°, 40°; uma pessoa pode ser rica morando em uma favela ou pobre morando em um bairro de empresários, na figura 2 quando podemos dizer que o copo está cheio?. Para esses casos a lógica clássica é difícil de ser usada, para essa situação foi introduzida a teoria da lógica fuzzy em 1965 pelo matemático Lotfi Asker Zadeh.



Figura 1: Lâmpada apagada e lâmpada acesa



Figura 2: Copos cheios, meio-cheios ou meio-vazios?

1.2 PERTINÊNCIA

Na lógica clássica (também chamada lógica crisp) um elemento pertence ou não pertence a um conjunto, não podendo estar contido parcialmente, geralmente associamos o 0 para "não pertence" e o 1 para "pertence".

Na lógica clássica um elemento pertence ou não pertence a um conjunto, em termos matemático podemos representar com o algarismo **1** que um elemento está (pertence a um conjunto) e o algarismo **0** para um elemento que não está em um conjunto. A relação de pertinência temos associado a noção de função característica.

Definição 1.1. Seja U um conjunto e A um subconjunto de U . A *função característica* de A é dada por

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

Temos que χ_A é uma função cujo domínio é U e a imagem está contida no conjunto $\{0,1\}$, sendo $\chi_A = 1$ indica que x está em A , e $\chi_A = 0$ indica que x não está em A . Assim a função característica descreve completamente o conjunto A já que tal função indica quais elementos de U são também elementos de A . Porém, existem casos em que a pertinência entre elementos e conjuntos não é precisa, isto é, não sabemos dizer se um elemento pertence efetivamente a um conjunto ou não.

1.3 SUBCONJUNTOS FUZZY

Existem situações em que um elemento não está totalmente em um conjunto, por exemplo, a cor verde está no conjunto amarelo? está em parte, pois a cor verde é a combinação da cor azul com a cor amarelo, portanto podemos dizer que a cor verde pertence parcialmente ao conjunto amarelo ou têm um grau de pertinência ao conjunto amarelo.

Começaremos por um exemplo:

Exemplo 1.1. (Números próximos de 3) Imagine um aluno do PROFMAT que faça o curso em Santo André e que more em Santos, sendo que o curso começa às 3:00hs da tarde, ele irá de carro para não atrasar, mas ele nunca chegará exatamente às 3:00hs da tarde, devido a uma particularidade do tempo. Supondo dar tudo certo, ele poderá chegar mais próximo do horário previsto, mas a vida é cheia de imprevistos, poderá chover e o trânsito ficar lento (lento quanto?), a concessionária poderá fechar uma das vias para manutenção, conseqüentemente terá acúmulo de carros que atrapalhará o trânsito (em quanto?), portanto neste caso temos informações incertas que podem auxiliar a uma tomada de decisão para sair antes e não atrasar, a lógica fuzzy auxilia em situações como essa, onde temos informações incertas. Podemos ver em 3 o conjunto F dos números reais *próximos* de 3:

$$F = \{x \in \mathbb{R} : x \text{ é próximo de } 3\}.$$

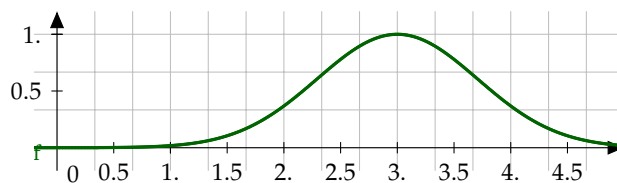


Figura 3: Exemplo função de pertinência $f(x) = e^{-(x-3)^2}$

Se definirmos a função $\varphi_F : \mathbb{R} \rightarrow [0, 1]$, que associa cada x real ao valor de proximidade ao ponto 3 pela expressão

$$\varphi_F(x) = \begin{cases} (1 - |x - 3|) & \text{se } 2 \leq x \leq 4 \\ 0 & \text{se } x \notin [2, 4] \end{cases}$$

então o subconjunto fuzzy F dos pontos *próximos* de 3, caracterizado por φ_F , é tal que $\varphi_F(3,01) = 0,99$ e $\varphi_F(5) = 0$. Neste caso, dizemos que $x = 3,01$ é um ponto próximo de 3 com grau de proximidade 0,99, e $x = 5$ não é próximo de 3. Poderíamos também definir outra função de proximidade a 3. Conforme descrito abaixo:

$$\nu_F(x) = \exp[-(x - 3)^2],$$

com $x \in \mathbb{R}$, então os elementos do conjunto fuzzy F , caracterizado pela função ν_F , teriam outros graus de pertinência: $\nu_F(3,01) = 0,9999$ e $\nu_F(5) = 0,018315$. Vemos que a caracterização de proximidade é subjetiva e depende da função de pertinência que pode ser dada por uma infinidade de maneiras diferentes, dependendo de como se quer avaliar o termo "próximo".

Isso motiva a seguinte definição:

Definição 1.2. Seja U um conjunto (clássico); um *subconjunto fuzzy* F de U é caracterizado por uma função

$$\varphi_F : U \rightarrow [0, 1],$$

pré-fixada, chamada *função de pertinência* do subconjunto fuzzy F . O índice F na função de pertinência é usado em analogia à função característica de subconjunto clássico, conforme definição 1.1.

1.4 OPERAÇÕES COM CONJUNTOS FUZZY

Considere para exemplificação a figura 4.

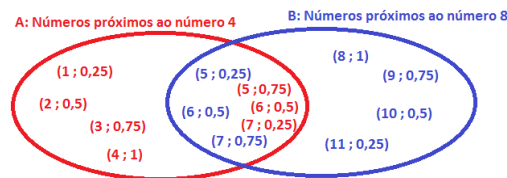


Figura 4: Conjuntos Fuzzy

Considerando que na lógica clássica a união de dois conjuntos tem como resultado um conjunto com todos os elementos desses conjuntos, e valores maiores(máximos)

englobam os valores menores e o conjunto resultado somente será vazio se não houver nenhum elemento nesses conjuntos, ver figura 5, podemos dizer que a operação união fuzzy é uma extensão da lógica clássica.

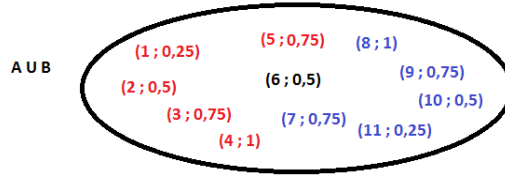


Figura 5: União Fuzzy

Definição 1.3. (União) A união entre A e B é o subconjunto fuzzy de U cuja função de pertinência é dada por

$$\varphi_{(A \cup B)}(x) = \max\{\varphi_A(x), \varphi_B(x)\}, x \in U.$$

Observamos que esta definição é uma extensão do caso clássico. De fato, quando A e B são subconjuntos clássicos de U temos:

$$\begin{aligned} \max\{\chi_A(x), \chi_B(x)\} &= \begin{cases} 1 & \text{se } x \in A \text{ ou } x \in B \\ 0 & \text{se } x \notin A \text{ e } x \notin B \end{cases} \\ &= \begin{cases} 1 & \text{se } x \in A \cup B \\ 0 & \text{se } x \notin A \cup B \end{cases} \\ &= \chi_{A \cup B}(x), \quad x \in U. \end{aligned}$$

Considerando que na lógica clássica a intersecção de dois conjuntos tem como resultado um conjunto com todos os elementos em comum desses conjuntos, e os valores menores(mínimos) estão contidos em valores maiores e o conjunto resultado somente será máximo se os elementos desses conjuntos forem também máximos, ver figura 6, podemos dizer que a operação intersecção fuzzy é uma extensão da lógica clássica.

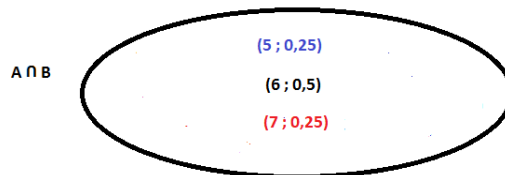


Figura 6: Intersecção Fuzzy

Definição 1.4. (Intersecção) A *Intersecção* entre A e B é o subconjunto fuzzy de U cuja função de pertinência é dada por

$$\varphi_{(A \cap B)}(x) = \min\{\varphi_A(x), \varphi_B(x)\}, x \in U.$$

Considerando que na lógica clássica a união de um conjunto com seu complementar produz o conjunto universo, o mesmo acontece com a operação complementar fuzzy, portanto pode ser considerada também uma extensão da lógica clássica.

Definição 1.5. (Complementar de subconjuntos fuzzy) O *Complementar* de A é o subconjunto fuzzy A' de U cuja função de pertinência é dada por

$$\varphi_{(A')}(x) = 1 - \varphi_A(x), x \in U.$$

Podemos verificar graficamente as operações na figura 7.

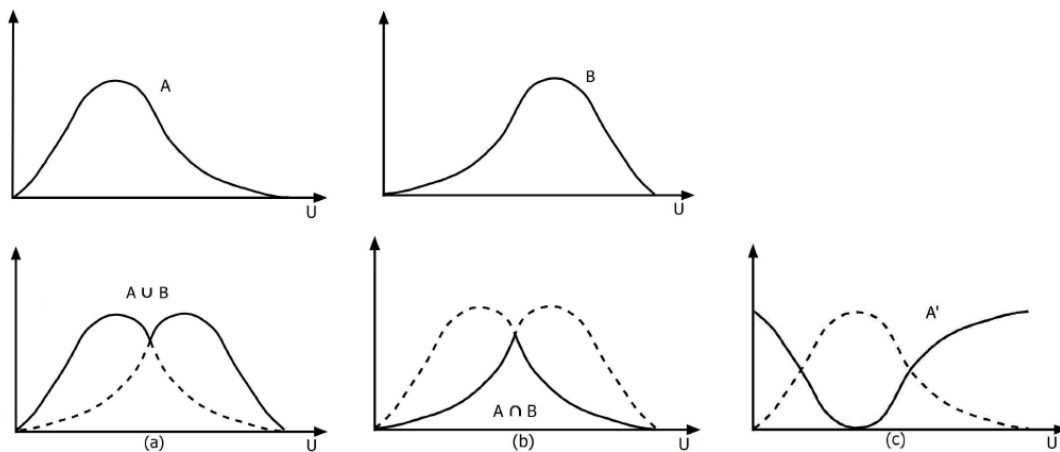


Figura 7: Operações com subconjuntos fuzzy: (a) união; (b) intersecção; (c) complemento. [7]

Exemplo 1.2. Vamos fazer uma comparação do IMC (Índice de Massa Corpórea), calculado usando a fórmula $IMC = \frac{peso}{(altura)^2}$, utilizando as operações fuzzy com os conjuntos altura(ser baixo) e peso(ser gordo) utilizando as funções de pertinência das figuras 8 e 9.

Verificamos na tabela 10 que o uso de operações dos subconjuntos fuzzy pode ser um método alternativo, no exemplo, a pessoa 1 diverge do IMC devido a estar nos extremos, observamos também que a operação $A \cap A'$ é diferente da lógica clássica, onde $A \cap A' = \emptyset$.

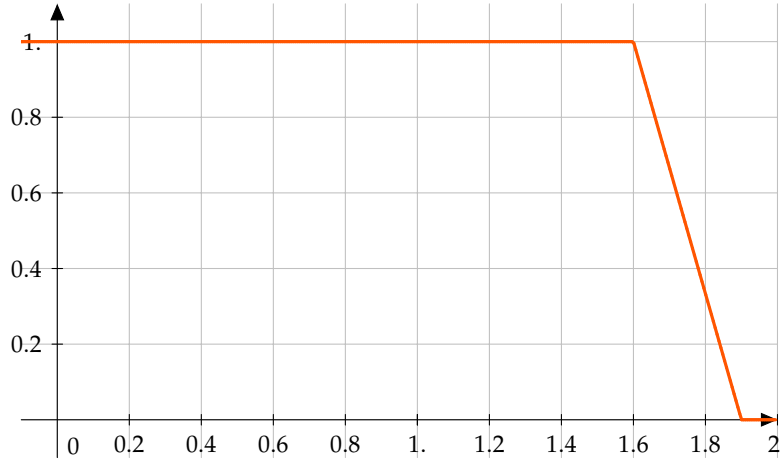


Figura 8: Altura baixa

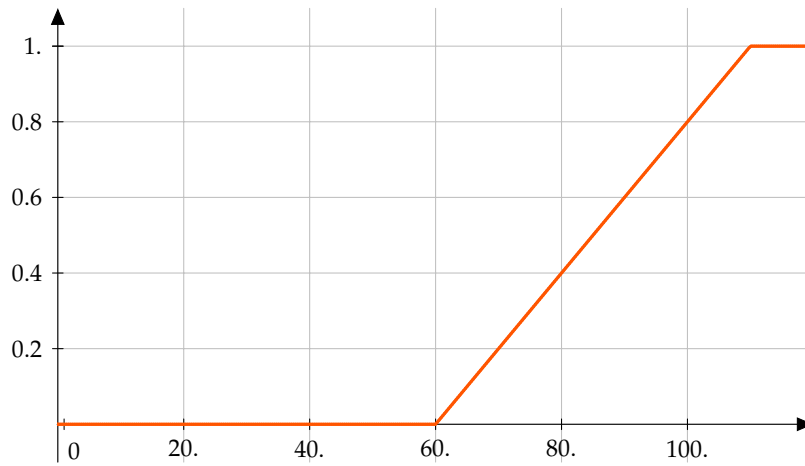


Figura 9: Peso gordo

Pessoa	Peso	Altura	IMC	Grau Peso (A)	Grau Altura (B)	$A \cup B$	A'	$A \cap A'$	$A \cap B$
1	120	1.93	32.21	1	0	1	0	0	0
2	90	1.80	27.77	0.6	0.33	0.6	0.4	0.4	0.33
3	75	1.75	24.49	0.3	0.5	0.5	0.7	0.3	0.3
4	80	1.68	28.34	0.4	0.73	0.73	0.6	0.4	0.4

Figura 10: Tabela exemplo operações fuzzy

Definição 1.6. Os subconjuntos A e B de U são iguais se suas funções de pertinência coincidem, isto é, se $\varphi_A(x) = \varphi_B(x)$ para todo $x \in U$.

Proposição 1.7. As operações entre subconjuntos fuzzy satisfazem as seguintes propriedades:

- $A \cup B = B \cup A$
- $A \cap B = B \cap A$
- $A \cup (B \cap C) = (A \cup B) \cap C$
- $A \cap (B \cup C) = (A \cap B) \cup C$
- $A \cup A = A$
- $A \cap A = A$
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- $A \cap \emptyset = \emptyset$ e $A \cup \emptyset = A$
- $A \cap U = A$ e $A \cup U = U$
- $(A \cup B)' = A' \cap B'$ e $(A \cap B)' = A' \cup B'$ (leis de DeMorgan).

Demonstração. A demonstração de cada propriedade é uma aplicação imediata das propriedades de máximo e mínimo entre funções, isto é,

$$\max[\varphi(x), \psi(x)] = \frac{1}{2}[\varphi(x) + \psi(x) + |\varphi(x) - \psi(x)|]$$

$$\min[\varphi(x), \psi(x)] = \frac{1}{2}[\varphi(x) + \psi(x) - |\varphi(x) - \psi(x)|]$$

onde, φ e ψ são funções com imagens em \mathbb{R} . Demonstraremos apenas as duas primeiras operações, as outras podem ser feitas de modo análogo.

$$A \cup B = \max[\varphi_A(x), \varphi_B(x)] = \frac{1}{2}[\varphi_A(x) + \varphi_B(x) + |\varphi_A(x) - \varphi_B(x)|],$$

$$\frac{1}{2}[\varphi_B(x) + \varphi_A(x) + |\varphi_B(x) - \varphi_A(x)|] = \max[\varphi_B(x), \varphi_A(x)] = B \cup A$$

$$A \cap B = \min \varphi_A(x), \varphi_B(x) = \frac{1}{2}[\varphi_A(x) + \varphi_B(x) - |\varphi_A(x) - \varphi_B(x)|],$$

$$\frac{1}{2}[\varphi_B(x) + \varphi_A(x) - |\varphi_B(x) - \varphi_A(x)|] = \min \varphi_B(x), \varphi_A(x) = B \cap A$$

□

1.5 ALPHA-NÍVEL

Os elementos de um subconjunto fuzzy A de U possuem uma certa ordem que podemos classificar por graus. Um elemento x de U está em uma classe se seu grau de pertinência

é maior que um determinado valor limiar ou nível $\alpha \in [0, 1]$ que define aquela classe. O conjunto clássico de tais elementos é um α -nível de A , denotado por $[A]^\alpha$.

Definição 1.8. (α -nível). Seja A um subconjunto fuzzy de U e $\alpha \in [0, 1]$.

O α -nível de A é o subconjunto clássico de U definido por

$$[A]^\alpha = \{x \in U : \varphi_A(x) \geq \alpha\} \text{ para } 0 < \alpha \leq 1.$$

O nível 0 de um subconjunto fuzzy A é definido como sendo o menor subconjunto (clássico) fechado de U que contém o conjunto suporte de A . Numa linguagem matemática, $[A]^0$ é o fecho do suporte de A e é indicado por $\overline{\text{supp}}A$.

Teorema 1.9. *Sejam A e B subconjuntos fuzzy de U . Uma condição necessária e suficiente para que $A = B$ é que $[A]^\alpha = [B]^\alpha$, para todo $\alpha \in [0, 1]$.*

Demonstração. É claro que $A = B \Rightarrow [A]^\alpha = [B]^\alpha$ para todo $\alpha \in [0, 1]$. Suponhamos agora que $[A]^\alpha = [B]^\alpha$ para todo $\alpha \in [0, 1]$. Se $A \neq B$ então existe $x \in U$ tal que $\varphi_A(x) \neq \varphi_B(x)$. Logo, temos $\varphi_A(x) < \varphi_B(x)$ ou $\varphi_A(x) > \varphi_B(x)$. Supondo $\varphi_A(x) > \varphi_B(x)$, podemos concluir que $x \in [A]^{\varphi_A(x)}$ e $x \notin [B]^{\varphi_A(x)}$ e, portanto, $[A]^{\varphi_A(x)} \neq [B]^{\varphi_A(x)}$, o que contradiz a hipótese $[A]^\alpha = [B]^\alpha$ para todo $\alpha \in [0, 1]$. De maneira análoga chegamos a uma contradição se admitirmos que $\varphi_A(x) < \varphi_B(x)$. \square

Uma consequência deste teorema é a relação existente entre a função de pertinência de um subconjunto fuzzy e as funções características de seus α -níveis.

Corolário 1.10. *A função de pertinência φ_A de um conjunto fuzzy A pode ser expressa em termos de funções características de seus α -níveis, isto é,*

$$\varphi_A(x) = \sup_{\alpha \in [0, 1]} \min[\alpha, \chi_{[A]^\alpha}(x)], \text{ onde } \chi_{[A]^\alpha}(x) = \begin{cases} 1 & \text{se } x \in [A]^\alpha \\ 0 & \text{se } x \notin [A]^\alpha \end{cases}$$

O teorema a seguir é de extrema importância no estudo da teoria dos conjuntos fuzzy e indica uma condição suficiente para que uma família de subconjuntos clássicos de U seja formada por α -níveis de um subconjunto fuzzy de U .

Teorema 1.11. *(Teorema de Representação de Negoita e Ralescu) Seja $A_\alpha, \alpha \in [0, 1]$, uma família de subconjuntos clássicos de U de modo que se verifiquem*

- (i) $\cup_{\alpha} A_\alpha \subset A_0$ com $\alpha \in [0, 1]$;
- (ii) $A_\alpha \subset A_\beta$ se $\beta \leq \alpha$;
- (iii) $A_\alpha = \bigcap_{k \geq 0} A_{\alpha_k}$ se α_k convergir para α com $\alpha_k \leq \alpha$.

Nestas condições existe um único subconjunto fuzzy A de U cujos α -níveis são exatamente os subconjuntos clássicos A_α , isto é,

$$[A]^\alpha = A_\alpha.$$

Demonstração. Para cada $x \in U$, o valor da função de pertinência de A como sendo

$$\varphi_A(x) = \sup\{\alpha \in [0, 1] : x \in A_\alpha\}.$$

□

Usando a definição de α -nível podemos elencar as seguintes propriedades:

$$[A \cup B]^\alpha = [A]^\alpha \cup [B]^\alpha,$$

$$[A \cap B]^\alpha = [A]^\alpha \cap [B]^\alpha.$$

Por outro lado, como em geral $[A]^\alpha \cup [A']^\alpha \neq U$, então $[A']^\alpha \neq ([A]^\alpha)'$.

Definição 1.12. Um subconjunto fuzzy é dito *normal* se todos seus α -níveis forem não vazios, ou seja, se $[A]^1 \neq \emptyset$.

Lembrando que o suporte do subconjunto fuzzy A é o conjunto clássico

$$\text{supp}A = \{x \in U : \varphi_A(x) > 0\},$$

é comum descrever A com a seguinte notação

$$A = \varphi_A(x_1)/x_1 + \varphi_A(x_2)/x_2 + \cdots = \sum_{i=1}^{\infty} \varphi_A(x_i)/x_i,$$

quando o subconjunto fuzzy A tem suporte enumerável, e

$$A = \varphi_A(x_1)/x_1 + \varphi_A(x_2)/x_2 + \cdots + \varphi_A(x_n)/x_n = \sum_{i=1}^n \varphi_A(x_i)/x_i,$$

se A tem suporte finito, isto é, $\text{supp}A = \{x_1, x_2, \dots, x_n\}$.

Vale observar que a notação $\varphi_A(x_i)/x_i$ não indica "divisão". É apenas uma forma de visualizar um elemento x_i e seu respectivo grau de pertinência $\varphi_A(x_i)$. Também, aqui, o símbolo "+" na notação não significa "adição", bem como \sum não significa somatório. É apenas uma forma de conectar os elementos de U que estão em A com seus respectivos graus.

1.6 O PRINCÍPIO DE EXTENSÃO DE ZADEH

Podemos estender a teoria de conjuntos clássica para a teoria fuzzy, utilizaremos para isso o método de extensão proposto por Zadeh, também conhecido como *Princípio de Extensão*.

O Princípio da Extensão de Zadeh para uma função $f : X \rightarrow Z$ tem por objetivo indicar como deve ser a imagem de um subconjunto fuzzy A de X por meio de f . É de se esperar que esta imagem seja um subconjunto fuzzy de Z .

Definição 1.13. (Princípio de Extensão de Zadeh) Sejam f uma função tal que $f : X \rightarrow Z$ e A um subconjunto fuzzy de X . A *extensão de Zadeh* de f é a função \hat{f} que, aplicada a A , fornece o subconjunto fuzzy $\hat{f}(A)$ de Z , cuja função de pertinência é dada por

$$\varphi_{\hat{f}(A)}(z) = \begin{cases} \sup_{f^{-1}(z)} \varphi_A(x) & \text{se } f^{-1}(z) \neq \emptyset \\ 0 & \text{se } f^{-1}(z) = \emptyset \end{cases} \quad (1.1)$$

onde $f^{-1}(z) = \{x; f(x) = z\}$ denomina-se a *pré-imagem* de z . Podemos observar que se f for uma função bijetora, então

$$\{x : f(x) = z\} = \{f^{-1}(z)\},$$

em que f^{-1} é a função inversa de f .

Observamos que se A é um subconjunto fuzzy de X , com função de pertinência φ_A , e se f é bijetora, então, a função de pertinência de $\hat{f}(A)$ é dada por

$$\varphi_{\hat{f}(A)}(z) = \sup_{\{x: f(x)=z\}} \varphi_A(x) = \sup_{\{x \in f^{-1}(z)\}} \varphi_A(x) = \varphi_A(f^{-1}(z)). \quad (1.2)$$

O processo gráfico para a obtenção de extensão \hat{f} de f está ilustrada na figura 11, no caso em que f for bijetora.

Note que se f for injetora então $z = f(x)$ pertence ao subconjunto fuzzy $\hat{f}(A)$, com o mesmo grau α com que x pertence a A . Isto pode não ocorrer se f não for injetora.

Seja $f : X \rightarrow Z$ uma função injetora e A um subconjunto fuzzy de X , enumerável (ou finito), e dado por

$$A = \sum_{i=1}^{\infty} \varphi_A(x_i)/x_i.$$

Então, o Princípio de Extensão garante que $\hat{f}(A)$ é um subconjunto fuzzy de Z , dado por

$$\hat{f}(A) = \hat{f}\left(\sum_{i=1}^{\infty} \varphi_A(x_i)/x_i\right) = \sum_{i=1}^{\infty} \varphi_A(x_i)/f(x_i).$$

Portanto, a imagem de A por f pode ser deduzida do conhecimento das imagens de x_i por f . O grau de pertinência de $z_i = f(x_i)$ em $\hat{f}(A)$ é o mesmo de x_i em A .

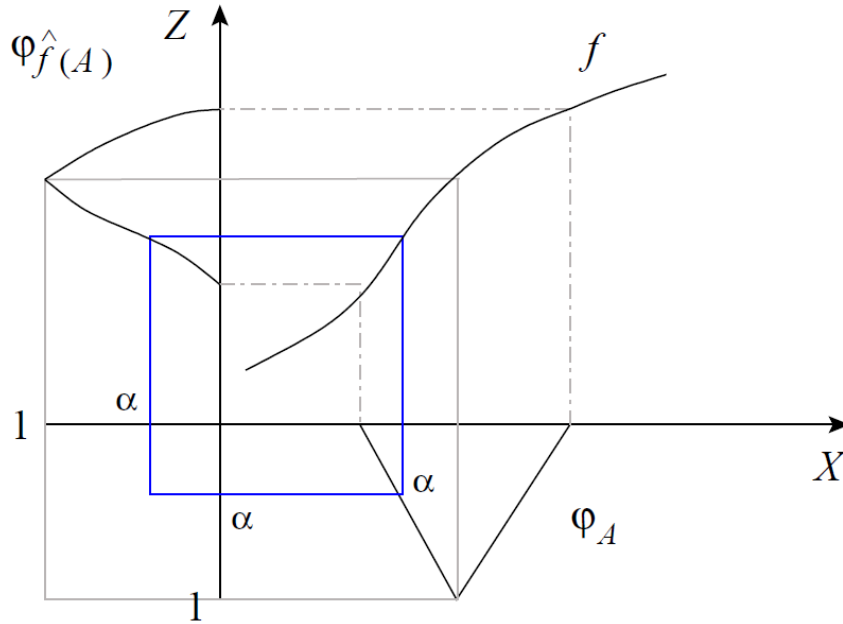


Figura 11: Imagem de um subconjunto fuzzy a partir do princípio de extensão para uma função f [7]

Teorema 1.14. *Sejam $f : X \rightarrow Z$ uma função contínua e A um subconjunto fuzzy de X . Então, para todo $\alpha \in [0, 1]$ vale*

$$[\hat{f}(A)]^\alpha = f([A]^\alpha). \tag{1.3}$$

Este resultado implica que os α -níveis do conjunto fuzzy, obtidos pelo Princípio de Extensão de Zadeh, coincidem com as imagens dos α -níveis pela função crisp.

Definição 1.15. *Sejam $f : X \times Y \rightarrow Z$ e A e B subconjunto fuzzy de X e Y , respectivamente. A extensão \hat{f} de f , aplicada a A e B , é o subconjunto fuzzy $\hat{f}(A, B)$ de Z , cuja função de pertinência é dada por:*

$$\varphi_{\hat{f}(A,B)}(z) = \begin{cases} \sup_{f^{-1}(z)} \min[\varphi_A(x), \varphi_B(y)] & \text{se } f^{-1}(z) \neq \emptyset \\ 0 & \text{se } f^{-1}(z) = \emptyset \end{cases}, \tag{1.4}$$

onde $f^{-1} = \{(x, y) : f(x, y) = z\}$.

1.7 NÚMEROS FUZZY

Na nossa vida, quando efetuamos alguns tipos de medições como por exemplo o nosso peso, conseguimos um valor que é impreciso, pois depende de muitos fatores como tipo de balança, ajustes, como é feita a medição, etc; na verdade o que temos é um valor "em torno de", e optamos por usar um valor preciso (um número real) p para o peso. Matematicamente podemos indicar a expressão *em torno de* p por um subconjunto fuzzy A cujo domínio da função de pertinência φ_A é o conjunto dos números reais, podemos esperar que $\varphi_A(p) = 1$, sendo que escolhemos os números reais como domínio pois os valores para *peso* são números reais.

Definição 1.16. (Números Reais) Um subconjunto fuzzy A é chamado de *número fuzzy* quando o conjunto universo no qual φ_A está definida, é o conjunto dos números reais \mathbb{R} e satisfaz às condições:

- (i) todos os α -níveis de A são não vazios, com $0 \leq \alpha \leq 1$;
- (ii) todos os α -níveis de A são intervalos fechados de \mathbb{R} ;
- (iii) $\text{supp}A = \{x \in \mathbb{R} : \varphi_A(x) > 0\}$ é limitado.

Vamos denotar os α -níveis do número fuzzy A por

$$[A]^\alpha = [a_1^\alpha, a_2^\alpha].$$

Observamos que todo número real r é um número fuzzy particular cuja função de pertinência é a sua função característica:

$$\chi_r(x) = \begin{cases} 1 & \text{se } x = r \\ 0 & \text{se } x \neq r \end{cases}.$$

Denotaremos $\chi_r(x)$ apenas por \hat{r} .

A família dos números fuzzy será indicada por $F(\mathbb{R})$ e, de acordo com o observado acima, o conjunto de números reais \mathbb{R} é um subconjunto (clássico ou crisp) de $F(\mathbb{R})$.

Os números fuzzy mais comuns são os *triangulares*, *trapezoidais* e os *em forma de sino*.

Definição 1.17. Um número fuzzy A é dito *triangular* se sua função de pertinência é da forma

$$\varphi_A(x) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x-a}{u-a} & \text{se } a < x \leq u \\ \frac{x-b}{u-b} & \text{se } u < x \leq b \\ 0 & \text{se } x \geq b \end{cases}. \quad (1.5)$$

O gráfico da função de pertinência de um número fuzzy triangular tem a forma de um triângulo, tendo como base o intervalo $[a, b]$ e, como único vértice fora desta base, o ponto $(u, 1)$.

Deste modo, os números reais a, u e b definem o número fuzzy triangular A que será denotado pela terna ordenada $(a; u; b)$ ou por $a/u/b$.

Os α -níveis desses números fuzzy têm a seguinte forma simplificada

$$[a_1^\alpha, a_2^\alpha] = [(u - a)\alpha + a, (u - b)\alpha + b] \tag{1.6}$$

para todo $\alpha \in [0, 1]$.

Note que o gráfico 12 de um número fuzzy triangular não é necessariamente simétrico já que $b - u$ pode ser diferente $u - a$, porém, $\varphi_A(u) = 1$. Podemos dizer que o número fuzzy A é um modelo matemático razoável para a expressão linguística "perto de u ". Para a expressão "em torno de u " espera-se uma simetria. A imposição de simetria acarreta uma simplificação na definição de número fuzzy triangular. De fato, seja u simétrico em relação a a e b , isto é, $u - a = b - u = \delta$. Neste caso,

$$\varphi_A(x) = \begin{cases} 1 - \frac{|x-u|}{\delta} & \text{se } u - \delta \leq x \leq u + \delta \\ 0 & \text{caso contrário} \end{cases}$$

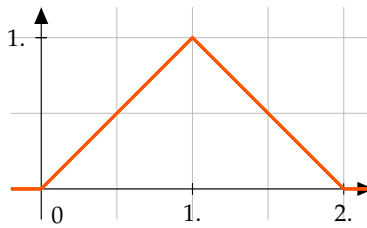


Figura 12: Número Fuzzy Triangular

Definição 1.18. Um número fuzzy A é dito *trapezoidal* 13 se sua função de pertinência tem a forma de um trapézio e é dada por

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

Os α -níveis de um conjunto fuzzy trapezoidal são os intervalos

$$[a_1^\alpha, a_2^\alpha] = [(b - a)\alpha + a, (c - d)\alpha + d] \tag{1.7}$$

para todo $\alpha \in [0, 1]$.

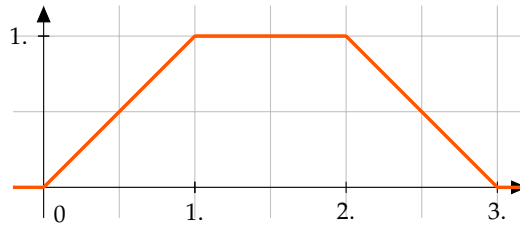


Figura 13: Número Fuzzy Trapezoidal

Definição 1.19. Um número fuzzy tem *forma de sino* se a função de pertinência for suave e simétrica em relação a um número real. A seguinte função de pertinência tem estas propriedades para u , a e δ dados (veja a figura 14).

$$\varphi_A(x) = \begin{cases} \exp(-(\frac{x-u}{a})^2) & \text{se } u - \delta \leq x \leq u + \delta \\ 0 & \text{caso contrário} \end{cases} .$$

Os α -níveis dos números fuzzy em forma de sino são os intervalos:

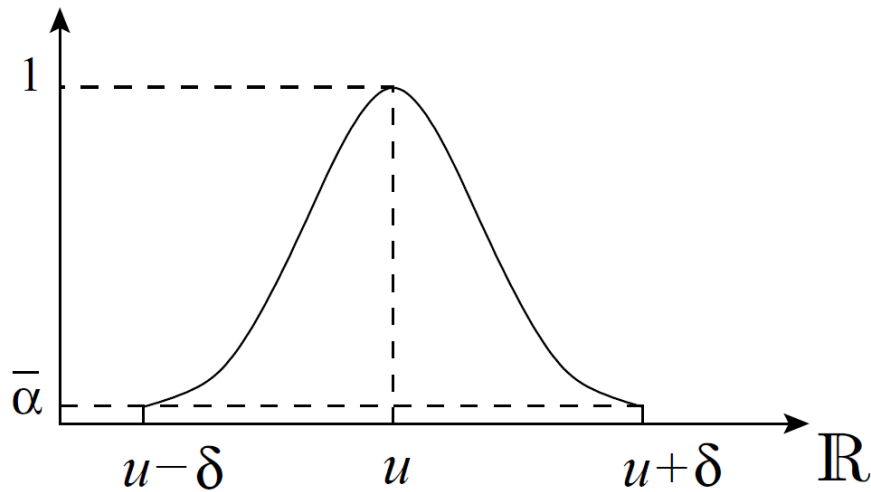


Figura 14: Número fuzzy em forma de sino [7]

$$[a_1^\alpha, a_2^\alpha] = \begin{cases} [u - \sqrt{\ln(\frac{1}{\alpha^{a^2}})}, u + \sqrt{\ln(\frac{1}{\alpha^{a^2}})}] & \text{se } \alpha \geq \bar{\alpha} = e^{-(\frac{\delta}{a})^2} \\ [u - \delta, u + \delta] & \text{se } \alpha < \bar{\alpha} = e^{-(\frac{\delta}{a})^2} \end{cases} . \tag{1.8}$$

1.7.1 Operações aritméticas com números fuzzy

As operações aritméticas envolvendo números fuzzy estão estreitamente ligadas às operações aritméticas intervalares. Abordaremos alguns intervalos fechados da reta real \mathbb{R} .

Operações aritméticas intervalares

Considere λ um número real e, A e B dois intervalos fechados da reta dados por

$$A = [a_1, a_2] \text{ e } B = [b_1, b_2].$$

Veremos a seguir as operações intervalares com números fuzzy, que podem servir na definição das operações aritméticas com números fuzzy.

Definição 1.20. (Operações Intervalares) As operações aritméticas entre intervalos pode ser definidas como:

(a) A soma entre A e B é o intervalo

$$A + B = [a_1 + b_1, a_2 + b_2].$$

(b) A diferença entre A e B é o intervalo

$$A - B = [a_1 - b_2, a_2 - b_1].$$

(c) A multiplicação de A por um escalar λ é o intervalo

$$\lambda A = \begin{cases} [\lambda a_1, \lambda a_2] & \text{se } \lambda \geq 0 \\ [\lambda a_2, \lambda a_1] & \text{se } \lambda < 0 \end{cases}.$$

(d) A multiplicação de A por B é o intervalo

$$A \cdot B = [\min P, \max P], \text{ onde } P = \{a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2\}.$$

(e) A divisão de A por B , se $0 \notin B$, é o intervalo

$$A/B = [a_1, a_2] \cdot \left[\frac{1}{b_2}, \frac{1}{b_1} \right].$$

Teorema 1.21. (Princípio de extensão para intervalos da reta). Sejam A e B dois intervalos fechados de \mathbb{R} , e \otimes uma das operações aritméticas entre números reais. Então

$$\chi_{A \otimes B}(x) = \sup_{\{(y,z): y \otimes z = x\}} \min[\chi_A(y), \chi_B(z)]$$

Demonstração. É muito simples verificar que

$$\min(\chi_A(y), \chi_B(z)) = \begin{cases} 1 & \text{se } y \in A \text{ e } z \in B \\ 0 & \text{se } y \notin A \text{ ou } z \notin B \end{cases}.$$

Assim, para o caso da soma ($\otimes = +$), temos

$$\sup_{\{(y,z):y+z=x\}} \min(\chi_A(y), \chi_B(z)) = \begin{cases} 1 & \text{se } x \in A + B \\ 0 & \text{se } x \notin A + B \end{cases}.$$

Os demais casos podem ser obtidos de maneira análoga. \square

Uma consequência importante do Teorema 1.21, para as operações com números fuzzy, é o seguinte corolário:

Corolário 1.22. *Os α -níveis do conjunto crisp $A + B$, com função característica $\chi_{(A+B)}$, são dados por*

$$[A + B]^\alpha = A + B$$

para todo $\alpha \in [0, 1]$.

Demonstração. Lembrando que os intervalos A e B são particulares subconjuntos fuzzy da reta real, o resultado é uma consequência imediata da definição de função característica de um conjunto clássico. \square

As operações aritméticas para números fuzzy serão definidas a partir do Princípio de Extensão para conjuntos fuzzy. Na verdade, são casos particulares do Princípio de Extensão em que as funções a serem estendidas são as operações tradicionais para números reais.

Operações aritméticas com números fuzzy

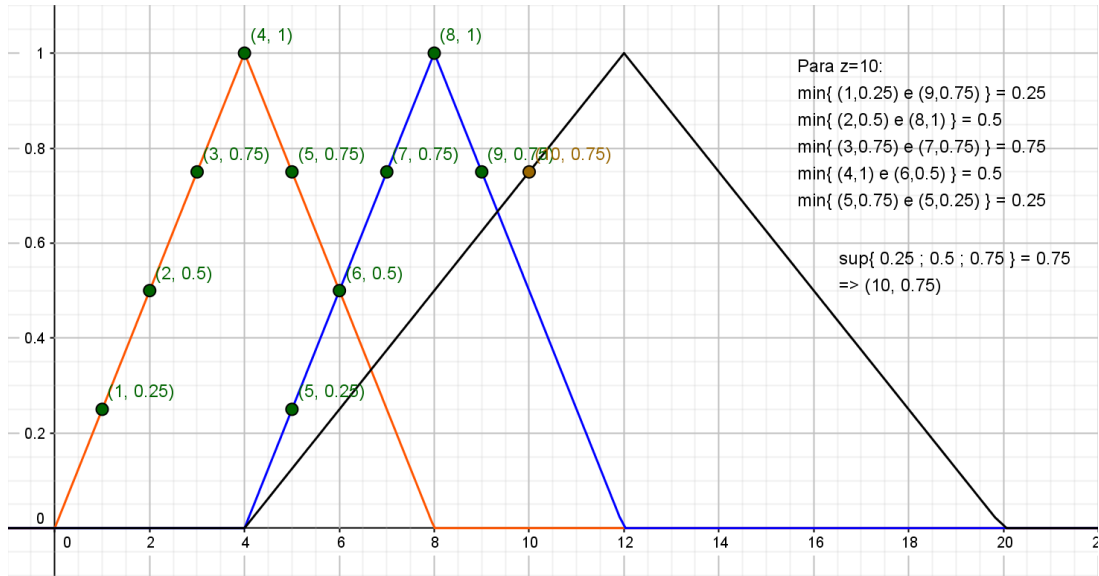
As definições que seguem podem ser vistas como casos particulares do princípio de extensão, tanto para funções de uma como de duas variáveis.

Definição 1.23. Sejam A e B dois números fuzzy e λ um número real.

(a) A soma dos números fuzzy A e B é o número fuzzy, $A + B$, cuja função de pertinência é

$$\varphi_{(A+B)}(z) = \begin{cases} \sup_{\varnothing(z)} \min[\varphi_A(x), \varphi_B(y)] & \text{se } \varnothing(z) \neq \emptyset \\ 0 & \text{se } \varnothing(z) = \emptyset \end{cases},$$

onde $\varnothing(z) = \{(x, y) : x + y = z\}$.



(b) A multiplicação de λ por A é o número fuzzy λA , cuja função de pertinência é

$$\varphi_{\lambda A}(z) = \begin{cases} \sup_{\{x:\lambda x=z\}} [\varphi_A(x)] & \text{se } \lambda \neq 0 \\ \chi_{\{0\}}(z) & \text{se } \lambda = 0 \end{cases} = \begin{cases} \varphi_A(\lambda^{-1}z) & \text{se } \lambda \neq 0 \\ \chi_{\{0\}}(z) & \text{se } \lambda = 0 \end{cases}$$

onde $\chi_{\{0\}}$ é a função característica de $\{0\}$.

(c) A diferença $A - B$ é o número cuja função de pertinência é dada por:

$$\varphi_{(A-B)}(z) = \begin{cases} \sup_{\varnothing(z)} \min(\varphi_A(x), \varphi_B(z)) & \text{se } \varnothing(z) \neq 0 \\ 0 & \text{se } \varnothing(z) = 0 \end{cases}$$

onde $\varnothing(z) = \{(x, y) : x - y = z\}$.

(d) A multiplicação de A por B é o número fuzzy $A \cdot B$, cuja função de pertinência é dada por:

$$\varphi_{(A \cdot B)}(z) = \begin{cases} \sup_{\varnothing(z)} \min(\varphi_A(x), \varphi_B(z)) & \text{se } \varnothing(z) \neq 0 \\ 0 & \text{se } \varnothing(z) = 0 \end{cases}$$

onde $\varnothing(z) = \{(x, y) : x \cdot y = z\}$.

(e) A divisão é o número fuzzy A/B cuja função de pertinência é

$$\varphi_{(A/B)}(z) = \begin{cases} \sup_{\varnothing(z)} \min(\varphi_A(x), \varphi_B(z)) & \text{se } \varnothing(z) \neq 0 \\ 0 & \text{se } \varnothing(z) = 0 \end{cases}$$

onde $\varnothing(z) = \{(x, y) : x/y = z\}$.

O próximo teorema garante que o resultado das operações aritméticas entre números fuzzy é um número fuzzy. Mais ainda, generaliza o Corolário 1.22, relacionando, por meio dos α -níveis, as operações aritméticas para números fuzzy com as respectivas operações aritméticas para intervalos.

Teorema 1.24. *Os α -níveis do conjunto fuzzy $A \otimes B$ são dados por*

$$[A \otimes B]^\alpha = [A]^\alpha \otimes [B]^\alpha$$

para todo $\alpha \in [0, 1]$, sendo \otimes qualquer uma das operações aritméticas mencionadas anteriormente.

Podemos observar que os α -níveis de um número fuzzy é sempre um intervalo de \mathbb{R} , dado por:

$$[A]^\alpha = [a_1^\alpha, a_2^\alpha], \text{ com } a_1^\alpha = \min\{\varphi_A^{-1}(\alpha)\} \text{ e } a_2^\alpha = \max\{\varphi_A^{-1}(\alpha)\},$$

sendo $\varphi_A^{-1}(\alpha) = \{x \in \mathbb{R} : \varphi_A(x) = \alpha\}$ a pré-imagem de α .

Proposição 1.25. *Sejam A e B números fuzzy com α -níveis dados, respectivamente, por $[A]^\alpha = [a_1^\alpha, a_2^\alpha]$ e $[B]^\alpha = [b_1^\alpha, b_2^\alpha]$. Então valem as seguintes propriedades:*

(a) *A soma entre A e B é o número fuzzy $A + B$ cujos α -níveis são*

$$[A + B]^\alpha = [A]^\alpha + [B]^\alpha = [a_1^\alpha + b_1^\alpha, a_2^\alpha + b_2^\alpha].$$

(b) *A diferença entre A e B é o número fuzzy $A - B$ cujos α -níveis são*

$$[A - B]^\alpha = [A]^\alpha - [B]^\alpha = [a_1^\alpha - b_2^\alpha, a_2^\alpha - b_1^\alpha].$$

(c) *A multiplicação de λ por A é o número fuzzy λA cujos α -níveis são*

$$[\lambda A]^\alpha = \lambda [A]^\alpha = \begin{cases} [\lambda a_1^\alpha, \lambda a_2^\alpha] & \text{se } \lambda \geq 0 \\ [\lambda a_2^\alpha, \lambda a_1^\alpha] & \text{se } \lambda < 0 \end{cases}.$$

(d) *A multiplicação de A por B é o número fuzzy $A \cdot B$ cujos α -níveis são*

$$[A \cdot B]^\alpha = [A]^\alpha [B]^\alpha = [\min P, \max P],$$

onde $P = \{a_1^\alpha b_1^\alpha, a_1^\alpha b_2^\alpha, a_2^\alpha b_1^\alpha, a_2^\alpha b_2^\alpha\}$.

(e) *A divisão de A por B , se $0 \notin \text{supp} B$, é o número fuzzy cujos α -níveis são*

$$\left[\frac{A}{B}\right]^\alpha = \frac{[A]^\alpha}{[B]^\alpha} = [a_1^\alpha, a_2^\alpha] \left[\frac{1}{b_2^\alpha}, \frac{1}{b_1^\alpha}\right].$$

Definição 1.26. (Diferença de Hukuhara: $A -_H B$).

Sejam A e B dois números fuzzy. Se existir um número fuzzy C tal que $A = B + C$, então C é chamado de Diferença de Hukuhara de A e B e a denotamos por $A -_H B$.

Em α -níveis, isto equivale a dizer que

$$[A -_H B]^\alpha = [a_1^\alpha - b_1^\alpha, a_2^\alpha - b_2^\alpha] \quad \forall \alpha \in [0, 1].$$

Como

$$[A - B]^\alpha = [a_1^\alpha - b_2^\alpha, a_2^\alpha - b_1^\alpha],$$

segue que

$$A - B = A -_H B \Leftrightarrow b_1^\alpha = b_2^\alpha,$$

ou seja,

$$A - B = A -_H B \Leftrightarrow B \in \mathbb{R}.$$

Note que

$$A - B = A + (-1)B \neq A -_H B$$

em geral.

LÓGICA FUZZY

Neste capítulo trataremos da teoria necessária para o desenvolvimento de um controlador fuzzy, a ser utilizado no capítulo de aplicações para a tomada de decisão da taxa selic. Foi utilizado como referência o livro [7], para outras aplicações temos [11].

2.1 CONECTIVOS BÁSICOS DA LÓGICA CLASSICA

Os conectivos "e", "ou", "não" e "implicação" são os mais utilizados na modelagem matemática em sentenças como:

Exemplo 2.1. "Se a está em A ou b está em B então c não está em C e d está em D"

Os conectivos são dispostos em tabelas verdades onde obtém-se o valor lógico da sentença, considerando como conjuntos A, B, C e D, na sentença acima para que c não pertença à C e d pertença à D, terá que ser verdadeiro uma das proposições: que a pertença à A ou b pertença à B.

Sentenças verdadeiras na lógica clássica tem valor lógico igual a **1**, enquanto as falsas tem valor lógico igual a **0**. Como extensão para a lógica fuzzy, utilizaremos os seguintes símbolos para os conectivos: \wedge (mínimo) para a conjunção **e**, \vee (máximo) para **ou**, \sim para **negação** e \Rightarrow para **implicação**. Sendo p e q proposições, as tabelas verdades para cada conectivo estão nas tabelas abaixo:

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 1: Conectivo **e**

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 2: Conectivo **ou**

p	$\sim p$
0	1
1	0

Tabela 3: Conectivo **negação**

p	q	$p \Rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Tabela 4: Conectivo **implicação**

Verificamos que os valores nas tabelas verdades são apenas o **0** e o **1**, por isso a lógica clássica, as vezes é chamada "lógica a dois valores", temos também as seguintes operações binárias:

- Conectivo \wedge :

$$\wedge : \{0, 1\} \times \{0, 1\} \longrightarrow \{0, 1\}$$

$$(p, q) \mapsto \wedge(p, q) = p \wedge q = \min\{p, q\}.$$

Então,

$$\wedge(0, 0) = 0 \wedge 0 = 0;$$

$$\wedge(0, 1) = 0 \wedge 1 = 0;$$

$$\wedge(1, 0) = 1 \wedge 0 = 0;$$

$$\wedge(1, 1) = 1 \wedge 1 = 1.$$

- Conectivo \vee :

$$\vee : \{0, 1\} \times \{0, 1\} \longrightarrow \{0, 1\}$$

$$(p, q) \mapsto \vee(p, q) = p \vee q = \max\{p, q\}.$$

Então,

$$\vee(0, 0) = 0 \vee 0 = 0;$$

$$\vee(0, 1) = 0 \vee 1 = 1;$$

$$\vee(1, 0) = 1 \vee 0 = 1;$$

$$\vee(1, 1) = 1 \vee 1 = 1.$$

- A negação considerada operação unária:

$$\sim : \{0, 1\} \longrightarrow \{0, 1\}$$

$$p \mapsto \sim p,$$

onde, $\sim 1 = 0$ e $\sim 0 = 1$. Verificamos que $\sim p = 1 - p$

• Implicação:

$$\Rightarrow: \{0, 1\} \times \{0, 1\} \longrightarrow \{0, 1\}$$

$$(p, q) \mapsto \Rightarrow (p, q) = (p \Rightarrow q).$$

Através das operações anteriores conseguimos produzir a tabela verdade da **implicação** usando as fórmulas:

$$(1) (p \Rightarrow q) = (\sim p) \vee q;$$

$$(2) (p \Rightarrow q) = (\sim p) \vee (p \wedge q);$$

$$(3) (p \Rightarrow q) = \max\{x \in \{0, 1\} : p \wedge x \leq q\}.$$

Verificamos que "não p ou q " é a implicação " p implica q ", isto é, $(\sim p) \vee q = (p \Rightarrow q)$ é de fato uma implicação:

$$\Rightarrow (0, 0) = (0 \Rightarrow 0) = (\sim 0) \vee 0 = 1;$$

$$\Rightarrow (0, 1) = (0 \Rightarrow 1) = (\sim 0) \vee 1 = 1;$$

$$\Rightarrow (1, 0) = (1 \Rightarrow 0) = (\sim 1) \vee 0 = 0;$$

$$\Rightarrow (1, 1) = (1 \Rightarrow 1) = (\sim 1) \vee 1 = 1.$$

Usando a sentença 2.1 na página 25, temos que os conjuntos são clássicos, portanto podemos avaliar a sentença utilizando os operadores vistos anteriormente com os valores 0 e 1.

$$\underbrace{\underbrace{\text{"Se } a \text{ está em } A}_{p} \text{ ou } \underbrace{b \text{ está em } B}_{q}}_P \text{ então } \underbrace{\underbrace{c \text{ não está em } C}_{r} \text{ e } \underbrace{d \text{ está em } D}_{s}}_Q$$

Os valores das expressões p , q , r e s podem ser apenas 0 para não pertencer a um conjunto ou 1 para pertencer, exemplo: $p = 1$ se $a \in A$ e $p = 0$ se $a \notin A$, do mesmo modo para q , r e s .

Avaliando a sentença 2.1 nas situações:

$$a \notin A (p = 0); b \in B (q = 1); c \notin C (r = 1) \text{ e } d \notin D (s = 0),$$

então o valor lógico da sentença 2.1 é:

$$(0 \vee 1) \Rightarrow (1 \wedge 0) = (1 \Rightarrow 0) = 0.$$

Em termos de lógica fuzzy o valor lógico da sentença "a está em A", coincide com o obtido com a função característica do conjunto A em a, isto é, o valor de p é dado por $\chi_A(a)$, do mesmo modo, quando "a não está em A", é dada por $1 - \chi_A(a)$.

2.2 CONECTIVOS BÁSICOS DA LÓGICA FUZZY

Avaliamos a sentença 2.1 apenas usando os valores 0 e 1, isto é válido para a lógica clássica; para analisarmos na lógica fuzzy devemos indicar quanto a proposição "a está em A" é verdadeira, sendo A fuzzy, sabendo que a pode pertencer a A com valores no intervalo $[0, 1]$. A avaliação lógica dos conectivos usando fuzzy é feita estendendo por meio de normas e conormas triangulares.

2.2.1 Operações *t-norma* e *t-conorma*

A operação *t-norma* estende o operador \wedge que modela o conectivo "e".

Definição 2.1. (*t-norma*). O operador $\Delta: [0, 1] \times [0, 1] \longrightarrow [0, 1]$, $\Delta(x, y) = x \Delta y$, é uma *t-norma*, se satisfizer as seguintes condições:

- t_1) elemento neutro: $\Delta(1, x) = 1 \Delta x = x$;
- t_2) comutativa: $\Delta(x, y) = x \Delta y = y \Delta x = \Delta(y, x)$;
- t_3) associativa: $x \Delta (y \Delta z) = (x \Delta y) \Delta z$;
- t_4) monotonicidade: se $x \leq u$ e $y \leq v$, então $x \Delta y \leq u \Delta v$.

Exemplo 2.2. Considerando o operador

$$\Delta_1(x, y) = \min\{x, y\} = x \wedge y.$$

Somente resultará o valor 1 se $x = 1$ e $y = 1$, qualquer outro valor será o menor entre x e y (na lógica clássica o 0). Temos abaixo outros exemplos:

$$\begin{aligned} \Delta_2(x, y) &= xy; \\ \Delta_3(x, y) &= \max\{0, x + y - 1\}; \\ \Delta_4(x, y) &= \begin{cases} x & \text{se } y = 1 \\ y & \text{se } x = 1 \\ 0 & \text{caso contrário} \end{cases}. \end{aligned}$$

O operador *t-conorma* estende o operador \vee do conectivo "ou".

Definição 2.2. (*t-conorma*). O operador $\nabla(x, y) = x \nabla y$, é uma *t-conorma*, se satisfizer as seguintes condições:

- c_1) elemento neutro: $\nabla(0, x) = 0 \nabla x = x$;

c_2) comutativa: $\nabla(x, y) = x \nabla y = y \nabla x = \nabla(y, x)$;

c_3) associativa: $x \nabla (y \nabla z) = (x \nabla y) \nabla z$;

c_4) monotonicidade: se $x \leq u$ e $y \leq v$, então $x \nabla y \leq u \nabla v$.

Exemplo 2.3. O operador

$$\nabla_1(x, y) = \max\{x, y\} = x \vee y.$$

é uma *t-conorma* que reproduz a tabela verdade do conectivo \vee . Outros exemplos abaixo:

$$\nabla_2(x, y) = \min\{1, x + y\};$$

$$\nabla_3(x, y) = x + y - xy.$$

A operação abaixo estende a tabela verdade da negação:

Definição 2.3. (negação). Uma aplicação $\eta : [0, 1] \rightarrow [0, 1]$ é uma negação se satisfizer as seguintes condições:

n_1) fronteiras: $\eta(0) = 1$ e $\eta(1) = 0$;

n_2) involução: $\eta(\eta(x)) = x$;

n_2) monotonicidade: η é decrescente.

As aplicações

$$\eta_1(x) = 1 - x \quad e \quad \eta_2 = \frac{1 - x}{1 + x}$$

reproduzem a tabela verdade da negação \sim .

Observamos que as operações $\Delta = \wedge$, $\nabla = \vee$ e $\eta = 1 - x$, satisfazem as leis de Morgan, isto é, para todo par (x, y) de $[0, 1] \times [0, 1]$ valem

$$\eta(x \wedge y) = \eta(x) \vee \eta(y)$$

$$\eta(x \vee y) = \eta(x) \wedge \eta(y)$$

Dizemos que a t-norma Δ e a t-conorma ∇ são **duais** em relação a uma negação η se satisfazem a uma das duas leis de Morgan.

Definição 2.4. (Implicação fuzzy). Um operador $\Rightarrow: [0, 1] \rightarrow [0, 1]$ é uma implicação fuzzy se satisfizer as seguintes condições:

1. reproduzir a tabela da implicação clássica;
2. for decrescente na primeira variável, ou seja, para cada $x \in [0, 1]$ tem-se

$$(a \Rightarrow x) \leq (b \Rightarrow x) \quad \text{se } a \geq b;$$

3. for crescente na segunda variável, ou seja, para cada $x \in [0, 1]$ tem-se

$$(x \Rightarrow a) \leq (x \Rightarrow b) \quad \text{se } a \geq b.$$

Assim, a classe das implicações fuzzy consiste de todas as aplicações do quadrado $[0, 1] \times]0, 1]$ em $[0, 1]$ cuja restrição aos vértices coincide com os valores da implicação clássica; que sejam decrescentes com relação às abscissas e crescentes em relação às ordenadas.

As fórmulas das implicações clássicas **não** produzem as mesmas implicações fuzzy. Temos a seguir outras implicações:

- (4) uma S-implicação tem a forma $(x \Rightarrow y) = \eta(x) \nabla y$;
- (5) uma Q-implicação tem a forma $(x \Rightarrow y) = \eta(x) \nabla (x \Delta y)$;
- (6) uma R-implicação tem a forma $(x \Rightarrow y) = \sup\{z \in [0, 1] : x \nabla z \leq y\}$.

Exemplo 2.4. (Implicações fuzzy).

Além dessas aplicações, para aprofundar nesse assunto temos o livro [4], as seguintes operações são implicações fuzzy:

- a) Implicação de Gödel:

$$(x \Rightarrow y) = g(x, y) = \begin{cases} 1 & \text{se } x \leq y \\ y & \text{se } x > y \end{cases}.$$

- b) Implicação de Goguen:

$$(x \Rightarrow y) = g_n(x, y) = \begin{cases} 1 & \text{se } x \leq y \\ \frac{y}{x} & \text{se } x > y \end{cases}.$$

- c) Implicação de Lukasiewicz:

$$(x \Rightarrow y) = l(x, y) = \min\{(1 - x + y), 1\}.$$

d) Implicação de Kleene-Dienes:

$$(x \Rightarrow y) = k_d(x, y) = \max\{(1 - x), y\}.$$

e) Implicação de Reichenbach:

$$(x \Rightarrow y) = r(x, y) = (1 - x + xy).$$

f) Implicação de Zadeh:

$$(x \Rightarrow y) = z(x, y) = \max\{(1 - x), \min(x, y)\}.$$

g) Implicação de Gaines-Rescher:

$$(x \Rightarrow y) = g_r(x, y) = \begin{cases} 1 & \text{se } x \leq y \\ 0 & \text{se } x > y \end{cases}.$$

h) Implicação de Wu:

$$(x \Rightarrow y) = w(x, y) = \begin{cases} 1 & \text{se } x \leq y \\ \min\{1 - x, y\} & \text{se } x > y \end{cases}.$$

Observe que a implicação de Zadeh é uma Q-implicação

$$(x \Rightarrow y) = (1 - x) \vee (x \wedge y) = \eta(x) \vee (x \wedge y).$$

Exemplo 2.5. Voltemos à expressão 2.1 para obter o valor lógico quando temos $\Delta = \wedge, \nabla = \vee, \eta(x) = 1 - x$ e a implicação de Zadeh.

Consideremos como exemplo o grau de pertinência de cada célula p, q, r e s ao conjunto relacionado como: $\varphi_A(a) = 0,3$; $\varphi_B(b) = 0,7$; $\varphi_C(c) = 0,4$; $\varphi_D(d) = 0,8$. Então, temos: Portanto, o valor lógico de 2.1 é o resultado da aplicação

$$\begin{aligned} p \nabla q &= \max(0,3;0,7) = 0,7; \\ r &= 1 - \varphi_C(c) = 1 - 0,4 = 0,6; \\ r \Delta s &= \min(0,6;0,8) = 0,6. \end{aligned}$$

$$(p \nabla q) \Rightarrow (r \Delta s).$$

Usando a implicação de Zadeh, temos:

$$(p \nabla q) \Rightarrow (r \Delta s) = \max\{1 - (p \nabla q), \min((p \nabla q), (r \Delta s))\} = 0,6.$$

Então, para as pertinências acima, a expressão 2.1 é verdade com grau 0,6.

2.3 RACIOCÍNIO APROXIMADO E VARIÁVEIS LINGUÍSTICAS

O raciocínio aproximado refere-se ao processo onde se pode obter conclusões a partir de premissas incertas. Quando esta incerteza é considerada fuzzy, é frequente o uso do termo raciocínio fuzzy.

Como exemplo temos:

Exemplo 2.6. "Se o aluno estudou, então ele irá bem na prova"

De modo geral temos:

Exemplo 2.7. "Se X é \diamond então Y é \square ."

Portanto, sabendo-se que X é \diamond , concluímos que Y é \square .

Isto é uma generalização do método conhecido como **modus ponens**, a diferença está na subjetividade dos predicados envolvidos. Nas sentenças acima é utilizada a linguagem "natural" sem o formalismo da linguagem matemática.

Na sentença 2.7 não há qualquer conjunto envolvido, apenas qualificações a respeito das variáveis X e Y, para termos uma avaliação lógica de 2.7 representando na forma da sentença 2.1 veremos o conceito de *variável linguística*.

Definição 2.5. (Variável linguística.) Uma *variável linguística* X no universo U é uma variável cujos valores assumidos são subconjuntos fuzzy de U.

Intuitivamente, uma variável linguística é um substantivo, enquanto seus valores são adjetivos, representados por conjuntos fuzzy. Por exemplo, "gripe" é uma variável linguística que pode assumir os atributos "forte" ou "fraca".

Sentenças em que aparecem variáveis linguísticas junto com seus atributos são geralmente chamadas de proposições fuzzy, o que interessará serão as variáveis cujos valores são **números fuzzy** onde o universo de discurso é o conjunto dos números reais, dizemos que o suporte da variável é o conjunto dos números reais e X é uma variável linguística real.

Esclarecemos que X é A (ou X está em A) significa " $X = x$ é A " (ou " $X = x$ está em A "). O valor lógico de " $X = x$ é A " é o número $\varphi_A(x)$ que indica quanto $X = x$ está em concordância com o termo linguístico modelado pelo conjunto fuzzy A , para facilitar a notação, usaremos " x é A " (ou " x está em A "), no lugar de " $X=x$ é A " (ou " $X=x$ está em A ").

2.4 MODUS PONENS E MODUS PONENS GENERALIZADO

Modelemos matematicamente o **modus ponens fuzzy**:

$p \Rightarrow q$: "Se x é A então y é B "

Fato: " x é A "

Conclusão: " y é B "

Note que $(p \Rightarrow q)$ é uma proposição fuzzy condicional que é modelada por uma relação fuzzy \mathcal{R} de $U \times V$, cuja função de pertinência é

$$\varphi_{\mathcal{R}}(x, y) = [\varphi_A(x) \Rightarrow \varphi_B(y)],$$

onde x e y são valores de variáveis linguísticas de U e V , respectivamente.

Assim, o valor da sentença "Se x é A então y é B " depende da implicação a ser escolhida.

A implicação clássica, isto é, $\varphi_A(x) \in \{0, 1\}$ e $\varphi_B(y) \in \{0, 1\}$, produz a relação fuzzy cuja função de pertinência é dada por:

$$\begin{aligned} \varphi_{\mathcal{R}}(x, y) &= \chi_{\mathcal{R}}(x, y) = (\chi_A(x) \Rightarrow \chi_B(y)) = \\ &= \begin{cases} 1 & \text{se } (x \notin A \text{ e } y \text{ qualquer}) \text{ ou } (x \in A \text{ e } y \in B) \\ 0 & \text{se } x \in A \text{ e } y \notin B \end{cases} \end{aligned}$$

De modo que

$$\sup_{x \in U} [\chi_{\mathcal{R}}(x, y) \wedge \chi_A(x)] = \begin{cases} 1 & \text{se } y \in B \\ 0 & \text{se } y \notin B \end{cases} = \chi_B(y),$$

ou seja, para o caso clássico, o modus ponens pode ser escrito matematicamente por meio da fórmula:

$$\chi_B(y) = \sup_{x \in U} [\chi_{\mathcal{R}}(x, y) \wedge \chi_A(x)].$$

O modus ponens clássico pode ser dado pela regra de composição de inferência $B = A \circ \mathcal{R}$, em que a relação \mathcal{R} é obtida por meio de uma implicação fuzzy que modela a sentença condicional

"Se $x \in A$ então $y \in B$ ".

Com o propósito de inferir conjuntos fuzzy, esta fórmula será estendida para situações fuzzy mais gerais tais como o **modus ponens fuzzy** e o **modus ponens fuzzy generalizado**.

O modus ponens fuzzy modela o seguinte silogismo:

Regra: "Se a aluno estudou, então ele irá bem na prova"

Fato: "O aluno estudou"

Conclusão: "O aluno irá bem na prova"

A lógica fuzzy revela seu grande potencial na modelagem de cada uma das sentenças acima. Os substantivos e seus atributos são modelados por conjuntos fuzzy (por funções de pertinência) enquanto que os conectivos por operadores como t-normas e t-conormas, implicações e/ou negações. A conclusão, que deve ser um conjunto fuzzy, é obtida pela extensão da regra de composição de inferência

$$\chi_B(y) = \sup_{x \in U} [\chi_{\mathcal{R}}(x, y) \wedge \chi_A(x)], \quad (2.1)$$

substituindo as funções características por funções de pertinência e o operador \wedge por alguma t-norma, ou seja,

$$\varphi_B(y) = \sup_{x \in U} [\varphi_{\mathcal{R}}(x, y) \triangle \varphi_A(x)]. \quad (2.2)$$

Em resumo, a fórmula 2.2 é a regra de inferência que modela o **modus ponens fuzzy**

O lado direito da Equação 2.2 exige que a t-norma e a implicação fuzzy lá adotadas sejam escolhidas de modo que a saída coincida com $\varphi_B(y)$ para todo y . Esse é um problema de equação relacional que pode trazer dificuldades, para facilitar estenderemos a Equação 2.2 admitindo uma entrada A^* no lugar de A e a flexibilizaremos deixando de exigir que a saída B^* seja B , para a entrada $A = A^*$. Neste caso, denominaremos o silogismo por **modus ponens generalizado**, que tem a forma geral:

Regra: "Se x é A , então y é B "

Fato: " x é A^* "

Conclusão: " y é B^* "

A saída do **modus ponens generalizado** é o conjunto fuzzy B^* , cuja função de pertinência é

$$\chi_{B^*}(y) = \sup_{x \in U} [\varphi_{\mathcal{R}}(x, y) \Delta \varphi_{A^*}(x)], \quad (2.3)$$

que, por analogia à regra de composição de inferência, tem a forma

$$\mathcal{R}(A^*) = A^* \otimes^t \mathcal{R} = B^*,$$

onde \otimes^t é uma operação análoga à " \circ ", trocando-se a t -norma do mínimo por Δ .

Observação: no caso clássico sempre $\mathcal{R}(A) = B$, porém, no caso fuzzy, dependendo da t -norma e da implicação, nem sempre teremos $\mathcal{R}(A) = B$.

É bastante comum se obter um funcional teórico, sem que esse funcional reproduza os dados que o originaram. Esse é o caso do famoso método dos mínimos quadrados, cuja principal propriedade é obter o funcional com menor erro quadrático possível para o conjunto de dados. Métodos para obter funcionais que reproduzam exatamente os dados que o originaram costumam ser chamados interpolação.

Exemplo 2.8. Suponha a t -norma do produto, $x \Delta y = xy$, e a implicação

$$(x \Rightarrow y) = \begin{cases} 1 & \text{se } x = 0 \\ 1 \wedge \frac{y}{x} & \text{se } x \neq 0 \end{cases}.$$

Dados os números fuzzy A e B , temos

$$(\varphi_A(x) \Rightarrow \varphi_B(y)) \cdot \varphi_A(x) = \varphi_A(x) \wedge \varphi_B(y).$$

Assim,

$$\begin{aligned} \sup_{x \in U} [\varphi_{\mathcal{R}}(x, y) \Delta \varphi_{A^*}(x)] &= \sup_{x \in U} [\varphi_A(x) \Rightarrow \varphi_B(y)] \cdot \varphi_A(x) \\ &= \sup_{x \in U} [\varphi_A(x) \wedge \varphi_B(y)] = \varphi_B(y). \end{aligned}$$

ou seja, $\mathcal{R}(A) = B$.

Exemplo 2.9. Consideramos os subconjuntos fuzzy:

$$A = 0,4/x_1 + 1,0/x_2 + 0,6/x_3 \quad e \quad B = 0,8/y_1 + 0,4/y_2,$$

a implicação de Lukasiewicz:

$$(x \Rightarrow y) = \min\{(1 - x + y), 1\}$$

e a t-norma $\Delta_1 = \wedge$.

temos

$$\mathcal{R} = 1,0/(x_1, y_1) + 1,0/(x_1, y_2) + 0,8/(x_2, y_1) + 0,4/(x_2, y_2) + 1,0/(x_3, y_1) + 0,8/(x_3, y_2).$$

Desta forma, para a entrada

$$A = 0,4/x_1 + 1,0/x_2 + 0,6/x_3,$$

temos a saída \tilde{B} com função de pertinência

$$\begin{aligned} \varphi_{\tilde{B}}(y_1) &= \max_{x_i} [\varphi_{\mathcal{R}}(x_i, y_1) \wedge \varphi_A(x_i)] = \\ &= \max[\min(1, 0; 0, 4); \min(0, 8; 1, 0); \min(1, 0; 0, 6)] = 0, 8; \end{aligned}$$

$$\begin{aligned} \varphi_{\tilde{B}}(y_2) &= \max_{x_i} [\varphi_{\mathcal{R}}(x_i, y_2) \wedge \varphi_A(x_i)] = \\ &= \max[\min(1, 0; 0, 4); \min(0, 4; 1, 0); \min(0, 8; 0, 6)] = 0, 6. \end{aligned}$$

e, para a entrada

$$A^* = 0,6/x_1 + 0,9/x_2 + 0,7/x_3,$$

temos a saída B^* com função de pertinência

$$\begin{aligned} \varphi_{\tilde{B}}(y_1) &= \max_{x_i} [\varphi_{\mathcal{R}}(x_i, y_1) \wedge \varphi_{A^*}(x_i)] = \\ &= \max[\min(1, 0; 0, 6); \min(0, 8; 0, 9); \min(1, 0; 0, 7)] = 0, 8; \end{aligned}$$

$$\begin{aligned} \varphi_{\tilde{B}}(y_2) &= \max_{x_i} [\varphi_{\mathcal{R}}(x_i, y_2) \wedge \varphi_{A^*}(x_i)] = \\ &= \max[\min(1, 0; 0, 6); \min(0, 4; 0, 9); \min(0, 8; 0, 7)] = 0, 7. \end{aligned}$$

Desta forma, para este exemplo, as saídas obtidas pelo modus ponens fuzzy generalizado são

$$\mathcal{R}(A) = \tilde{B} = 0,8/y_1 + 0,6/y_2 \quad e \quad \mathcal{R}(A^*) = \tilde{B} = 0,8/y_1 + 0,7/y_2.$$

É interessante notar que $B \subset \tilde{B}$. Este fato é de interesse em raciocínio aproximado, pois indica que B poderia ser uma saída ótima em algum sentido.

Método Prático

A partir da observação feita após a fórmula 2.3, temos

$$B^* = A^* \otimes^t \mathcal{R},$$

em que, para o caso de domínios finitos, A^* e B^* são escritos em linha e

$$[\mathcal{R}] = [\varphi_{\mathcal{R}}(x_i, y_j)]_{ixj}$$

onde " \otimes^t " é a composição sup-t. No exemplo anterior temos

$$\mathcal{R} = [\varphi_{\mathcal{R}}(x_i, y_j)]_{3 \times 2} = \begin{bmatrix} 1,0 & 1,0 \\ 0,8 & 0,4 \\ 1,0 & 0,8 \end{bmatrix}_{3 \times 2},$$

$$A = \begin{bmatrix} 0,4 & 1,0 & 0,6 \end{bmatrix}$$

e a composição é a max-min, uma vez que $\Delta = \wedge$. Logo,

$$\tilde{B} = A \circ \mathcal{R} = \begin{bmatrix} 0,4 & 1,0 & 0,6 \end{bmatrix} \circ \begin{bmatrix} 1,0 & 1,0 \\ 0,8 & 0,4 \\ 1,0 & 0,8 \end{bmatrix} = \begin{bmatrix} 0,8 & 0,6 \end{bmatrix},$$

ou seja,

$$\tilde{B} = \frac{0,8}{y_1} + \frac{0,6}{y_2}.$$

2.5 CONTROLE FUZZY

Ao longo de sua vida, o ser humano recebe ordens que nem sempre são precisas, então deve tomar uma decisão. Podemos considerar isso um processo em que a execução das tarefas devem seguir uma sequência de "ordens" linguísticas, traduzidas por um

conjunto de regras, capazes de serem decodificadas pelo controlador. Um **Controlador Fuzzy** pode ser considerado como uma maneira de reproduzir um controlador humano, sendo um caso típico de um **Sistema Baseado em Regras Fuzzy (SBRF)**, isto é, um sistema que se utiliza da lógica fuzzy para produzir saídas para cada entrada fuzzy.

Nas aplicações dos **SBRF** cada saída representa a "ação" correspondente à "condição" ou entrada do SBRF. Quando as entradas e saídas têm esta conotação os SBRF são denominados *Controladores Fuzzy*.

Utilizando os métodos que vimos anteriormente podemos traduzir termos linguísticos, utilizados por especialistas para controlar suas tarefas, no formato matemático, de forma à automatizar essas tarefas.

Em controladores fuzzy as tarefas são comandadas por meio de termos da linguagem usual, relacionando com uma variável de interesse. Estes termos, traduzidos por conjuntos fuzzy, são utilizados para transcrever a base de conhecimentos através de uma coleção de regras fuzzy, denominada **base de regras fuzzy**. A partir dessa base de regras obtém-se a relação fuzzy, a qual produzirá a saída (resposta, ação) para cada entrada (estado, condição).

2.6 BASE DE REGRAS FUZZY

Uma base de regras fuzzy tem o seguinte formato

R_1 :	"Proposição fuzzy 1"
ou	R_2 :
	"Proposição fuzzy 2"

ou	R_r :
	"Proposição fuzzy r"

Tabela 5: Forma Geral de uma base de regras fuzzy.

Nos sistemas baseados em regras fuzzy cada proposição tem a forma

Se "estado" Então "resposta"

Tanto "estado" como "resposta" são valores assumidos por variáveis linguísticas que são modelados por conjuntos fuzzy. Os conjuntos fuzzy que compõem o "estado" são chamados *antecedentes* e os que compõem a "resposta" são chamados *consequentes*, tendo como particularidade cada regra a forma

Se "condição" Então "ação"

Cada uma das classificações das variáveis que constam na base de regras é modelada por um conjunto fuzzy. A lógica fuzzy é a outra parte utilizada na obtenção da relação fuzzy que sintetiza as informações matemáticas contidas na base de regras.

A base de regras cumpre o papel de "traduzir" matematicamente as informações que formam a base de conhecimentos do sistema fuzzy. Num certo sentido, pode-se afirmar que quanto mais precisas forem tais informações, menos fuzzy (mais crisp) será a relação fuzzy que representa a base de conhecimentos. Numa situação ideal, tal relação pode mesmo ser uma função no sentido clássico.

2.7 CONTROLADOR FUZZY

Em um sistema fuzzy a cada entrada corresponde uma saída fuzzy, um controlador fuzzy tem essa característica também, mas se a entrada for crisp espera-se uma saída crisp, neste caso um sistema fuzzy é uma função de \mathbb{R}^n em \mathbb{R}^m construída de alguma maneira específica. A seguir veremos módulos que indicam um roteiro para construção desta função.

Módulo de fuzzificação

Neste estágio as entradas do sistema são modeladas por conjuntos fuzzy com seus domínios respectivos. É muito importante a atuação de especialistas na modelagem das funções de pertinência. Mesmo que os conjuntos sejam crisp, eles são transformados em conjuntos fuzzy por meio de sua função característica.

Módulo de base de regras

Este módulo é composto por proposições fuzzy que são descritas na forma linguística

Se x_1 é A_1 e x_2 é A_2 e ... e x_n é A_n

Então u_1 é B_1 e u_2 é B_2 e ... e u_m é B_m

de acordo com as informações de um especialista. As variáveis e suas classificações linguísticas são catalogadas e, em seguida, modeladas por conjuntos fuzzy, isto é, funções de pertinência.

Existem vários métodos para obter as funções de pertinência, tais como: apelos intuitivos, ajustes de curvas, interpolações e até mesmo redes neurais.

Módulo de inferência fuzzy

Neste módulo as proposições são "traduzidas" matematicamente por meio das técnicas da lógica fuzzy, definindo-se quais t-normas, t-conormas e regras de inferência (podendo ser implicações fuzzy) serão utilizadas para se obter a relação fuzzy que modela a base de regras.

O sucesso do controlador depende muito desse módulo, pois ele fornecerá a saída fuzzy a partir de cada entrada fuzzy.

Módulo de defuzzificação

A defuzzificação é um processo que permite representar um conjunto fuzzy por um valor crisp (número real). Na figura 15 vemos um esquema geral de um controlador fuzzy.

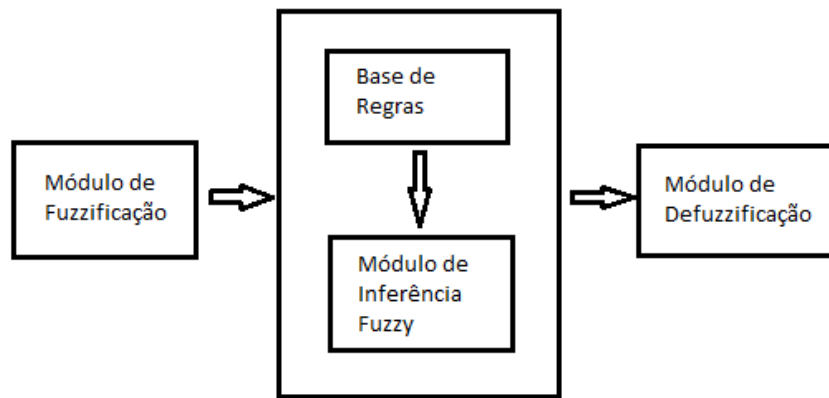


Figura 15: Esquema geral de um controlador fuzzy. [7]

A base de regras é modelada matematicamente por uma relação fuzzy \mathcal{R} , a partir dos conjuntos fuzzy que a compõem e da lógica fuzzy adotada.

A função de pertinência de \mathcal{R} é dada por

$$\varphi_{\mathcal{R}}(x, u) = \nabla(\varphi_{\mathcal{R}_i}(x, u)), \text{ com } 1 \leq i \leq r, \quad (2.4)$$

onde ∇ é uma t-conorma e \mathcal{R}_i é uma relação fuzzy obtida da regra i , cuja função de pertinência $\varphi_{\mathcal{R}_i}$ é obtida de alguma forma - por exemplo, por meio de um *modus ponens* generalizado. Os valores x e u representam o estado e o controle, respectivamente.

A inferência, que representa o controle B para um estado A , é dada por uma regra de composição de inferência: $B = R(A)$ cuja função de pertinência é dada por

$$\varphi_B(u) = \sup_x (\varphi_{\mathcal{R}}(x, u) \triangle \varphi_A(x)), \quad (2.5)$$

onde \triangle é uma t-norma.

Conforme a tabela 6 podemos ilustrar a obtenção da relação \mathcal{R} de modo análogo a um modus ponens generalizado.

	$R_1:$	"Se x_1 é A_{11} e ... e x_n é A_{1n} então u_1 é B_{11} e ... e u_m é B_{1m} "
ou		
	$R_2:$	"Se x_1 é A_{21} e ... e x_n é A_{2n} então u_1 é B_{21} e ... e u_m é B_{2m} "
ou		
	.	.
	.	.
	.	.
ou		
	$R_r:$	"Se x_1 é A_{r1} e ... e x_n é A_{rn} então u_1 é B_{r1} e ... e u_m é B_{rm} "
Fato:	$A = x_1$ é A_1 e x_2 é A_2 e ... e x_n é A_n .	
Conclusão:	u é $B = \mathcal{R}(A)$.	

Tabela 6: Ilustração da obtenção da relação \mathcal{R} .

2.8 O MÉTODO DE INFERÊNCIA DE MAMDANI

Mamdani propõe uma relação fuzzy binária \mathcal{M} entre x e u para modelar matematicamente a base de regras.

O método de Mamdani é baseado na regra de composição de inferência max-min conforme o procedimento:

- Em cada regra R_j , da base de regras fuzzy, a condicional "se x é A_j então u é B_j " é modelada pela aplicação \wedge (mínimo) que, erroneamente, costuma ser denominada por implicação de Mamdani (\wedge não é uma implicação fuzzy pois não preserva a tabela de uma implicação clássica);

- Adota-se a t-norma \wedge (mínimo) para o conectivo lógico "e";

- Para o conectivo lógico "ou" adota-se a t-conorma \vee (máximo) que conecta as regras fuzzy das bases de regras.

Formalmente, a relação fuzzy \mathcal{M} é o subconjunto fuzzy de $X \times U$ cuja função de pertinência é dada por

$$\varphi_{\mathcal{M}}(x, u) = \max_{1 \leq j \leq r} (\varphi_{R_j}(x, u)) = \max_{1 \leq j \leq r} [\varphi_{A_j}(x) \wedge \varphi_{B_j}(u)], \quad (2.6)$$

onde r é o número de regras que compõem a base de regras e, A_j e B_j são os subconjuntos fuzzy da regra j . Cada um dos valores $\varphi_{A_j}(x)$ e $\varphi_{B_j}(u)$ são interpretados como os graus com que x e u estão nos subconjuntos A_j e B_j , respectivamente, de maneira que \mathcal{M} nada mais é que a união dos produtos cartesianos fuzzy entre os antecedentes e os consequentes de cada regra.

Observação: Os subconjuntos fuzzy A_j e B_j que aparecem na fórmula (2.6), podem representar o produto cartesiano fuzzy de subconjuntos fuzzy A_{ji} e B_{jk} . Por exemplo, pode ocorrer

$$\varphi_{A_j}(x) = \varphi_{A_{j1}}(x_1) \wedge \varphi_{A_{j2}}(x_2) \text{ e } \varphi_{B_j}(u) = \varphi_{B_{j1}}(u_1) \wedge \varphi_{B_{j2}}(u_2),$$

Exemplo 2.10. Considere um controlador fuzzy com duas entradas e uma saída, cuja base de regras é dada pela tabela abaixo

$R_1: \text{ "Se } x_1 \text{ é } A_{11} \text{ e } x_2 \text{ é } A_{12} \text{ então } u \text{ é } B_1 \text{ "}$ ou $R_2: \text{ "Se } x_1 \text{ é } A_{21} \text{ e } x_2 \text{ é } A_{22} \text{ então } u \text{ é } B_2 \text{ "}$
--

Tabela 7: Base de regras controlador fuzzy com duas entradas e uma saída.

Assim, para cada terna $t = (x_1, x_2, u)$ temos

$$\begin{aligned} \varphi_{\mathcal{M}}(t) &= \{\varphi_{A_{11}}(x_1) \wedge \varphi_{A_{12}}(x_2) \wedge \varphi_{B_1}(u)\} \vee \{\varphi_{A_{21}}(x_1) \wedge \varphi_{A_{22}}(x_2) \wedge \varphi_{B_2}(u)\} \\ &= \max\{\varphi_{A_{11}}(x_1) \wedge \varphi_{A_{12}}(x_2) \wedge \varphi_{B_1}(u), \varphi_{A_{21}}(x_1) \wedge \varphi_{A_{22}}(x_2) \wedge \varphi_{B_2}(u)\} \end{aligned}$$

representando a relação fuzzy obtida da base de regras pelo método de Mamdani.

Agora, para um dado conjunto fuzzy de entrada $A = A_1 \times A_2$, com A_1 e A_2 dois números fuzzy, o conjunto de saída, que representa o controle a ser adotado para A pelo método de Mamdani, é dado por $B = \mathcal{M} \circ A$, cuja função de pertinência é

$$\varphi_B(u) = (\varphi_{\mathcal{M} \circ A})(u) = \sup_x \{\varphi_{\mathcal{M}}(x, u) \wedge \varphi_A(x)\}.$$

Como $A = A_1 \times A_2$, então $\varphi_A(x_1, x_2) = \varphi_{A_1}(x_1) \wedge \varphi_{A_2}(x_2)$.

Desta forma

$$\begin{aligned} \varphi_B(u) &= \sup_x \{ \varphi_{\mathcal{M}}(x, u) \wedge \varphi_A(x) \} \\ &= \sup_{(x_1, x_2)} \{ \varphi_{\mathcal{M}}(x_1, x_2, u) \wedge [\varphi_{A_1}(x_1) \wedge \varphi_{A_2}(x_2)] \} \\ &= \sup_{(x_1, x_2)} \{ [\varphi_{A_{11}}(x_1) \wedge \varphi_{A_{12}}(x_2) \wedge \varphi_{B_1}(u)] \vee (\varphi_{A_{21}}(x_1) \wedge \varphi_{A_{22}}(x_2) \wedge \varphi_{B_2}(u)) \} \wedge [\varphi_{A_1}(x_1) \wedge \varphi_{A_2}(x_2)] \} \\ &= \sup_{(x_1, x_2)} \{ [\varphi_{A_1}(x_1) \wedge \varphi_{A_{11}}(x_1)] \wedge [\varphi_{A_2}(x_2) \wedge \varphi_{A_{12}}(x_2)] \wedge \varphi_{B_1}(u) \} \vee \sup_{(x_1, x_2)} \{ [\varphi_{A_1}(x_1) \wedge \varphi_{A_{21}}(x_1)] \wedge [\varphi_{A_2}(x_2) \wedge \varphi_{A_{22}}(x_2)] \wedge \varphi_{B_2}(u) \} \\ &= \varphi_{B_{R_1}}(u) \vee \varphi_{B_{R_2}}(u). \end{aligned}$$

onde B_{R_1} e B_{R_2} são as saídas parciais devido às regras R_1 e R_2 , respectivamente.

Da fórmula acima, podemos observar que a saída do método de Mamdani resulta da união entre as saídas parciais de cada regra. Para obter cada saída parcial, procede-se da seguinte forma: faz-se a intersecção das entradas com cada antecedente da regra e em seguida, faz-se o produto cartesiano (universos distintos) dessas interseções com os consequentes da regra. A projeção desse produto cartesiano no espaço U é a saída parcial para o conjunto fuzzy de entrada A. Vemos na figura 16.

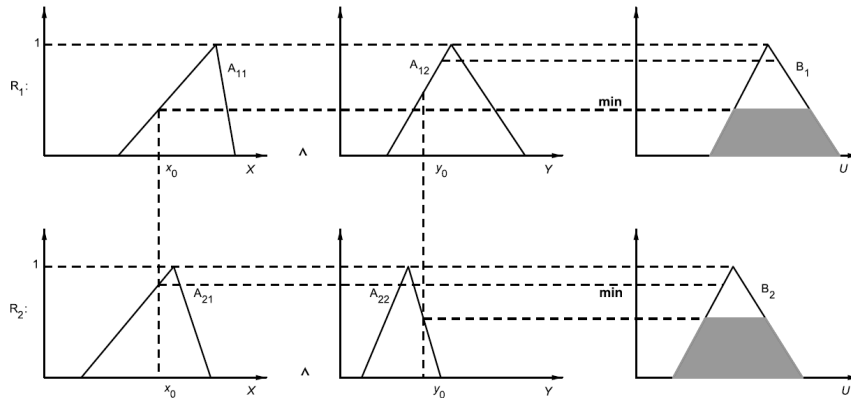


Figura 16: Saídas parciais no método de Mamdani. [7]

e a saída geral é dada pela união das saídas parciais, conforme a figura 17

Notemos que a figura 17 representa a função de pertinência φ_B do controle B que foi obtido pelo conectivo \vee , que é a t-conorma do máximo.

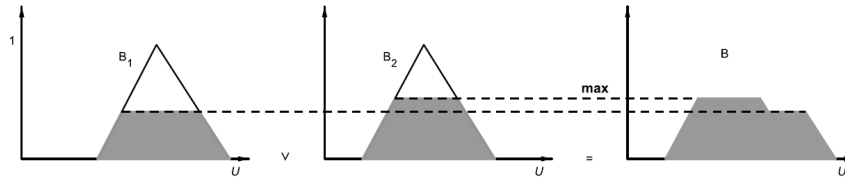


Figura 17: Saída final do controlador fuzzy de Mamdani. [7]

2.9 O MÉTODO DE INFERÊNCIA DE TAKAGI-SUGENO-KANG

As diferenças deste método(TSK) para o de Mamdani estão na forma de escrever o consequente de cada regra e no procedimento de defuzzificação para se obter a saída geral do sistema. O consequente de cada regra é dado explicitamente por uma função dos valores de entrada desta regra.

Este método esta ilustrado na tabela 8, tendo uma base com r regras fuzzy, onde cada uma delas tem n entradas $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, e uma saída $u \in \mathbb{R}$, sendo A_{ij} subconjuntos fuzzy de \mathbb{R} .

R_1 : Se x_1 é A_{11} e x_2 é A_{12} e ... e x_n é A_{1n} então $u_1 = g_1(x_1, x_2, \dots, x_n)$ ou R_2 : Se x_1 é A_{21} e x_2 é A_{22} e ... e x_n é A_{2n} então $u_2 = g_2(x_1, x_2, \dots, x_n)$ ou . . . ou R_r : Se x_1 é A_{r1} e x_2 é A_{r2} e ... e x_n é A_{rn} então $u_r = g_r(x_1, x_2, \dots, x_n)$
--

Tabela 8: Base de regras do método TSK.

A saída geral do método é dada por

$$\begin{aligned}
 u &= f_r(x_1, x_2, \dots, x_n) \\
 &= \frac{\sum_{j=1}^r \omega_j \cdot g_j(x_1, x_2, \dots, x_n)}{\sum_{j=1}^r \omega_j} = \frac{\sum_{j=1}^r \omega_j \cdot u_j}{\sum_{j=1}^r \omega_j}, \tag{2.7}
 \end{aligned}$$

onde os pesos ω_j são dados por $\omega_j = \varphi_{A_{j1}}(x_1) \Delta \varphi_{A_{j2}}(x_2) \dots \Delta \varphi_{A_{jn}}(x_n)$, e Δ é uma t-norma. O peso ω_j corresponde à contribuição da regra R_j para a saída geral. Os casos mais comuns de t-normas são o produto e o mínimo.

Para o caso de duas regras, cada uma com duas variáveis de entrada e uma saída, o método TSK é ilustrado no quadro a seguir.

$R_1: \text{ "Se } x_1 \text{ é } A_{11} \text{ e } x_2 \text{ é } A_{12} \text{ então } u \text{ é } u_1 = g_1(x_1, x_2)\text{"}$ ou $R_2: \text{ "Se } x_1 \text{ é } A_{21} \text{ e } x_2 \text{ é } A_{22} \text{ então } u \text{ é } u_2 = g_2(x_1, x_2)\text{"}$
--

Tabela 9: Base de duas regras do método TSK.

Supondo que Δ seja a t-norma mínimo, temos como saída geral, representando o controle para as ações x_1 e x_2 , o valor de u dado pela equação:

$$u = \frac{\omega_1 u_1 + \omega_2 u_2}{\omega_1 + \omega_2} = \frac{\omega_1 g_1(x_1, x_2) + \omega_2 g_2(x_1, x_2)}{\omega_1 + \omega_2} = f_r(x_1, x_2), \quad (2.8)$$

onde $\omega_i = \min[\varphi_{A_{i1}}(x_1), \varphi_{A_{i2}}(x_2)]$ corresponde ao peso da regra R_i na saída geral do processo.

O caso mais frequente é o que os consequentes de cada regra são funções lineares afins, tendo as funções g_i a forma

$$g_i(x_1, x_2) = a_i x_1 + b_i x_2 + c_i.$$

sendo conhecido como método Takagi-Sugeno (TS).

Exemplo 2.11. Considere um controlador fuzzy com duas entradas e uma saída, onde os conjuntos fuzzy envolvidos, A_{ij} , são números fuzzy triangulares e as saídas de cada regra são dadas por funções g_i , lineares afins. Para cada par de entrada x_0 e y_0 , a figura [e uma representação gráfica para obtenção da saída, a qual representa o controle a ser adotado para tais entradas. Para este exemplo temos a base de regras dadas na tabela 10

$R_1: \text{ "Se } x \text{ é } A_{11} \text{ e } y \text{ é } A_{12} \text{ então } u \text{ é } u_1 = g_1(x, y) = a_1 x_i + b_1 x_2 + c_1\text{"}$ ou $R_2: \text{ "Se } x \text{ é } A_{21} \text{ e } y \text{ é } A_{22} \text{ então } u \text{ é } u_2 = g_2(x, y) = a_2 x_i + b_2 x_2 + c_2\text{"}$
--

Tabela 10: Base de regras para o exemplo.

Neste caso o controle fuzzy, cuja obtenção gráfica está ilustrada na figura 18, é dado por

$$u = \frac{\omega_1 u_1 + \omega_2 u_2}{\omega_1 + \omega_2} = \frac{\omega_1 g_1(x_0, y_0) + \omega_2 g_2(x_0, y_0)}{\omega_1 + \omega_2}. \quad (2.9)$$

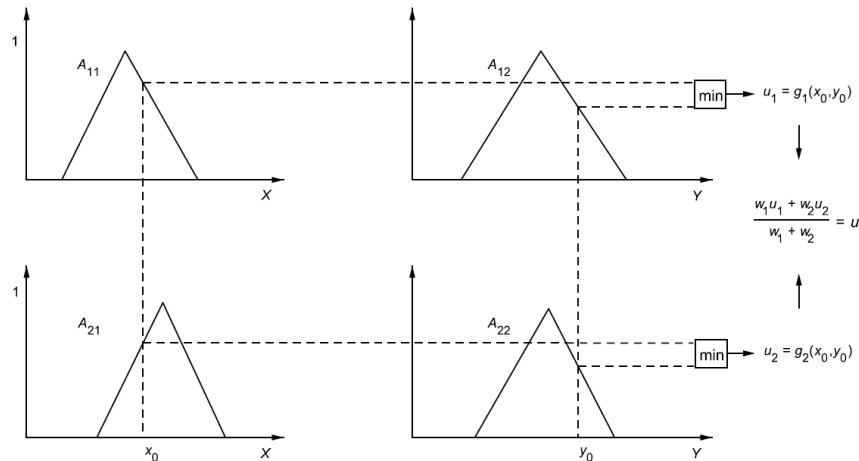


Figura 18: Saída do controlador fuzzy TSK para o exemplo [7]

Os controladores fuzzy têm a propriedade de modelar fenômenos (processos) por meio de um funcional com informações incertas dadas por variáveis linguísticas.

Supondo que certo fenômeno estudado possa ser modelado por um funcional, ao se admitir que a saída de cada regra é dada por uma função linear afim, podemos pensar que o funcional pode ser aproximado localmente por uma reta. Por exemplo, se soubermos a priori que o funcional tem alguma propriedade de suavidade, podemos eleger retas tangentes ao funcional para serem as funções consequentes de cada regra.

É interessante observar que se as funções \$g_i\$ dos consequentes forem constantes, então os dois métodos, de Mamdani e Takagi-Sugeno, produzem as mesmas saídas, se a defuzzificação usada no método de Mamdani for o centro de gravidade. É importante ressaltar que os controladores fuzzy têm propriedades matemáticas importantes como, por exemplo, a capacidade de aproximar funções contínuas. Observamos que os controladores TSK podem ser bem mais gerais, pois as funções de cada consequente podem ter argumentos fuzzy, e os valores das entradas podem ser subconjuntos fuzzy.

2.10 MÉTODOS DE DEFUZZIFICAÇÃO

Em um controlador fuzzy para cada entrada fuzzy o módulo de inferência produz uma saída fuzzy, mesmo que a entrada seja crisp, assim é necessário indicar um método para defuzzificar a saída e obtermos um número real que servirá para o controle, apresentaremos a seguir, esses métodos mais utilizados.

2.10.1 Centro de gravidade, Centroide ou Centro de Área

Este método é semelhante à média aritmética para uma distribuição de frequências de uma dada variável, com a diferença que os pesos aqui são os valores $\varphi_B(u_i)$, que indicam o grau de compatibilidade do valor u_i com o conceito modelado pelo conjunto fuzzy B.

O centro de gravidade dá a média das áreas de todas as figuras que representam os graus de pertinência de um subconjunto fuzzy. As equações 2.10 e 2.11 referem-se ao domínio discreto e domínio contínuo, respectivamente.

$$G(B) = \frac{\sum_{i=0}^n u_i \varphi_B(u_i)}{\sum_{i=0}^n \varphi_B(u_i)} \quad (2.10)$$

$$G(B) = \frac{\int_{\mathbb{R}} u \varphi_B(u) du}{\int_{\mathbb{R}} \varphi_B(u) du} \quad (2.11)$$

Exemplo 2.12. Vamos usar o método de defuzzificação centro de gravidade para determinar qual a sensação de calor de uma pessoa onde temos as saídas do sistema fuzzy dos conjuntos fuzzy quente e frio com suas pertinências, conforme a figura 19. Iremos seguir os passos:

- Determinar o x centroide para cada saída ativada na inferência;
- Calcular a área entre o grau de pertinência e o eixo x para cada saída ativada;
- Calcular a média ponderada dos pontos centroides pelas respectivas áreas.

O cálculo para a obtenção do centroide ?? é descrito a seguir:

Saída Frio:

- no eixo x o centroide é 15;
- para calcular a área do trapézio é necessário encontrar a base menor. Então para o grau de pertinência 0,8 encontram-se os pontos (12;0,8) e (18;0,8) nas respectivas

funções de pertinência. Logo a base menor tem tamanho igual a 6 e a base maior é igual a 30. Dessa forma a área resulta em:

$$A_{\text{frio}} = \frac{0,8(6 + 30)}{2} = 14,4$$

Saída Quente:

- no eixo x o centroide é 35;
- para calcular a área do trapézio é necessário encontrar a base menor. Então para o grau de pertinência 0,2 encontram-se os pontos (23;0,2) e (47;0,2) nas respectivas funções de pertinência. Logo a base menor tem tamanho igual a 24 e a base maior é igual a 30. Dessa forma a área resulta em:

$$A_{\text{quente}} = \frac{0,2(24 + 30)}{2} = 5,4$$

$$\text{Média ponderada} = \frac{(15,4 + 35,4)}{(14,4 + 5,4)} \cong 20,45^{\circ}\text{C}$$

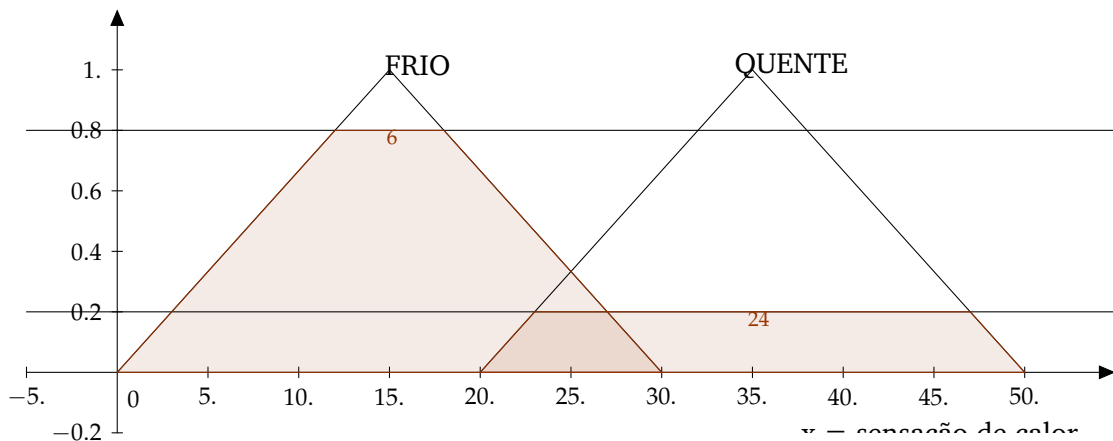


Figura 19: Exemplo defuzzificação centroide

2.10.2 Centro dos Máximos

Neste método são levados em conta apenas as regiões de maior possibilidade entre os possíveis valores da variável que modela o conceito fuzzy em questão. temos então:

$$G(B) = \frac{i + s}{2}, \quad (2.12)$$

onde

$$i = \inf\{u \in \mathbb{R} : \varphi_B(u) = \max_u \varphi_B(u)\}$$

$$s = \sup\{u \in \mathbb{R} : \varphi_B(u) = \max_u \varphi_B(u)\} \quad (2.13)$$

Temos representado este método na figura 20

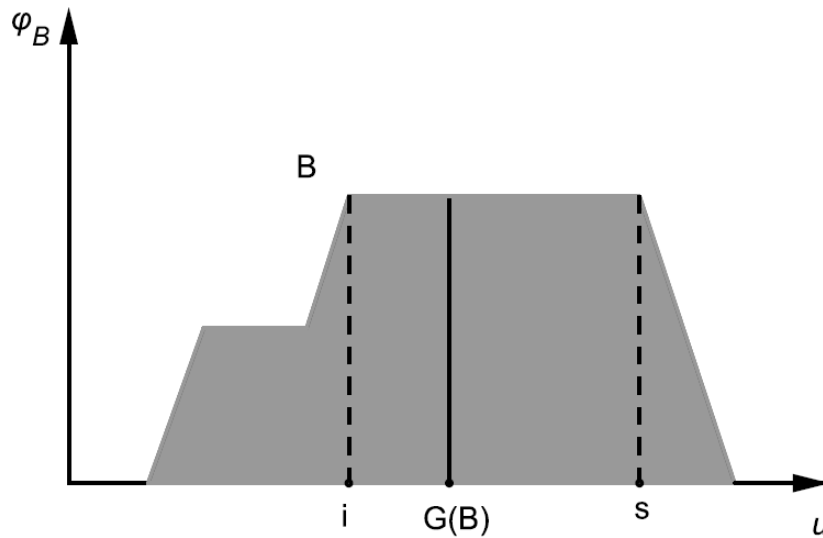


Figura 20: Defuzzificador centro de máximo. [7]

2.10.3 Média dos Máximos

O defuzzificador média dos máximos é comum ser utilizado quando tem-se um domínio discreto, temos a definição a seguir:

$$M(B) = \sum_{i=1}^n \frac{u_i}{n}, \quad (2.14)$$

onde n é dado e u_i , com $1 \leq i \leq n$ são os elementos de maior pertinência ao conjunto fuzzy B .

Exemplo 2.13. No exemplo 19 seriam considerados os valores $x = 15$ e $x = 35$

então $\frac{(15+35)}{2} = 25^\circ\text{C}$

APLICAÇÕES

Abordaremos nesse capítulo o uso de um controlador fuzzy na tomada de decisão da taxa SELIC, incluindo a taxa de desemprego também como variável de entrada, com as informações atualizadas e utilizando o programa R [10] em relação ao artigo [6], assim como o uso da lógica fuzzy no controle de um robô para desviar de obstáculos, a ideia foi criada de modo a demonstrar de maneira lúdica aos alunos o conceito de proximidade utilizando a lógica fuzzy.

3.1 TAXA SELIC

Uma das aplicações da lógica fuzzy é na tomada de decisões, existem situações em que um preço está baixo, razoável ou alto, se um produto é feio, bonito ou adequado, entre outras situações em que não podemos afirmar com certeza, quando temos certeza podemos usar a lógica clássica.

Abordaremos nessa aplicação a decisão da taxa SELIC, baseado no artigo [6]. A taxa SELIC é a taxa básica de juros pelo qual o governo remunera os bancos quando são tomados empréstimos, sendo portanto usada como referência para outras taxas.

A taxa SELIC é muito utilizada como ferramenta de controle de inflação, pois a principal guia da economia é a lei da oferta/procura, por exemplo: quando temos uma safra recorde a quantidade é maior do que as pessoas precisam, então para poder vender o produto é necessário ter um atrativo, o principal utilizado é a diminuição do preço; quando se têm uma quebra de safra, as pessoas ficam dispostas a pagar mais pelo produto que lhes é necessário, portanto acaba forçando o aumento do preço.

A lógica fuzzy será utilizada num processo que consiste em etapas:

- Escolha das variáveis de entrada;
- Funções de pertinência para cada entrada;

- Criar as regras fuzzy adequadas por um especialista;
- Escolher o método de inferência do sistema fuzzy.
- Fazer a defuzzyficação;

No caso da taxa SELIC temos várias entradas possíveis: risco país, desvio da expectativa da inflação, câmbio, dívida líquida, superávit primário, desemprego, capacidade de produção da indústria, índice de confiança do consumidor (ICC), SELIC-Meta defasada. Utilizaremos apenas três nesse trabalho o IPCA (Índice de Preços ao Consumidor Amplo) como referência para inflação, o preço do dólar como referência da situação econômica, e o índice de desemprego (quando a SELIC está alta existe a preferência das pessoas aplicarem seu dinheiro no mercado financeiro do que na atividade produtiva).

Conforme os dados coletados nos sites do IPEA [3] e IBGE [2] colocados na tabela 11, temos uma correlação (conforme cálculo abaixo) entre a entrada IPCA e saída SELIC de 0,79, da entrada Dólar e saída SELIC de 0,9 e da entrada desemprego e saída SELIC de 0,67. Este índice de correlação varia de -1 (alta relação inversamente proporcional) a 1 (alta relação diretamente proporcional), sendo 0 não existe relação das variáveis, portanto essas variáveis são adequadas para serem utilizadas nesse processo.

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X) \cdot \text{Var}(Y)}}$$

onde x_1, x_2, \dots, x_n e y_1, y_2, \dots, y_n , são os valores medidos das variáveis.

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i$$

e

$$\bar{y} = \frac{1}{n} \cdot \sum_{i=1}^n y_i$$

são as médias aritméticas das variáveis.

3.1.1 Programa para modelagem Fuzzy

Existem diversos programas para efetuar a modelagem Fuzzy, entre eles estão o Matlab bastante utilizado, mas é um programa pago, e o R [10] que utilizaremos

por ser gratuito e que pode realizar a modelagem através do inclusão da biblioteca **sets** [9].

No programa **R** foi utilizado para a modelagem o script abaixo:

```
# Carrega a biblioteca de funções de lógica

library(sets)

# Lê os dados da tabela selic

selic < read.table("selic.txt",header = T,dec = ",")

# Define o universo de valores das variáveis do menor até o maior e do incremento

sets_options("universe", seq(from = 1, to = 15, by = 0.01))

# Valor inicial da saída fuzzy

nebula < numeric(0)

# Funções de pertinência para as variáveis de entrada e saída fuzzy

variaveis < set(

  ipca=fuzzy_variable(

    B=fuzzy_trapezoid(corners = c(4,5,6,7)),

    M=fuzzy_trapezoid(corners = c(6,7,8,9)),

    R=fuzzy_trapezoid(corners = c(8,9,10,11))),

  dolar=fuzzy_variable(

    B=fuzzy_cone(1.7, radius = 0.8),

    M=fuzzy_cone(2.7, radius = 0.5),

    R=fuzzy_cone(3.1, radius = 1.4)),

  desemprego=fuzzy_variable(

    B=fuzzy_normal(mean=8,sd = 2),

    R=fuzzy_normal(mean=10,sd = 2)),

  selic=fuzzy_variable(
```

```

baixar=fuzzy_normal(mean=6.5,sd=1.0),
manter=fuzzy_normal(mean=11.3,sd=0.6),
aumentar=fuzzy_normal(mean=16,sd=2.3))

# Regras do módulo fuzzy

regras < set(

fuzzy_rule(ipca %is% B && dolar %is% B && desemprego %is% B, selic %is% baixar),
fuzzy_rule(ipca %is% M && dolar %is% B && desemprego %is% B, selic %is% manter),
fuzzy_rule(ipca %is% R && dolar %is% B && desemprego %is% B, selic %is% manter),
fuzzy_rule(ipca %is% B && dolar %is% B && desemprego %is% R, selic %is% baixar),
fuzzy_rule(ipca %is% M && dolar %is% B && desemprego %is% R, selic %is% baixar),
fuzzy_rule(ipca %is% R && dolar %is% B && desemprego %is% R, selic %is% baixar),
fuzzy_rule(ipca %is% B && dolar %is% M && desemprego %is% B, selic %is% manter),
fuzzy_rule(ipca %is% M && dolar %is% M && desemprego %is% B, selic %is% manter),
fuzzy_rule(ipca %is% R && dolar %is% M && desemprego %is% B, selic %is% aumentar),
fuzzy_rule(ipca %is% B && dolar %is% M && desemprego %is% R, selic %is% baixar),
fuzzy_rule(ipca %is% M && dolar %is% M && desemprego %is% R, selic %is% manter),
fuzzy_rule(ipca %is% R && dolar %is% M && desemprego %is% R, selic %is% manter),
fuzzy_rule(ipca %is% B && dolar %is% R && desemprego %is% B, selic %is% aumentar),
fuzzy_rule(ipca %is% M && dolar %is% R && desemprego %is% B, selic %is% aumentar),
fuzzy_rule(ipca %is% R && dolar %is% R && desemprego %is% B, selic %is% aumentar),
fuzzy_rule(ipca %is% B && dolar %is% R && desemprego %is% R, selic %is% manter),
fuzzy_rule(ipca %is% M && dolar %is% R && desemprego %is% R, selic %is% aumentar),
fuzzy_rule(ipca %is% R && dolar %is% R && desemprego %is% R, selic %is% aumentar)

)

```

```

# Criação do sistema fuzzy

sistema <- fuzzy_system(variaveis , regras)

# Inferencia e defuzzificacao do sistema fuzzy

for(i in 1:48){

  fi <- fuzzy_inference(sistema , list(ipca=selic[i,2] , dolar=selic[i,3] , desemprego=selic[i,4]))

  nebula[i] <- gset_defuzzify(fi , "centroid")

}

# Verificação dos valores da saída fuzzy

nebula

# Mostra como está montado o sistema fuzzy

print(sistema)

# Gráfico de pertinência das variáveis fuzzy

plot(sistema)

# variação do eixo x do gráfico de comparação das selic fuzzy e real

meses <- 1:48

# gráfico de comparação das selic fuzzy e real

plot(meses , selic$selic , type="l" , main="Comparação_SELIC_Fuzzy_e_Real" ,
      xlab="Jan13_à_Dez16" , ylab="Selic" , col="red")

legend(40,10,c("Fuzzy" , "Real") , col =c("blue" , "red") , pch=rep(20,2))

lines(meses , nebula , col="blue")

# correlação entre selic real e fuzzy

x=cbind(selic$selic , nebula)

x=data.frame(x)

cor(x , method='pearson')

```

3.1.2 Variáveis de entrada e saída

Nas variáveis de entrada foram utilizadas: **B** para cenário **bom**, **M** para cenário **moderado** e **R** para cenário **ruim**, os dados foram obtidos na internet nos sites do IBGE [2] e Banco Central [1], para a saída SELIC temos as decisões de baixar, manter ou aumentar. Os gráficos para cada variável podem ser vistos nas figuras 21 , 22 , 23 e 24.

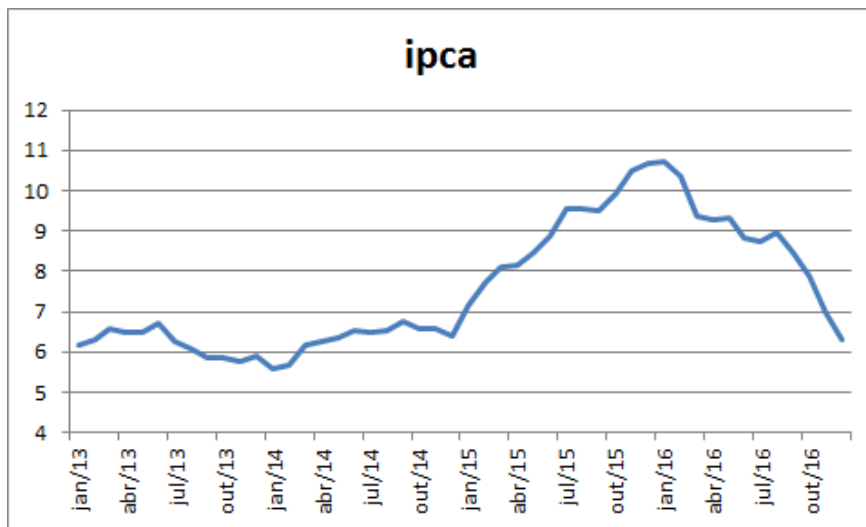


Figura 21: Entrada variável IPCA

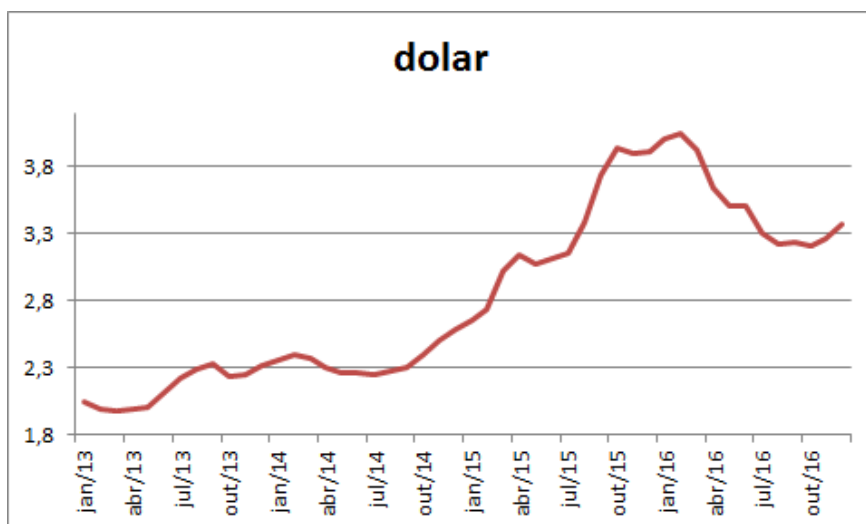


Figura 22: Entrada variável Dólar

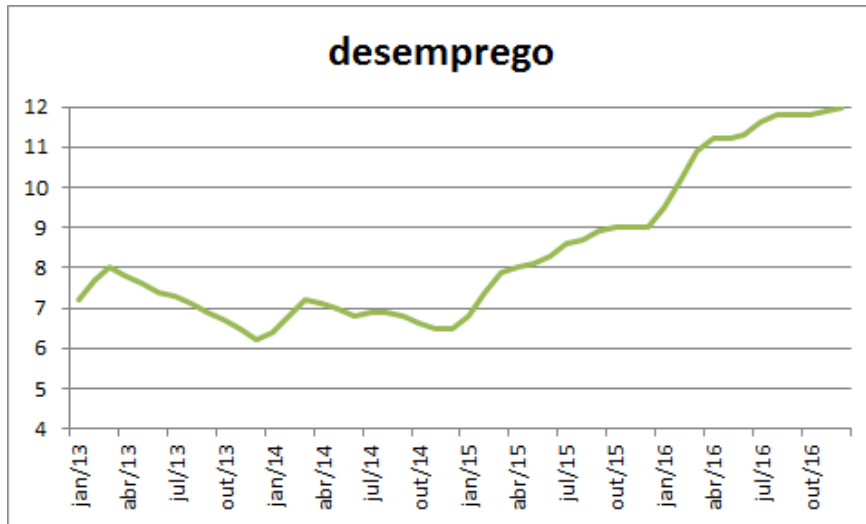


Figura 23: Entrada variável Desemprego

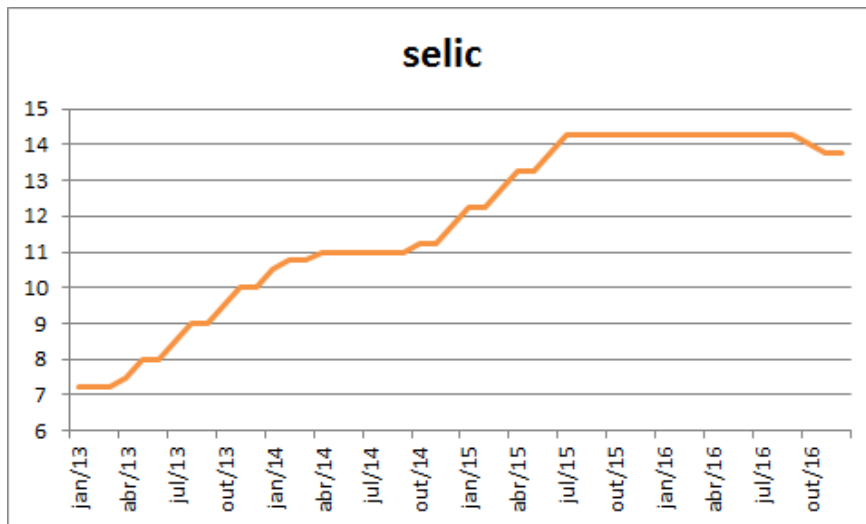


Figura 24: Saída variável SELIC

3.1.3 Funções de Pertinência

Depois de escolhidas as variáveis de entrada que têm relação e interferem na variável de saída em um sistema fuzzy, será necessário construir as funções de pertinência, essas funções são geralmente construídas por especialistas, no nosso caso, as variáveis escolhidas são constantemente citadas nas atas do COPOM (Comitê de Política Monetária), com a informação destas atas mais as colhidas na internet e adequando com a

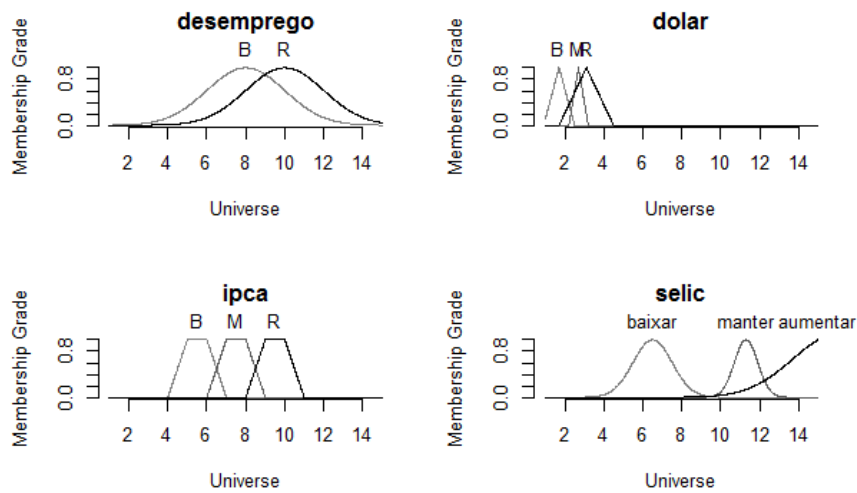


Figura 25: Funções de pertinência

saída do sistema utilizando o coeficiente de correlação de Pearson entre a SELIC real e a SELIC fuzzy, chegamos aos gráficos 25.

3.1.4 Regras

Assim como as funções de pertinência, as regras também devem ser feitas por especialistas, as regras são as tomadas de decisão conforme os valores das variáveis de entrada, produzindo uma saída fuzzy, as regras adotadas nessa aplicação podem ser vistas na tabela 12.

3.1.5 Inferência e defuzzificação

Os valores das variáveis de entrada conforme as regras produzem uma saída fuzzy (temos um exemplo na figura 26), que no nosso caso pode ser útil para a tomada de decisão, mas se quisermos ter um valor para comparar com a SELIC real, teremos que inferir e defuzzificar as saídas fuzzy, utilizamos o centroide (centro de área) para isso.

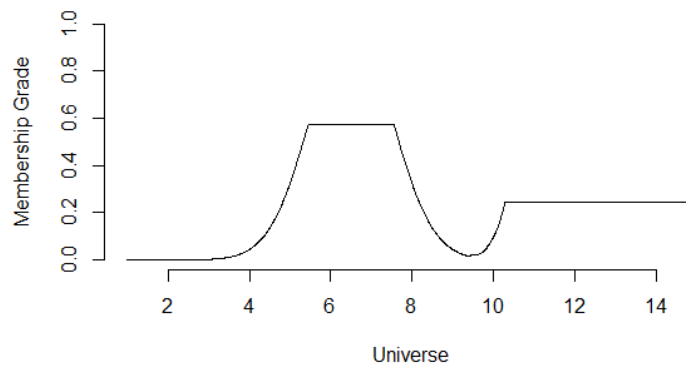


Figura 26: Saída sistema fuzzy com valores de janeiro 2013

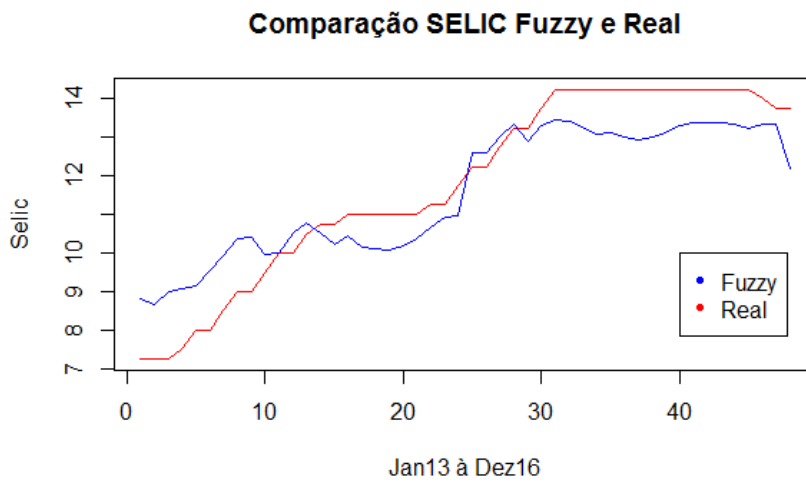


Figura 27: Comparação SELIC real e fuzzy

3.1.6 Conclusão

A modelagem fuzzy pode ser um método bastante útil para tomada de decisão, conforme podemos ver no gráfico 27, onde a saída fuzzy têm um grau de correlação de 0,95 comparada com a SELIC real.

3.2 USO DA LÓGICA FUZZY EM PROXIMIDADES

Quando encontramos situações de estar próximo ou perto de algo, qual valor podemos adotar para dizer que é perto ou próximo? Nesta aplicação mostraremos de forma prática que podemos adotar a lógica fuzzy como forma de determinar valores próximos de um valor fornecido.

Iremos utilizar a seguinte situação, um robô aproxima-se de uma fonte de calor, sendo que ele suporta uma temperatura em torno de 50°C, portanto devendo parar quando aproximar-se deste valor. Por quê não utilizar a lógica clássica para isso, fazendo o robô parar com valores acima de 50°C? porque o robô trabalha com sensores que podem ter erros de leitura, fazendo o robô ficar exposto a temperaturas maiores, consequentemente o danificando.

3.2.1 Programa para simulação de robô

Utilizaremos **LOGO** [8] que é uma linguagem de programação interpretada (podemos ver passo a passo a execução dos comandos), muito utilizada para educação foi criada por Wally Feurzeig e Seymour Papert como uma forma construcionista onde os alunos aprendem com seus erros através da visualização dos comandos. A forma clássica da linguagem **LOGO** é o controle de uma tartaruga que movendo-se na tela desenha por onde passa.

Existem vários compiladores diferentes, podendo ser utilizados para controle de robôs através da saída física da porta paralela dos computadores ou até para IA (inteligência artificial), na nossa aplicação utilizaremos o **Kturtle** [5] que é uma linguagem de programação gratuita no sistema operacional **LINUX**.

Vemos abaixo os comandos para o **Kturtle**:

```
#reinicia o programa
apague

#tamanho da tela
tt 200,200

#cor do fundo da tela
cf 0,0,0

#calor suportado
$calor = 50

#fonte calor aleatorio
```



```

$fonte = al 200,510

#cor da fonte
$cor = arredonda($fonte/4)

#desenha a fonte
ll 8
vápara 96,0
pd 90
cl 0,0,$cor
pf 8
pd 90
pf 8
pd 90
pf 8
pd 90
pf 8

#volta inicio
centralize

#desativa o risco
un
$scal = 0
$scalfuzzy = 0

#repete ate checar perto
enquanto $scalfuzzy < 1 {

#distancia com erro
$dist = pegay*(al 0.95,1.05)

#calor dependente da distancia
$scal = $fonte/(raizquadrada $dist)

#calor aproximado por fuzzy
se $scal < ($calor 0.1*$calor){$scalfuzzy = 0}
se ($calor 0.1*$calor) <= $scal e $scal <= $calor
{ $scalfuzzy = ($scal ($calor 0.1*$calor))/(0.1*$calor)}
se $scal > $calor {$scalfuzzy =1}

#movimenta tartaruga
pf 1}
vápara 0,100

#some tartaruga
da

```

```

#cor do texto
cl 255,255,255

#valor final do calor
mostre " calor final:"
vápara 100,100
mostre $cal

```

3.2.2 Comentários

- A fonte de calor varia de 200°C à 510°C variando também a cor da representação na tela do turtle;
- Conforme a distância o calor diminui seguindo a função $calor = \frac{\text{calor da fonte}}{\sqrt{\text{distancia}}}$;
- Foi utilizada como função de aproximação $f(x) = \frac{x-(50-0,1.50)}{0,1.50}$ conforme figura 28;
- Aplicou-se um erro de $\pm 5\%$ na distância em função do erro de leitura do sensor do robô.

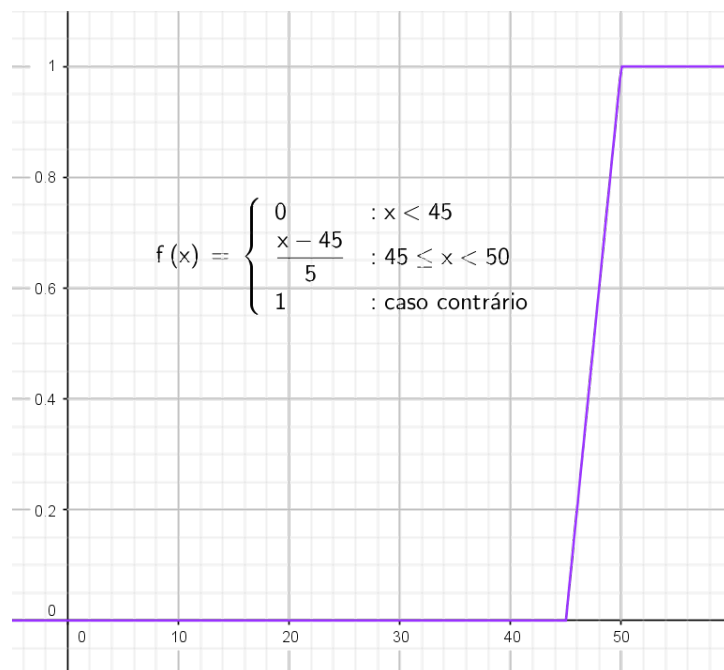


Figura 28: Função de aproximação

3.2.3 *Uso da aplicação*

Imaginemos um robô no centro de uma sala escura, onde somente existe uma saída onde advém uma luz, o robô possui 3 sensores: um sensor de luminosidade que indicará a saída, outro sensor de proximidade que verifica se o robô está perto da parede, para não haver colisão, e um sensor de calor usando a aplicação anterior como forma de obstáculo, determinando pela lógica fuzzy o valor suportado pelo robô. O robô se movimenta para frente e somente vira para a esquerda ou direita se estiver perto da fonte de calor ou estiver perto da parede, vemos na figura 29 um exemplo da execução do programa.

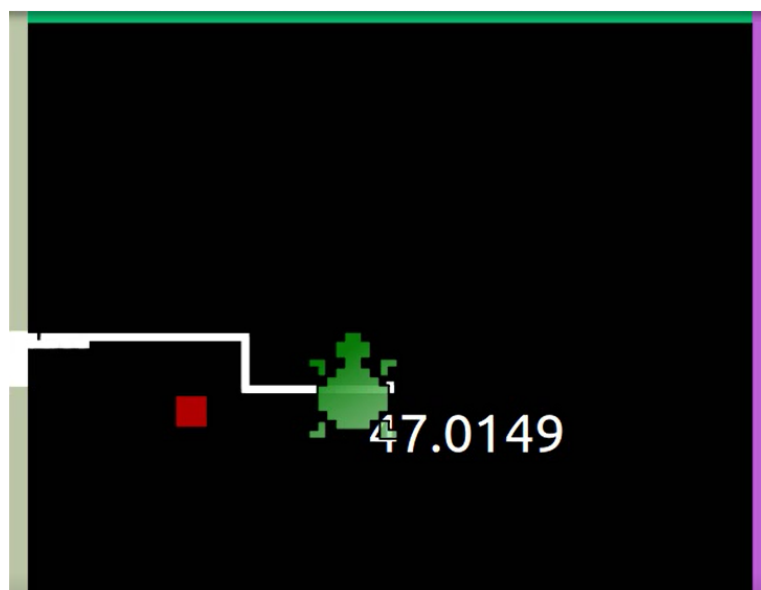


Figura 29: Percurso da tartaruga com desvio e calor final

Usando essas premissas chegamos ao código abaixo:

```
#desenha as paredes com cores aleatorias
apague
tt 200,200
cf 0,0,0
$perto = 20
vápara 0,0
ll 5
repita 4 {
pd 90
$r = al 0,255
$g = al 0,255
$b = al 0,255
cl $r,$g,$b
```

```

pf 200 }

#define a posição da porta
$x = arredonda (al 1,4)
cl 255,255,255
se $x==1 { $px = al 0,190
vápara $px,0
pd 90
pf 10}
senão se $x == 2 { $py = al 0,190
vápara 200,$py
pd 180
pf 10}
senão se $x == 3 { $px = al 0,190
vápara $px,200
pd 90
pf 10}
senão
{$py = al 0,190
vápara 0,$py
pd 180
pf 10}
$dir = arredonda((lerdireção / 90) + 1)
se $x==1 {$portax = $px
$portay = 0 }
senão se $x == 2 {$portax = 200
$portay = $py }
senão se $x == 3 {$portax = $px
$portay = 200 }
senão {$portax = 0
$portay = $py}
$distatual = raizquadrada((pegax $portax)^2 + (pegay $portay)^2)
$distpf = 90

#define posição da fonte de calor
enquanto $distpf>70 ou $distpf < 30 {
$fontex = al 10,190
$fontey = al 10,190
$distpf = raizquadrada(($fontex $portax)^2 + ($fontey $portay)^2)
}
$final=0

#calor suportado
$calor = 50

#fonte calor aleatorio
$fonte = al 400,1020

```

```

#cor da fonte
$cor = arredonda($fonte/4)

#desenha a fonte
ll 4
vápara $fontex,$fontey
pd 90
cl $cor,0,0
pf 4
pd 90
pf 4
pd 90
pf 4
pd 90
pf 4

#volta inicio
centralize
cl 255,255,255
ll 2
$cal = 0
$calfuzzy = 0

#verifica sw chegou na porta
enquanto $perto != 0 {
$distant = $distatual
pf 1
$distatual = raizquadrada((pegax $portax)^2 + (pegay $portay)^2)

#distancia com erro
$distf = raizquadrada((pegax $fontex)^2 + (pegay $fontey)^2)*(al 0.95,1.05)
#calor dependente da distancia
$cal = $fonte / ($distf)

#calor aproximado por fuzzy
se $cal < ($calor 0.1*$calor){$calfuzzy = 0}
se ($calor 0.1*$calor) <= $cal e $cal <= $calor
{ $calfuzzy = ($cal ($calor 0.1*$calor))/(0.1*$calor)}
se $cal > $calor {$calfuzzy =1}

se $dir == 1 {$distparede = pegay}
senão {se $dir == 2 {$distparede = 200 pegax}
senão {se $dir == 3 {$distparede = 200 pegay}
senão {$distparede = pegax}}}

#verifica se esta perto da fonte de calor ou da parede
se ($calfuzzy > 0) ou (($calfuzzy < 1) e ($distatual > $distant)) ou (($calfuzzy < 1)
e ($distparede < $perto)) {

```

```

#vira para esquerda ou direita depende da distancia para a saida mais perto
se $dir == 1 {
$c11 = $fonte / ( raizquadrada(((pegax 1) $fontex)^2 + (pegay $fontey)^2)))
$c12 = $fonte / ( raizquadrada((pegax $fontex)^2 + (pegay $fontey)^2)))

$d11=raizquadrada(((pegax 1) $portax)^2 + (pegay $portay)^2)
$d12=raizquadrada((pegax $portax)^2 + (pegay $portay)^2)
se $scalfuzzy > 0 {se $c11 < $c12 { pe 90
pf 1
$perto = $perto + 1} senão {pd 90
pf 1
$perto = $perto + 1} } senão {se $d11 < $d12 { pe 90
$perto = $perto 1} senão {pd 90
$perto = $perto 1} }}

se $dir == 2 {
$c21 = $fonte / ( raizquadrada((pegax $fontex)^2 + ((pegay 1) $fontey)^2)))
$c22 = $fonte / ( raizquadrada((pegax $fontex)^2 + (pegay $fontey)^2)))
$d21=raizquadrada((pegax $portax)^2 + ((pegay 1) $portay)^2)
$d22=raizquadrada((pegax $portax)^2 + (pegay $portay)^2)

se $scalfuzzy > 0 {se $c21 < $c22 { pe 90
pf 1
$perto = $perto + 1} senão {pd 90
pf 1
$perto = $perto + 1} } senão {se $d21 < $d22 { pe 90
$perto = $perto 1} senão {pd 90
$perto = $perto 1} }}

se $dir == 3 {
$c31 = $fonte / ( raizquadrada(((pegax 1) $fontex)^2 + (pegay $fontey)^2)))
$c32 = $fonte / ( raizquadrada((pegax $fontex)^2 + (pegay $fontey)^2)))
$d31=raizquadrada(((pegax 1) $portax)^2 + (pegay $portay)^2)
$d32=raizquadrada((pegax $portax)^2 + (pegay $portay)^2)

se $scalfuzzy > 0 {se $c31 < $c32 { pd 90
pf 1
$perto = $perto + 1} senão {pe 90
pf 1
$perto = $perto + 1} } senão {se $d31 < $d32 { pd 90
$perto = $perto 1} senão {pe 90
$perto = $perto 1} }}

se $dir == 4 {
$c41 = $fonte / ( raizquadrada((pegax $fontex)^2 + ((pegay 1) $fontey)^2)))
$c42 = $fonte / ( raizquadrada((pegax $fontex)^2 + (pegay $fontey)^2)))
$d41=raizquadrada((pegax $portax)^2 + ((pegay 1) $portay)^2)
$d42=raizquadrada((pegax $portax)^2 + (pegay $portay)^2)

```

```

se $calfuzzy > 0 {se $c41 < $c42 { pd 90
pf 1
$perto = $perto + 1} senão {pe 90
pf 1
$perto = $perto + 1} } senão {se $d41 < $d42 { pd 90
$perto = $perto - 1} senão {pe 90
$perto = $perto - 1} }}
$dir = arredonda((lerdireção / 90) + 1)
se $dir>4 ou $dir<=0 {$dir=1}
se $final<$cal {$final=$cal}
}}

#mostra o calor final
vápara 90,100
direção 0
mostre $final

```

3.2.4 Conclusão

Embora a decisão do desvio de um obstáculo possa ser feita pela lógica clássica, usando a lógica fuzzy podemos ter a vantagem de poder desviar da fonte de calor pela proximidade, isto é, quanto mais próximo maior poderá ser esse desvio, embora não tenha sido utilizado isto na aplicação pois o código seria muito extenso e a finalidade da aplicação é ser a mais didática possível, podemos também utilizar a pertinência para o caso do robô ter um extintor de incêndio e variando a quantidade de líquido expelido de acordo com a proximidade, decidir desviar quando tiver mais de um obstaculo para longe do mais próximo, além de outras que possam utilizar a lógica fuzzy. Este método também poderá ser utilizado para qualquer tipo de obstáculo no percurso de um robô.

Data	IPCA	Dolar	Desemprego	SELIC
jan/13	6,15	2,04	7,2	7,25
fev/13	6,31	1,99	7,7	7,25
mar/13	6,59	1,98	8	7,25
abr/13	6,49	1,99	7,8	7,5
mai/13	6,5	2,01	7,6	8
jun/13	6,7	2,11	7,4	8
jul/13	6,27	2,22	7,3	8,5
ago/13	6,09	2,29	7,1	9
set/13	5,86	2,33	6,9	9
out/13	5,84	2,24	6,7	9,5
nov/13	5,77	2,25	6,5	10
dez/13	5,91	2,32	6,2	10
jan/14	5,59	2,36	6,4	10,5
fev/14	5,68	2,4	6,8	10,75
mar/14	6,15	2,37	7,2	10,75
abr/14	6,28	2,3	7,1	11
mai/14	6,37	2,26	7	11
jun/14	6,52	2,26	6,8	11
jul/14	6,5	2,25	6,9	11
ago/14	6,51	2,27	6,9	11
set/14	6,75	2,3	6,8	11
out/14	6,59	2,4	6,6	11,25
nov/14	6,56	2,5	6,5	11,25
dez/14	6,41	2,59	6,5	11,75
jan/15	7,14	2,65	6,8	12,25
fev/15	7,7	2,73	7,4	12,25
mar/15	8,13	3,02	7,9	12,75
abr/15	8,17	3,14	8	13,25
mai/15	8,47	3,07	8,1	13,25
jun/15	8,89	3,12	8,3	13,75
jul/15	9,56	3,16	8,6	14,25
ago/15	9,53	3,39	8,7	14,25
set/15	9,49	3,73	8,9	14,25
out/15	9,93	3,94	9	14,25
nov/15	10,48	3,9	9	14,25
dez/15	10,67	3,91	9	14,25
jan/16	10,71	4,01	9,5	14,25
fev/16	10,36	4,05	10,2	14,25
mar/16	9,39	3,92	10,9	14,25
abr/16	9,28	3,64	11,2	14,25
mai/16	9,32	3,51	11,2	14,25
jun/16	8,84	3,51	11,3	14,25
jul/16	8,74	3,3	11,6	14,25
ago/16	8,97	3,22	11,8	14,25
set/16	8,48	3,23	11,8	14,25
out/16	7,87	3,21	11,8	14
nov/16	6,99	3,26	11,9	13,75
dez/16	6,29	3,37	12	13,75

Tabela 11: Tabela de valores das entradas fuzzy.

Ipca	Dolar	Desemprego	Selic
B	B	B	baixar
M	B	B	manter
R	B	B	manter
B	B	R	baixar
M	B	R	baixar
R	B	R	baixar
B	M	B	manter
M	M	B	manter
R	M	B	aumentar
B	M	R	baixar
M	M	R	manter
R	M	R	manter
B	R	B	aumentar
M	R	B	aumentar
R	R	B	aumentar
B	R	R	manter
M	R	R	aumentar
R	R	R	aumentar

Tabela 12: Regras de Inferência

BIBLIOGRAFIA

- [1] Banco Central dados econômicos. <http://www.bcb.gov.br/pt-br/#!/home>. Accessed: 2017-08-01.
- [2] IBGE ipca. http://www.ibge.gov.br/home/estatistica/indicadores/precos/inpc_ipca/defaultinpc.shtm. Accessed: 2017-08-01.
- [3] IPEA dados econômicos. <http://www.ipeadata.gov.br/Default.aspx>. Accessed: 2017-08-01.
- [4] Michal Baczynski and Balasubramaniam Jayaram. *Fuzzy implications*. Springer, 2008.
- [5] Cies Breijjs, Anne-Marie Mahfouf, and Mauricio Piacentini. Manual do kturtle. *Acesso em*, 16, 2013.
- [6] Marco Antonio Leonel Caetano. *Lógica Fuzzy para tomada de decisão em negócios e finanças*. Revista de Economia e Administração, 2006.
- [7] Laécio Carvalho de Barros and Rodney Carlos Bassanezi. *Tópicos de lógica fuzzy e biomatemática*. Editora da Unicamp, 2006.
- [8] Wally Feurzeig and Seymour Papert. Logo. *ODP-Open Directory Project*, 1967.
- [9] David Meyer and Kurt Hornik. Generalized and customizable sets in R. *Journal of Statistical Software*, 31(2):1–27, 2009.
- [10] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.
- [11] Timothy J Ross. *Fuzzy logic with engineering applications*. John Wiley & Sons, 2009.
- [12] Leo Weber and Pedro Antonio Trierweiler Klein. *Aplicação da lógica fuzzy em software e hardware*. Editora da ULBRA, 2003.