

**Universidade Federal de Santa Catarina - UFSC**  
**Curso de Pós-Graduação em Ciências da Computação**

**ARANDÚ: UM SISTEMA ESPECIALISTA DIFUSO PARA O  
DIAGNÓSTICO DE HEPATOPATIAS CRÔNICAS**

**por**

**Angel Freddy Godoy Viera**

**Dissertação submetida à Universidade Federal de Santa Catarina para a  
obtenção do grau de Mestre em Ciências da Computação**

**Prof. Luiz Fernando Jacintho Maia, Dr.**

**Orientador**

**Florianópolis, Fevereiro de 1996**

# ARANDÚ: UM SISTEMA ESPECIALISTA DIFUSO PARA O DIAGNÓSTICO DE HEPATOPATIAS CRÔNICAS

ANGEL FREDDY GODOY VIERA

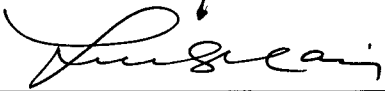
ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO  
DE

## MESTRE EM CIÊNCIA DA COMPUTAÇÃO

ESPECIALIDADE SISTEMAS DE CONHECIMENTO E APROVADA EM SUA FORMA  
FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO.

### BANCA EXAMINADORA:

  
Prof. Rogério Cid Bastos, Dr.  
Coordenador do Curso

  
Prof. Luiz Fernando Jacintho Maia, Dr.  
Orientador

  
Prof. Rogério Cid Bastos, Dr.

  
Prof. Vanir Cardoso, Dr.

  
Prof. Fernando Ostuni Gauthier, Dr.

***Dedico este trabalho***

*Com muito amor e carinho a meus queridos pais Gregorio Godoy Martínez e Rita Viera de Godoy, que deram tudo de si, para me incentivar e orientar com palavras, atitudes e exemplo para ter um espírito de superação contínua na vida.*

*A meus queridos irmãos César Gregorio Godoy Viera, Sonia Dominga Godoy Viera e Lourdes Elizabeth Godoy Viera, pelo contínuo apoio e ajuda que me deram em todo momento.*

***A mondó co tembiapó***

*Taihú ha cunu'ú che tuva cuerape Gregorio Godoy Martínez ha Rita Viera de Godoy, o me'éngatueté haguére maimá icatuva ha ndóicatuvaí aveí, che mbo temimokire'y ha mbohecocuaá, ñe'épe ha techaukape añembotuvichá haguá co tecovépe.*

*Che hermano cuerape avei che a haihueteva: César Gregorio Godoy Viera, Sonia Dominga Godoy Viera e Lourdes Elizabeth Godoy Viera, che pytyvó haguére maimaba che aicotevebape.*

***Dedico este trabajo***

*Con mucho amor y cariño a mis queridos padres Gregorio Godoy Martínez y Rita Viera de Godoy, que dieron todo de si, para incentivar y orientarme con palabras, actitudes y ejemplos para tener siempre un espíritu de superación continua en la la vida.*

*A mis queridos hermanos Cesar Gregorio Godoy Viera, Sonia Dominga Godoy Viera y Lourdes Elizabeth Godoy Viera, por el contínuo apoyo y ayuda que me dieron en todo momento.*

## AGRADECIMENTOS

Agradeço em primeiro lugar a Deus pelas graças que me há concedido na vida, especialmente pela inteligência e saúde necessária para concluir meus estudos.

A meus pais Gregorio Godoy Martínez e Rita Viera de Godoy, pelos ensinamentos que me deram e continuam dando, que constituiu e constituem o maior tesouro que norteia minha vida em todos os aspectos, pelo apoio incondicional e a força que deram durante todo este caminhar na realização deste curso.

A meus irmãos o Mestre Eng. César Gregorio Godoy Viera e a Mestranda Lic. Sonia Dominga Godoy Viera que compartilhamos juntos momentos inolvidáveis de nossas vidas, sobrelevando dificuldades, compartilhando triunfos e alegrias neste país Brasil que nos acolheu para realizar nossos estudos de Pós-Graduação. À minha querida irmã Lourdes Elizabeth Godoy Viera que longe na distancia mas muito perto no sentimento e espírito, sempre nos deu o maior apoio, para seguir sempre adiante.

A meu orientador o Prof. Luiz Fernando Jacintho Maia, por confiar sempre na minha capacidade, as orientações dadas e a amizade brindada.

Aos médicos Dr. Vanir Cardoso e Dr. Tiago Ramos do Hospital Universitário, pelas valiosas contribuições e o tempo dispendido nas longas entrevistas, sem as quais a conclusão deste trabalho não seria possível.

Ao Prof. Hermann Adolf Harry Lucke (in memoriam).

A minha mãe brasileira Vera Lúcia (Verinha), que com o espírito sempre alegre e rosto sorridente, brindara sua ajuda desinteressada em todos os momentos que precisava.

Ao Dr. Ricardo Franco e Antonio Figueredo pela amizade, confiança, conselhos e incentivo brindado.

Ao Curso de Pós-Graduação em Ciências da Computação e seu corpo docente, pela oportunidade de fazer parte do curso e os ensinamentos que me deram nestes dois anos.

À Universidade Federal de Santa Catarina e ao CNPq, que proporcionaram as condições necessárias para a realização desse trabalho.

*À Honorable Cámara de Senadores del Paraguay*, pelo apoio brindado.

A meu time de tradutoras e amigas, Anita, Angelita e Ana Paula, pelas correções do português nos meus trabalhos.

Aos meus colegas e amigos do curso com quem compartilhei nestes anos de estudo.

A todos meus parentes, amigos e colegas do Paraguai, que sempre desejaram o melhor para mim.

E a todas as pessoas que de alguma forma ou outra contribuíram para a conclusão deste trabalho.

## RESUMO

Os sistemas especialistas difusos têm apresentado grande aplicabilidade, devendo-se isso ao fato de que tratam a expressão da ambigüidade da linguagem e seus conceitos e, além disso, introduzem novos métodos para a aquisição, representação e utilização do conhecimento. Na área de diagnóstico médico vem aumentando o interesse neste tipo de sistemas, devido ao fato de que nela os dados que indicam uma condição normal ou anormal, geralmente, não têm limites bem definidos.

Neste trabalho é proposta a utilização de um sistema especialista difuso para o diagnóstico de hepatopatias crônicas, na área de gastroenterologia. Tal sistema utiliza uma base de conhecimento que tem regras de inferência difusas, e a expressão dos dados clínicos e laboratoriais estão estruturados na forma de conjuntos difusos.

## **ABSTRACT**

Fuzzy Expert Systems have presented broad acceptance because it deals with the ambiguity of language and concepts. It introduces new approaches to knowledge acquisition, representation and utilization. In the medical diagnostic domain, the interest in this kind of systems has improved, because the figures that indicate normal or abnormal conditions, usually, don't have a well defined boundary, having fuzzy frontiers.

This paper proposes the use of fuzzy expert systems for the diagnostic of chronic liver diseases, in Gastro-Enterology domain. This system uses knowledge of the clinical and laboratory data structured in the form of fuzzy sets.

# SUMÁRIO

<b>CAPÍTULO I</b>	<b>1</b>
<b>1.1 Introdução</b>	<b>1</b>
<b>1.2 Objetivos</b>	<b>2</b>
<b>1.2.1 Objetivos Gerais</b>	<b>2</b>
<b>1.2.2 Objetivos Específicos</b>	<b>2</b>
<b>1.3 Estrutura do trabalho</b>	<b>2</b>
<b>CAPÍTULO II</b>	<b>4</b>
<b>2 INTELIGÊNCIA ARTIFICIAL</b>	<b>4</b>
<b>2.1 DEFINIÇÃO E HISTÓRIA</b>	<b>4</b>
<b>2.2 Questões importantes na IA</b>	<b>7</b>
<b>2.2.1 Suposição subjacente</b>	<b>7</b>
<b>2.2.2 Técnica de IA</b>	<b>8</b>
<b>2.2.3 O nível do modelo</b>	<b>8</b>
<b>2.2.4 Critérios de sucesso</b>	<b>9</b>
<b>CAPÍTULO III</b>	<b>11</b>
<b>3 LÓGICA DIFUSA</b>	<b>11</b>
<b>3.1 Introdução</b>	<b>11</b>
<b>3.2 Quantificação da ambigüidade</b>	<b>14</b>
<b>3.3 Conjuntos difusos</b>	<b>15</b>
<b>3.3.1 Definição</b>	<b>15</b>
<b>3.3.2 Características de uma função de pertinência</b>	<b>17</b>



<b>3.4 Operações com conjuntos difusos</b>	<b>19</b>
<b>3.4.1 Operações básicas com conjuntos difusos</b>	<b>19</b>
3.4.1.1 União	20
3.4.1.2 Interseção	20
3.3.1.3 Complemento	20
3.3.1.4 De Morgan	21
<b>3.5 Operações algébricas</b>	<b>22</b>
3.5.1 Produto Cartesiano	22
3.5.2 Enésima potência	22
3.5.3 Soma algébrica	24
3.5.4 Soma limitada	24
3.5.5 Diferença limitada	24
3.5.6 Produto Algébrico	25
<b>3.6 Classes de operadores</b>	<b>26</b>
<b>3.7 Propriedades dos conjuntos difusos</b>	<b>27</b>
<b>3.8 Tipos de conjuntos difusos</b>	<b>29</b>
3.8.1 Conjunto difuso TIPO 1	29
3.8.2 Conjunto difuso TIPO 2	29
3.8.3 Conjunto difuso TIPO m	29
3.8.4 $\alpha$ -Cortes (Conjunto de nível $\alpha$ )	30
<b>3.9 Relações difusas</b>	<b>31</b>
3.9.1 Introdução	31
3.9.2 Relação difusa	31
3.9.3 Matrizes e grafos difusos	32
<b>3.10 Operações com relações difusas</b>	<b>34</b>
<b>3.11 Composição de uma relação difusa (RD)</b>	<b>34</b>
<b>3.12 Relação difusa complementar</b>	<b>35</b>

	x
<b>3.13 Propriedades básicas das relações difusas</b>	<b>36</b>
<b>3.14 Proposição difusa</b>	<b>36</b>
<b>CAPÍTULO IV</b>	<b>39</b>
<b>4 SISTEMAS ESPECIALISTAS</b>	<b>39</b>
<b>4.1 Definição e história</b>	<b>39</b>
<b>4.2 Conhecimento e técnica</b>	<b>42</b>
<b>4.3 Característica de um sistema especialista</b>	<b>43</b>
<b>4.3.1 Corpo de conhecimento</b>	<b>43</b>
<b>4.3.2 Conhecimento de alto nível</b>	<b>43</b>
<b>4.3.3 Poder de modelamento preditivo</b>	<b>43</b>
<b>4.3.4 Memória Institucional</b>	<b>44</b>
<b>4.3.5 Facilidade de treinamento</b>	<b>44</b>
<b>4.4 Participantes na construção de um SE</b>	<b>45</b>
<b>4.4.1 Sistema especialista</b>	<b>45</b>
<b>4.4.2 O Especialista no Domínio</b>	<b>46</b>
<b>4.4.3 O Engenheiro de Conhecimento</b>	<b>46</b>
<b>4.4.4 A ferramenta de construção do sistema especialista</b>	<b>46</b>
<b>4.4.5 Usuário Final</b>	<b>46</b>
<b>4.5 Vantagens e desvantagens de um sistema especialista</b>	<b>47</b>
<b>4.6 Estrutura de um sistema especialista</b>	<b>48</b>
<b>4.7 Etapas na construção de sistemas especialistas</b>	<b>49</b>
<b>4.7.1 Identificação</b>	<b>49</b>
<b>4.7.2 Conceitualização</b>	<b>49</b>
<b>4.7.3 Formalização</b>	<b>49</b>
<b>4.7.4 Implementação</b>	<b>50</b>
<b>4.7.5 Testes</b>	<b>50</b>

	xi
4.8 Adquirindo o conhecimento de um especialista	51
4.9 Sistemas especialistas aplicados à área médica	52
<b>CAPÍTULO V</b>	<b>55</b>
<b>5 SISTEMAS ESPECIALISTAS DIFUSOS (SED)</b>	<b>55</b>
5.1 Introdução	55
5.2 Sistema de inferência difusa	57
5.2.1 Base de dados	58
5.2.2 Fuzzificação	58
5.2.3 Base de Regras	59
5.2.4 Unidade de escalonamento	59
5.2.5 Defuzzificação	59
5.3 Tipos de raciocínio difuso	60
5.4 Expressão das regras em forma canônica num sistema especialista difuso baseado em regras (SEDBR)	61
5.4.1 Múltiplos antecedentes e múltiplos consequentes unidos por “AND” e “OR”	64
5.4.2 Sentenças condicionais com “ELSE”	65
5.4.3 Regras IF-THEN aninhadas	65
5.5 Agregação de regras	66
5.6 Classes especiais de modelos de sistemas especialistas difusos baseados em regras	66
5.7 Aplicação de sistemas especialistas difusos no diagnóstico médico	68
5.8 Métodos utilizados para o tratamento de ambigüidade no diagnóstico médico	69
5.8.1 Diagnóstico utilizando relações difusas	70
5.8.2 Diagnóstico utilizando padrões de sintomas e graus de conformidade	72
5.8.3 Aplicação do engenheiro do conhecimento no diagnóstico	74
5.9 Exemplos de sistemas especialistas na área médica que aplicam lógica difusa	76

<b>CAPITULO VI</b>	<b>78</b>
<b>6 ESTUDO DA FERRAMENTA</b>	<b>78</b>
<b>6.1 Introdução</b>	<b>78</b>
<b>6.2 Comparando KAPPA-PC a processadores convencionais de dados</b>	<b>79</b>
<b>6.3 Programação Orientada a Objetos (POO)</b>	<b>79</b>
6.3.1 Características da POO	79
6.3.2 Vantagens da POO	80
<b>6.4 Representação dos objetos em KAPPA</b>	<b>80</b>
6.4.1 Classes e instâncias	81
<b>6.5 Vantagens em utilizar classes na representação de grupos de instâncias</b>	<b>82</b>
<b>6.6 Organizando os objetos com herança</b>	<b>83</b>
<b>6.7 Representação dos processos</b>	<b>83</b>
6.7.1 Métodos	84
6.7.2 Funções	85
6.7.3 Regras	85
<b>6.8 Regras vs. Programação Convencional</b>	<b>86</b>
<b>6.9 Tipos de Raciocínio em KAPPA-PC</b>	<b>86</b>
6.9.1 Raciocínio para Frente ( <i>Forward Chaining</i> )	86
6.9.2 Raciocínio para trás ( <i>Backward Chaining</i> )	87
<b>6.10 Escolhendo entre raciocínio para frente ou raciocínio para trás</b>	<b>88</b>
<b>6.11 Determinando quando regras devem ser utilizados</b>	<b>88</b>
<b>6.12 Considerações importantes na escrita de regras em KAPPA</b>	<b>89</b>
<b>6.13 A agenda</b>	<b>90</b>

6.14 Resolução de conflitos no raciocínio para frente	90
6.14.1 Avaliação seletiva	90
6.14.2 Avaliação em profundidade ( <i>Depth First</i> )	91
6.14.3 Avaliação em amplitude ( <i>Breadth First</i> )	91
6.14.4 Avaliação <i>Best First</i>	91
6.15 Ferramentas para desenvolver sistemas	92
6.16 Utilizando KAL, a Linguagem de aplicação de KAPPA-PC	96
6.16.1 Sintaxe do KAL	97
5.17 Utilizando dados de origem externa	98
<b>CAPÍTULO VII</b>	<b>99</b>
<b>APLICAÇÃO DE UM SISTEMA ESPECIALISTA DIFUSO PARA O DIAGNÓSTICO DE HEPATOPATIAS CRÔNICAS</b>	<b>99</b>
7.1 Introdução	99
7.2 Descrição do Problema	100
7.3 Solução Proposta	101
7.3.1 Função de Fuzzificação	103
7.3.2 Definição dos Conjuntos Difusos	104
7.3.3 Base de Conhecimento	105
7.3.4. Interface com o Usuário	107
7.4 Resultados Obtidos	109
<b>CAPÍTULO VIII</b>	<b>110</b>
<b>CONCLUSÕES E RECOMENDAÇÕES</b>	<b>110</b>
8.1 Conclusões	110
8.2 Recomendações	111
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>112</b>
<b>ANEXO I</b>	<b>119</b>

## LISTA DE FIGURAS

<b>Figura 2.1</b>	<b>Alguns dos domínios de tarefas em IA.</b>	<b>6</b>
<b>Figura 3.1.</b>	<b>Grau de pertinência para alto</b>	<b>14</b>
<b>Figura 3.2.</b>	<b>Grau de pertinência para velho</b>	<b>14</b>
<b>Figura 3.3.</b>	<b>Conjunto difuso <math>A</math> do conjunto <math>x</math>.</b>	<b>16</b>
<b>Figura 3.4.</b>	<b>Núcleo, suporte e limites de um conjunto difuso.</b>	<b>17</b>
<b>Figura 3.5.</b>	<b>Conjuntos difusos normalizado e não normalizado.</b>	<b>18</b>
<b>Figura 3.6.</b>	<b>Conjuntos difusos convexos e não convexos</b>	<b>19</b>
<b>Figura 3.7.</b>	<b>Interseção de dois conjuntos difusos convexos.</b>	<b>19</b>
<b>Figura 3.8.</b>	<b>Representação gráfica das operações básicas com conjuntos difusos</b>	<b>21</b>
<b>Figura 3.9.</b>	<b>Representação gráfica das leis de exclusão media</b>	<b>23</b>
<b>Figura 3.10.</b>	<b><math>\alpha</math>-Corte fraco.</b>	<b>30</b>
<b>Figura 3.11.</b>	<b>Matriz difusa representando uma relação difusa <math>R</math>.</b>	<b>32</b>
<b>Figura 4.1.</b>	<b>Caraterísticas gerais de um sistema especialista.</b>	<b>44</b>
<b>Figura 4.2.</b>	<b>Participantes na construção de um sistema especialista.</b>	<b>45</b>
<b>Figura 4.3.</b>	<b>Estrutura de um sistema especialista.</b>	<b>48</b>

<b>Figura 4.4. Processo típico de aquisição de conhecimento para a construção de SE.</b>	<b>52</b>
<b>Figura 5.1. Sistema de inferência difuso.</b>	<b>57</b>
<b>Figura 5.2. Exemplo de conjuntos difusos definidos na base de dados.</b>	<b>58</b>
<b>Figura 5.3. Fuzzificação da temperatura de entrada 40°C.</b>	<b>59</b>
<b>Figura 5.5. Forma canônica de um conjunto de regras.</b>	<b>62</b>
<b>Figura 5.4. Tipos comuns de regras difusas na forma <i>if-then</i> e mecanismos difusos de raciocínio.</b>	<b>63</b>
<b>Figura 6.1. Exemplo de <i>slots</i> contidos na instância carrodemaria.</b>	<b>82</b>
<b>Figura 6.2. Representação de uma hierarquia, pelo <i>object browser</i> do kappa, na qual estão representadas as classes e instâncias.</b>	<b>83</b>
<b>Figura 6.3. Janela principal do KAPPA-PC.</b>	<b>92</b>
<b>Figura 6.4. Ferramentas de edição do KAPPA-PC.</b>	<b>93</b>
<b>Figura 7.1. Representação da função de pertinência e seus quatro parâmetros.</b>	<b>103</b>
<b>Figura 7.2. Conjuntos difusos definidos para a variável lingüística TGO.</b>	<b>105</b>
<b>Figura 7.3. Função de pertinência do valor de certeza lingüístico.</b>	<b>106</b>
<b>Figura 7.4. Tela de Anamnese para entrada de dados no sistema ARANDÚ.</b>	<b>108</b>

## LISTA DE TABELAS

<b>Tabela 3.1 Levantamento das utilizações dos conjuntos difusos nas investigações.</b>	<b>13</b>
<b>Tabela 3.2 Propriedades básicas das relações difusas.</b>	<b>36</b>
<b>Tabela 4.1 Vantagens de um SE frente a um especialista humano.</b>	<b>47</b>
<b>Tabela 4.2 Desvantagens de um SE frente a um especialista humano.</b>	<b>47</b>
<b>Tabela 4.3 Algumas das técnicas de aquisição do conhecimento.</b>	<b>53</b>
<b>Tabela 4.4 Exemplo de SE aplicados na área médica.</b>	<b>54</b>
<b>Tabela 5.1. Exemplo de SE da área médica que utilizam lógica difusa.</b>	<b>77</b>
<b>Tabela 7.1 Valores de certeza lingüísticos utilizados e os intervalos dos mesmos.</b>	<b>106</b>



# CAPITULO I

## INTRODUÇÃO

### 1.1 Introdução

Desde a sua formalização por Lotfi Zadeh em 1965, a teoria dos conjuntos difusos tem chamado cada vez mais a atenção dos investigadores. Isto se deve a que a teoria provê uma estrutura matemática com a qual se pode estudar e representar conceitos e fenômenos vagos.

A excelência dos conjuntos difusos está justamente na habilidade de expressar a ambigüidade e subjetividade do pensamento humano. A lógica difusa pode-se dizer é resultado de dois processos de fuzificação, primeiro a introdução de predicados vagos na linguagem objeto, e segundo, o tratamento dos predicados metalinguísticos verdadeiro e falso como vago ou difuso [TUR68].

Segundo mesmo Zadeh, as razões principais para adotar a lógica difusa são, em primeiro, que ela evita as complexidades introduzidas pelo regimento dos argumentos informais, e segundo devido a que ela é uma maneira de reconhecer que os termos verdadeiro e falso não são precisos mas sim difusos [ZAD74].

A aplicação da lógica difusa nos sistemas especialistas deve-se à necessidade de fazer inferências com dados vagos, ambíguos, inexatos e incompletos. Ela também ajuda na representação dos conhecimentos e conceitos do especialista, permitindo modelar e analisar os processos do raciocínio de uma forma mais clara.

Os sistemas especialistas difusos têm-se destacado nestes últimos anos pela sua aplicabilidade surgindo primeiramente na área de controle, para logo se difundir em muitos

outras áreas. Na área de diagnóstico médico a sua aplicação tem chamado cada vez mais a atenção devido a que as informações nela são vagas e imprecisas.

## **1.2 Objetivos**

### **1.2.1 Objetivos Gerais**

Estudar a aplicação da lógica difusa nos sistemas especialistas e a sua aplicabilidade no diagnóstico médico.

### **1.2.2 Objetivos Específicos**

- ◆ Revisar os conceitos básicos inerentes ao assunto.
- ◆ Descrever e delimitar o problema a ser estudado.
- ◆ Desenvolver um protótipo de sistema especialista difuso na área médica para o diagnóstico de hepatopatias crônicas.

## **1.3 Estrutura do trabalho**

Este trabalho está estruturado em oito capítulos. No primeiro são apresentados os objetivos gerais e específicos deste trabalho, bem como sua estrutura.

No segundo capítulo é feita a introdução à Inteligência Artificial (IA), os domínios de tarefas da IA e suposições subjacentes sobre inteligência.

No terceiro capítulo é feito um estudo dos fundamentos da teoria difusa, englobando aspectos como: quantificação da ambigüidade, operações com conjuntos difusos, propriedades deles, classes de operadores, relações difusas, etc..

No quarto capítulo estuda-se os sistemas especialistas apresentando-se suas características, os participantes na sua construção, as suas vantagens e desvantagens, etapas na sua construção e dá-se exemplo de sistemas especialistas aplicados na área médica.

No quinto capítulo fala-se dos sistemas especialistas difusos, do seu sistema de inferência, da expressão das regras em forma canônica, dos modelos de sistemas especialistas difusos baseados em regras, da aplicação dos sistemas especialistas difusos no diagnóstico médico e do tratamento da ambigüidade no mesmo.

No sexto capítulo apresenta KAPPA-PC 2.3, faz-se um estudo desta *Shell* para o desenvolvimento de sistemas especialistas, fala-se das tecnologias que ela integra, da programação orientada a objetos, as suas características e vantagens, dos tipos de representação dos processos, se dá também considerações importantes a ter em conta na hora de utilização das regras, dos tipos de avaliação dos conflitos e as ferramentas do desenvolvedor de sistemas com KAPPA.

No sétimo capítulo é feita uma aplicação prática, envolvendo o estudo do problema do diagnóstico de hepatopatias crônicas. No capítulo é descrito o problema, a proposta de solução adotada e os resultados obtidos.

Finalmente, o oitavo capítulo apresenta as conclusões e recomendações deste trabalho.

# CAPÍTULO II

## INTELIGÊNCIA ARTIFICIAL

### 2 INTELIGÊNCIA ARTIFICIAL

#### 2.1 DEFINIÇÃO E HISTÓRIA

Inteligência Artificial (IA) é a parte da computação que estuda o projeto de sistemas computacionais inteligentes [BAR81], isto é, com características que associamos como inteligentes no comportamento humano (compreensão de linguagem, aprendizado, raciocínio, resolução de problemas, etc.).

A área de Inteligência Artificial (IA), como um ramo separado da ciência da computação, foi criada em 1956 por John MacCarthy [RIC93]. Acreditava-se que máquinas inteligentes como o humano poderiam ser construídas num futuro muito próximo. Após este entusiasmo inicial e à medida que as investigações progrediram, percebeu-se que essa idéia era só um sonho, então, entrou-se num período em que o interesse na área e as investigações decaíram muito.

Os trabalhos iniciais focalizavam tarefas formais, tais como jogos e demonstração de teoremas. Acreditava-se que a generalização das investigações nesta área seria a chave para posteriores resoluções de problemas na área de IA, o que tem acontecido de certa forma, mas não na expectativa que se esperava [TER94].

Samuel elaborou um jogo de damas que, ao longo de uma partida com seu oponente, usava a experiência adquirida para aperfeiçoar seu desempenho em jogos futuros, Newell e Simon desenvolveram um programa demonstrador de Teoremas (*Logic Theorist*), que

conseguia demonstrar vários teoremas do Livro (*Principia Mathematica*) de Whitehead e Russell. Estas duas áreas inicialmente exploradas têm como característica comuns que se considera inteligentes as pessoas que as resolvem bem [BUC84a].

Outra incursão pioneira da IA foi a solução de problemas de *raciocínio de senso comum*, que são problemas do tipo que enfrentamos todos os dias, incluindo o raciocínio sobre objetos físicos e seus relacionamentos uns com os outros, e também o raciocínio sobre ações e conseqüências. Neste âmbito Newell, Shaw e Simon criaram o Resolvedor Geral de Problemas GPS (*General Problem Solver*) que era um programa que resolvia teoremas matemáticos mas não da forma com que as pessoas geralmente fazem, porém utilizava um modelo matemático.

Com o avanço das pesquisas em IA, foi-se abrangendo novas tarefas, como percepção, compreensão de linguagem natural e solução de problemas especializados, como diagnóstico médico, análise química e engenharia. A figura 2.1 mostra algumas tarefas que são alvo da IA.

O avanço em IA segue, mais ou menos, uma ordem natural, pois uma pessoa desenvolve primeiro as habilidades básicas de percepção, lingüística, senso comum e por último as tarefas mais especializadas como engenharia, medicina ou finanças. Isto, é devido ao fato de que inicialmente se pensava que era mais fácil a duplicação computadorizada das habilidades básicas, mas logo ficou demonstrado que lidar com conhecimento especializado requer conhecimentos menos complexos que as habilidades comuns e é mais fácil lidar com eles no computador.

Como conseqüência, a área de maior sucesso da IA é justamente a representação dos conhecimentos especializados nos sistemas chamados **sistemas especialistas**, existindo atualmente milhares deles nas mais diversas áreas de aplicação como matemática, engenharia, direito, medicina, química, geologia, biologia, administração etc.

### TAREFAS COMUNS

- ⇒ **Percepção**
  - ◆ **Visão**
  - ◆ **Fala**
  
- ⇒ **Linguagem natural**
  - ◆ **Compreensão**
  - ◆ **Geração**
  - ◆ **Tradução**
  
- ⇒ **Raciocínio do senso comum**
  - ◆ **Objetos físicos e seu relacionamento**
  - ◆ **Ações e sua conseqüências**
  
- ⇒ **Controle de robôs**

### TAREFAS FORMAIS

- ⇒ **Jogos**
  - ◆ **Xadrez**
  - ◆ **Damas**
  - ◆ **etc.**
  
- ⇒ **Matemática**
  - ◆ **Geometria**
  - ◆ **Lógica**
  - ◆ **Cálculo integral**

### TAREFAS DE ESPECIALISTAS

- ⇒ **Engenharia**
- ⇒ **Análise científica**
- ⇒ **Diagnóstico médico**
- ⇒ **Análise financeira**

**Figura 2.1** Alguns dos domínios de tarefas em IA [RIC93].

## 2.2 Questões importantes na Inteligência Artificial

### 2.2.1 Suposição subjacente sobre inteligência

No coração da IA está o que é chamado de **hipótese do sistema de símbolos físicos**. Define-se um sistema de símbolos físicos da seguinte maneira:

“Um sistema de símbolos físicos consiste em um grupo de entidades, chamadas símbolos, que são padrões físicos que podem ocorrer como componentes de um outro tipo de entidade chamada expressão. Assim, uma estrutura de símbolos é composta de um número de instâncias de símbolos relacionados de alguma forma física. Num determinado momento, o sistema conterá uma coleção destas estruturas de símbolos. Além dessas estruturas, o sistema contém uma coleção de processos que operam nas expressões para produzir outras expressões: processos de criação, modificação, reprodução e destruição. Um sistema de símbolos físicos é uma máquina que produz ao longo do tempo uma coleção progressiva de estruturas de símbolos. Tal sistema existe em um mundo de objetos mais amplo do que apenas estas próprias estruturas simbólicas” [ NEW 76 ].

A única maneira de determinar a veracidade desta hipótese é através da experimentação, e isto é favorecido pela facilidade cada vez maior de construir computadores. Nestas investigações é escolhida uma determinada tarefa que se considere que requiera inteligência para ser realizada, propõe-se um programa para executar a tarefa e este, em seguida, é testado.

Muitas vezes esta hipótese do sistema de símbolos conseguem provar que são capazes de modelar alguns aspectos da inteligência, mas não outros. A importância desta hipótese está no fato de que ela é uma teoria significativa sobre a natureza da inteligência humana, além de ser a base da crença de que é possível criar programas que executem tarefas inteligentes.

### 2.2.2 Técnica de Inteligência Artificial

A técnica de IA explora o conhecimento, que deve ser representado de tal forma que:

- ◆ O conhecimento captura generalizações;
- ◆ Ele precisa ser compreendido pelas pessoas que o fornecem;
- ◆ Ele pode ser facilmente modificado para corrigir erros e refletir mudanças do mundo e da nossa visão do mundo;
- ◆ Ele pode ser usado em inúmeras situações, mesmo que não seja totalmente preciso nem esteja completo;
- ◆ Ele pode ser usado para ajudar a superar seu próprio volume [RIC93].

### 2.2.3 O nível do modelo

Antes de começar qualquer coisa, o primeiro a ser feito é decidir exatamente o que se está tentando fazer. Decidir se está tentando produzir programas que fazem as coisas inteligentes que as pessoas fazem, ou está tentando produzir programas que simplesmente executem as tarefas do modo que lhes pareça mais fácil.

Os programas que executam tarefas da mesma maneira que as pessoas podem ser divididos em duas classes.

Os programas de primeira classe tentam solucionar problemas que não se enquadram realmente na definição do que seja uma tarefa de IA, isto é, problemas facilmente solucionáveis por computador, embora a solução fácil explore mecanismos que não parecem estar disponíveis às pessoas. Um exemplo clássico deste tipo de programas é o Perceptor e



Memorizador Elementar EPAM (*Elementary Perceiver and Memorizer*) (Feigenbaum, 1963), que memorizava pares associados de sílabas sem nenhum sentido, tarefa esta que é difícil para as pessoas.

A segunda classe de programas tenta modelar o desempenho humano em tarefas que não são triviais para o computador, tais como:

- ◆ Testar teorias psicológicas sobre o desempenho humano;
- ◆ Permitir que os computadores compreendam o raciocínio humano;
- ◆ Permitir que as pessoas compreendam o processo de inferência realizado pelo computador;
- ◆ Explorar conhecimentos que podemos obter das pessoas.

#### 2.2.4 Critérios de sucesso

Em 1950 Turing propôs um método para determinar se uma máquina consegue ou não pensar necessitando, para tanto, de duas pessoas além da máquina a ser avaliada. Uma das pessoas é o interrogador que está em uma sala separada dos outros membros do teste e que faz perguntas tanto à pessoa, quanto ao computador e seu objetivo é determinar quem é a pessoa e quem é a máquina. Ainda levará muito tempo para que um computador consiga passar no teste de Turing, mas é possível medir o êxito de um programa em IA num domínio mais restrito, por exemplo, um computador pode adquirir classificação no *ranking* do xadrez do mesmo modo que um jogador humano.

Nos domínios técnicos, é possível comparar quanto tempo leva para um programa completar uma tarefa em comparação com uma pessoa. Como exemplo, na configuração de sistemas de computação de acordo com as necessidades do cliente, os computadores precisam minutos para realizar tarefas que anteriormente exigiam horas de trabalho [BAR81].

Em forma geral pode-se falar que é possível construir um programa que satisfaça algum padrão de desempenho para uma determinada tarefa. Isto não significa que o programa execute a tarefa da melhor maneira possível. Então, quando se parte para projetar programas de IA, deve-se tentar especificar o melhor possível os critérios que determinarão o sucesso daquele programa, em particular, a operação no seu domínio restrito.

## CAPITULO III

### LÓGICA DIFUSA

#### 3 Lógica Difusa

##### 3.1 Introdução

Após a formalização da lógica difusa por L.A. Zadeh, na metade da década de 60, poucos especialistas no mundo, nos dez anos seguintes, deram atenção à teoria dos conjuntos difusos lançada por ele, mas atualmente tem crescido rapidamente esse interesse, evidenciado pela quantidade de trabalhos, livros e artigos publicados. Para que uma idéia cresça, pode-se dizer que são necessárias três condições [TER94]: necessidade social, metodologia (ambos ideais e técnicas) e atratividade para as investigações. Analisaremos, abaixo, a satisfação dessas condições em relação à lógica difusa:

1. Necessidade social: As tecnologias avançadas estão cada vez mais intimadas a estar conectadas com as pessoas e que compreendam sua linguagem, mas a maior dificuldade existente é a grande ambigüidade na linguagem do dia a dia. Uma nova lógica que expressasse dita ambigüidade era necessária e a lógica difusa pode ser considerada a mais apropriada para isso.
2. Nova metodologia: A lógica difusa, que inicialmente foi pensada como uma técnica para expressar matematicamente as ambigüidades lingüísticas, agora é uma metodologia que está sendo aplicada numa ampla variedade de fenômenos ambíguos.

3. Atrativo para as investigações: Ela é muito interessante tanto do lado teórico como prático. Como sistema matemático, os conjuntos difusos expandem as estruturas atuais e introduzem novos conceitos que despertam o interesse de muitos investigadores na área teórica. Na parte prática as aplicações em sistemas de controle são muito difundidas atualmente.

Os campos de aplicação da Lógica Difusa (LD) são dos mais amplos, mas que podem ser classificados em três tipos de sistemas: máquina, homem e homem-máquina.

Máquina: o principal problema num sistema máquina é de dotá-la de um alto nível de inteligência como, por exemplo, nos robôs e máquinas automáticas. Atualmente inteligência é expressa em termos de proposições num mecanismo de conhecimento onde o reconhecimento, julgamento, avaliação e inferência estejam sendo imitados, mas a inteligência humana inclui pensamentos ilógicos como a intuição e a inspiração, sendo virtualmente impossível de expressá-los numa lógica bivalorada.

Sistemas Humanos: Contrastando com os sistemas baseados em máquina, estes tentam introduzir metodologia científica em complicados e ambíguos problemas do homem e da sociedade. As tecnologia de informação especialmente estão conectadas com atividades psicológicas, e têm uma grande influência. Nesta caso as investigações no lado humano do pensamento são importantes, mas até agora os modelos desenvolvidos são muito abstratos e utilizam modelos matemáticos baseados em afirmações inusuais.

Sistemas Homem-máquina: Este sistema utiliza em forma interativa o homem e a máquina para trabalhar juntos em tarefas que não pode ser feito por cada um deles em forma independente. Neste tipo de sistema a comunicação entre o homem e a máquina é muito importante, sendo o problema mais importante a simplificação do processo de entrada de dados do homem para a máquina. Dito de outra forma, as máquinas deveriam ser capazes de entender a linguagem natural e informações visuais. Outro problema também é a comunicação das máquinas para as pessoas e que ela forneça de um modo a que o mesmo possa fazer um julgamento sintético baseado nestas informações.

Na Tabela 3.1 pode-se observar as pesquisas levantadas em 1985, de acordo com IFSA (*International Fuzzy Systems Association*).

<b>SISTEMAS MAQUINAS</b>	<b>SISTEMAS HUMANOS</b>	<b>SISTEMAS HOMEM/MAQUINA</b>
Reconhecimento de voz/imagem.	Modelos de confiabilidade humana.	Diagnóstico Médico
Reconhecimento de caracteres chineses.	Modelos cognitivos psicológicos de pensamento/comportamento.	Inspeção de processamento de dados.
Entendimento de linguagem natural	Investigações sensores.	Sistemas Especialistas
Robots inteligentes	Análises de conhecimentos públicos.	CAI
Reconhecimento de plantios	Avaliação de riscos.	CAD
Controle de Processos	Avaliação ambiental.	Otimização de planificação
Gerenciamento de Produção	Estruturas de relacionamento humanas.	Desenvolvimento de Projetos
Operação de carros / trens	Modelos de Seleção de Mercados.	Diagnóstico de equipamentos.
Sistemas de mantimento / seguridade	Análises de Categorias.	Avaliação de qualidade
Diagnóstico de danos	Psicologia social.	Sistemas de seguros
Controladores Difusos		Interfaces humanas
Controle de sistemas elétricos nas casas		Gerenciamento de tomada de decisões.
		Base de Conhecimentos.
		Base de Dados

Tabela 3.1. Levantamento das utilizações dos conjuntos difusos nas pesquisas. Fonte IFSA 1985 Japão.

### 3.2 Quantificação da ambigüidade

A linguagem natural está cheia de ambigüidades e, por tanto, múltiplas interpretações, os objetos dos adjetivos especialmente não são claros e específicos, sendo ambíguos numa ampla forma. Por Exemplo, quando se fala de “pessoa alta”, não se pode determinar com precisão quem é alta ou quem não é alta. Da mesma forma a ambigüidade de “pessoa velha” está no adjetivo “velha”.

Considere-se agora a ambigüidade do significado de “alto” e “velho” em forma numérica. Se considerarmos um intervalo de alturas de 140 cm a 200 cm, o grau pelo qual um indivíduo com uma altura determinada  $x$  pudesse ser considerado “alto” é  $\mu$ , isto é, se faz que a altura  $x$  corresponda a um grau  $\mu$  ( $0 \leq \mu \leq 1$ ). Esse valor  $\mu$  é denominado grau de pertinência do indivíduo ao conjunto “alto”. Representando os valores de  $x$  na horizontal e os valores de  $\mu$  na vertical tem-se um gráfico como da Figura 3.1. Ele representa a ambigüidade do termo “alto” em forma quantificada. Da mesma forma podemos representar a ambigüidade do termo “velho” colocando no eixo horizontal os anos e no eixo vertical o grau pelo qual dita pessoa pode ser qualificada de “velha” figura 3.2.

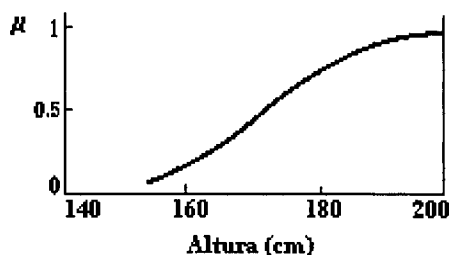


Figura 3.1. Grau de pertinência a alto.

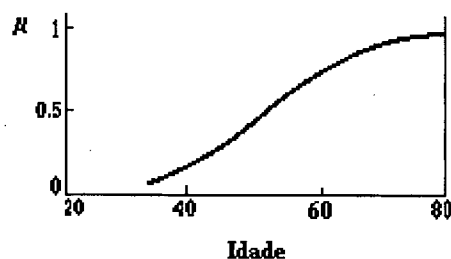


Figura 3.2. Grau de pertinência a velho.

Nos grafos anteriores nos eixos horizontais quantifica-se as palavras (idade e altura) num espaço unidimensional, e no eixo vertical quantifica-se o grau de ambigüidade. Este tipo de representação de uma palavra é denominado “**quantificação do significado**”, e o gráfico

geralmente é denominado “**significado quantificado**”. O significado da palavra é quantificado num intervalo específico no caso da idade 20 a 80 anos e altura de 140 a 200cm.

### 3.3 Conjuntos difusos

#### 3.3.1 Definição

Se  $X$  é uma coleção de objetos representados genericamente por  $x$ , então o conjunto difuso  $A$  em  $X$  é um conjunto de pares ordenados da seguinte forma:

$$\underline{A} = \{ (x, \mu_{\underline{A}}(x)) \mid x \in X \}$$

$\mu_{\underline{A}}(x)$  é chamada a função de pertinência, grau de pertinência ou grau de verdade de  $x$  pertencer a um conjunto difuso  $\underline{A}$ . Exemplo: tem-se o conjunto altura  $X = [140\text{cm}, 200\text{cm}]$ , representa-se o conjunto  $\underline{A} =$  “pessoas altas”. Sendo  $x$  a altura de uma determinada pessoa a representação é:

$$\underline{A} = \{ (140, 0.0), (160, 0.2), (180, 0.8), (200, 1.0) \}$$

Nota-se que os valores com grau de pertinência zero não são representados no conjunto. A interpretação seria de que se uma pessoa tem uma altura de  $x = 160$  cm ele pertence ao conjunto  $\underline{A} =$  “pessoas altas” com um grau de pertinência  $\mu_{\underline{A}}(160) = 0.2$ , já uma pessoa com 180 cm de altura têm um grau de pertinência ao conjunto de 0.8 e por último uma pessoa de 200 cm pode-se dizer que a pertinência do mesmo ao conjunto alto é total e isto é representado pelo valor de pertinência 1.

Uma representação abstrata de um conjunto difuso  $\underline{A}$  do conjunto universo  $X$  é feito na Figura 3.3. A estrutura retangular representa o conjunto universo  $X$ , o círculo ponteados são

limites ambíguos, do que esta dentro ou fora do conjunto, e  $\underline{A}$  (o “~” embaixo da letra A indica que A é um conjunto difuso, mas muitas vezes ele é eliminado e fica simplesmente a letra A) que é um subconjunto de X.

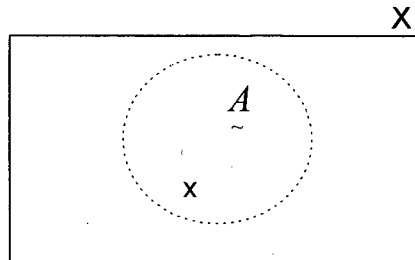


Figura 3.3. Conjunto difuso  $\underline{A}$  do conjunto X [TER94].

A função que atribui o grau com o qual o elemento  $x$  está incluído no subconjunto difuso  $\underline{A}$  é chamada de **função de pertinência  $\mu$** , atribuindo-lhe um valor de zero a um  $[0,1]$ .

A maneira de representar o valor de pertinência de um elemento  $x$  do universo,

$$\mu_{\underline{A}}(x) = 1 \text{ ou } \mu_{\underline{A}}(x) = 0.8 \text{ ou } \mu_{\underline{A}}(x) = 0.3 \text{ ou } \mu_{\underline{A}}(x) = 0$$

Onde:  $\mu_{\underline{A}}$  = função de pertinência de  $\underline{A}$ .

$\mu_{\underline{A}}(x)$  deve ser interpretada como o grau de pertinência do elemento  $x$  do universo X ao conjunto difuso  $\underline{A}$ .

A função  $\mu : \rightarrow [0, 1]$  está dando o rótulo  $\underline{A}$  e  $\underline{A}$  é chamado de conjunto difuso de X.  $\mu$  é chamado da função de pertinência de  $\underline{A}$ .



Pela definição anterior, pode-se interpretar a figura 3.1 como representando a função de pertinência do conjunto difuso “grupo de valores de alturas que pode-se considerar altas” dentro de um conjunto X de 140 cm a 200 cm. E a figura 3.2 como a função de pertinência do conjunto difuso “grupo de idades que pode-se considerar velhas” na faixa de 20 a 80 anos. Mas como estas denominações são muito longas, então geralmente se abrevia em simplesmente falando conjunto difuso “alto” ou “velho”.

### 3.3.2 Características de uma função de pertinência

Considerando que a informação sobre um conjunto difuso é expressa por sua função de pertinência, é importante familiarizar-se com a terminologia que descreve um conjunto difuso. Na figura 3.4 representa-se graficamente os significados da terminologia utilizada.

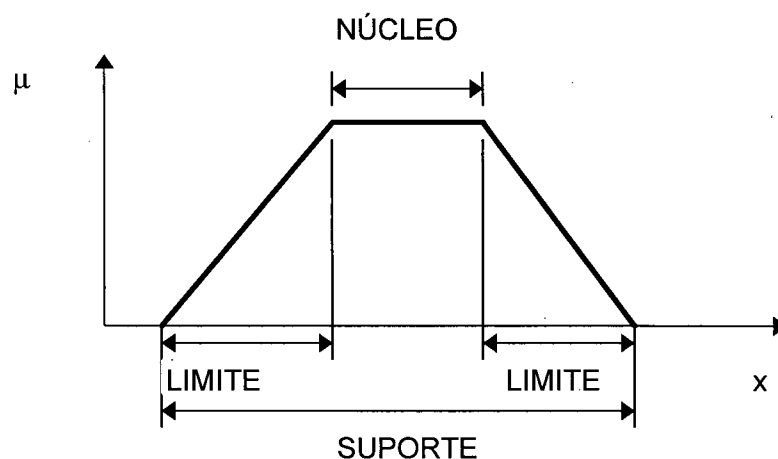


Figura 3.4. Núcleo, suporte e limites de um conjunto difuso [JAM93].

O núcleo de uma função de pertinência (FP), para algum conjunto difuso  $A$  é definido como a região do universo de discurso que é caracterizada por ter uma pertinência total ao conjunto,  $\mu_A(x) = 1$ .

O suporte de uma FP, para algum conjunto  $\underline{A}$  é aquela região do universo de discurso que está caracterizada por não ter uma pertinência zero,  $\mu_{\underline{A}}(x) \neq 0$ .

Os limites de uma FP para algum conjunto difuso  $\underline{A}$  são aquelas regiões do universo que contêm elementos cujos graus de pertinência não são zero mas também não têm um grau de pertinência total,  $0 < \mu_{\underline{A}}(x) < 1$  [JAM93].

Um **conjunto difuso normalizado** é aquele que tem pelo menos um elemento  $x$  no universo  $X$  cujo valor de pertinência seja um. Na figura 3.5 mostra-se um conjunto difuso normalizado e outro não normalizado.

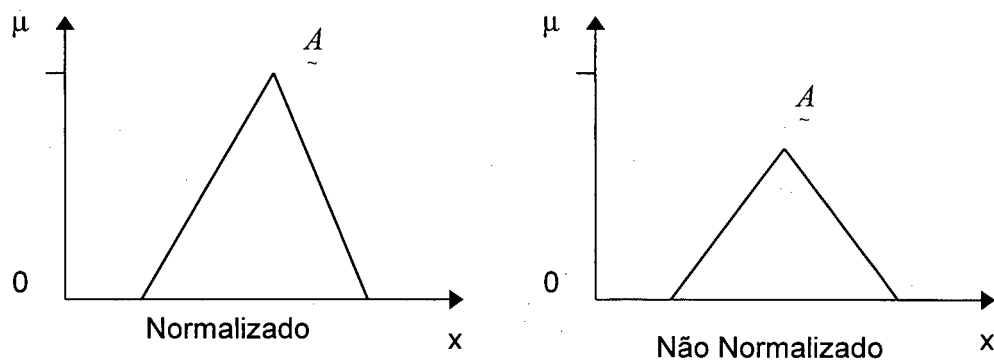


Figura 3.5. Conjunto difuso normalizado e não normalizado.

Conjunto Difuso convexo, é aquele que descreve uma função de pertinência cujos valores de pertinência estritamente aumentam ou diminuem monotonicamente, quando se incrementa os valores para os elementos no universo. Dito de outra maneira, se para todos os elementos de um conjunto difuso contínuo  $\underline{A}$  onde,  $x < y < z$  e  $\mu_{\underline{A}}(y) \geq \min[\mu_{\underline{A}}(x), \mu_{\underline{A}}(z)]$  então fala-se que  $\underline{A}$  é um conjunto difuso convexo. Na figura 3.6 pode-se observar um conjunto difuso convexo e outro não convexo [ZIM90].

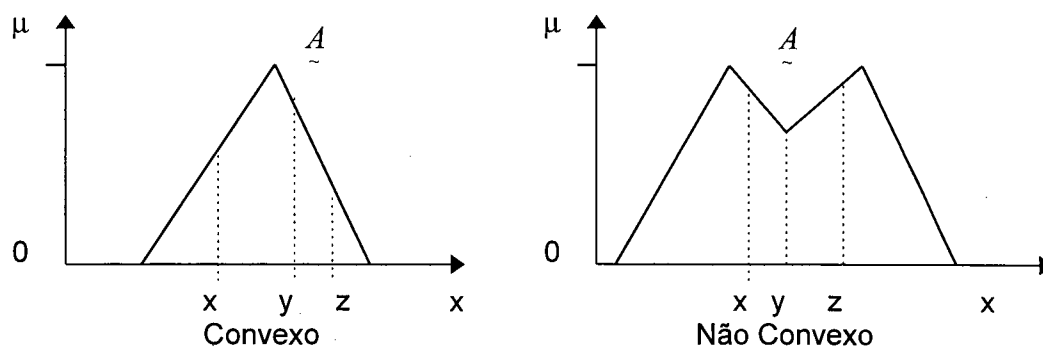


Figura 3.6. Conjuntos difusos convexas e não convexas [JAM93].

Deve-se ter em conta que uma das características especiais dos conjuntos difusos convexas  $\tilde{A}$  e  $\tilde{B}$  é que a interseção dos conjuntos difusos convexas sempre é também convexa. Na figura 3.7 mostra-se a interseção de dois conjuntos convexas.

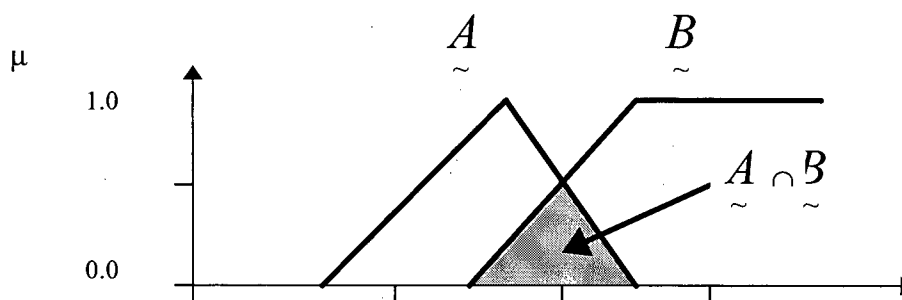


Figura 3.7. Interseção de dos conjuntos difusos convexas [TER94].

### 3.4 Operações com conjuntos difusos

#### 3.4.1 Operações básicas com conjuntos difusos

Seguem-se as operações com conjuntos difusos, como os conjuntos difusos estão definidos pela sua função de pertinência, então as operações com conjuntos difusos devem ser feitas com sua função de pertinência .

### 3.4.1.1 União

A união dos conjuntos difusos  $\underline{A}$  e  $\underline{B}$  representado por  $\underline{A} \cup \underline{B}$  é o conjunto difuso definido pela seguinte função de pertinência [ZIM90]:

$$\mu_{\underline{A} \cup \underline{B}}(x) = \mu_{\underline{A}}(x) \vee \mu_{\underline{B}}(x) \quad (3.1)$$

$$\mu_{\underline{A} \cup \underline{B}}(x) = \max \{ \mu_{\underline{A}}(x), \mu_{\underline{B}}(x) \} \quad x \in X$$

### 3.4.1.2 Interseção

A interseção dos conjuntos difusos  $\underline{A}$  e  $\underline{B}$  é representado por  $\underline{A} \cap \underline{B}$ , e sua função de pertinência é:

$$\mu_{\underline{A} \cap \underline{B}}(x) = \mu_{\underline{A}}(x) \wedge \mu_{\underline{B}}(x) \quad (3.2)$$

$$\mu_{\underline{A} \cap \underline{B}}(x) = \min \{ \mu_{\underline{A}}(x), \mu_{\underline{B}}(x) \} \quad x \in X$$

### 3.4.1.3 Complemento

O complemento do conjunto difuso  $\underline{A}$  é representado por  $\overline{\underline{A}}$  sendo definido pela função de pertinência:

$$\mu_{\overline{\underline{A}}}(x) = 1 - \mu_{\underline{A}}(x) \quad x \in X \quad (3.3)$$

A representação gráfica dos operadores difusos pode-se observar na figura 3.8.

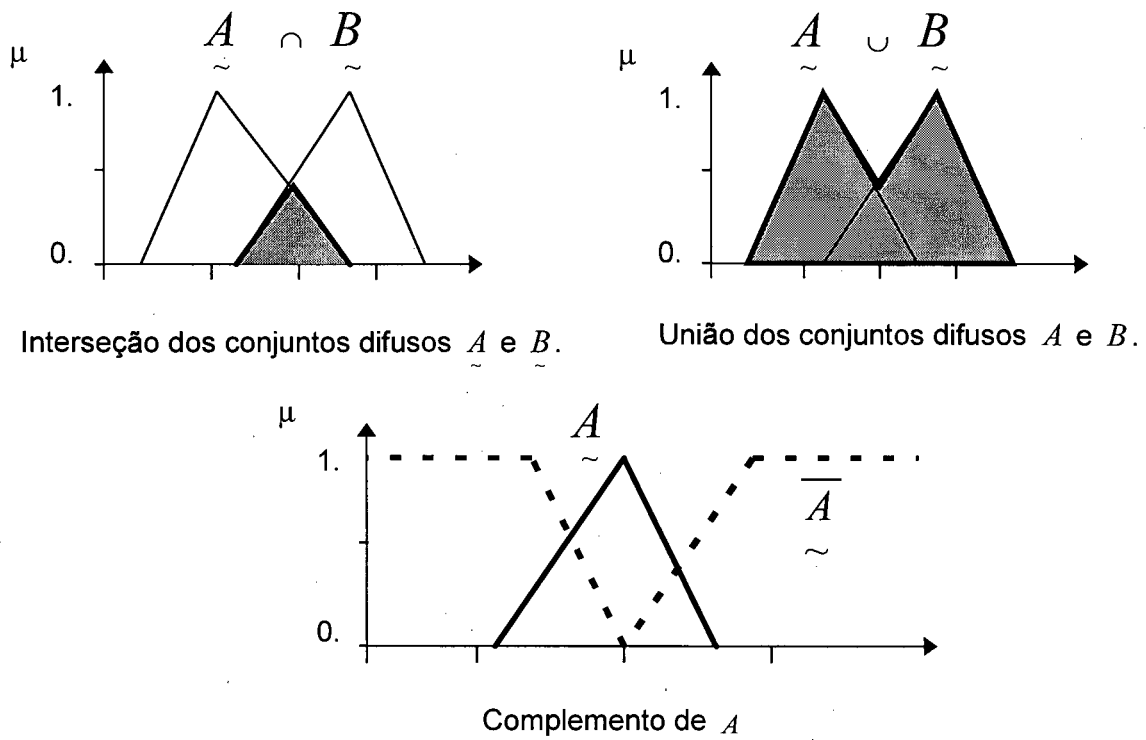


Figura 3.8. Representação gráfica das operações básicas com conjuntos difusos [JAM93].

#### 3.4.1.4 Lei De Morgan

A lei de De Morgan também se aplica aos conjuntos difusos como pode-se apreciar nas seguintes expressões:

$$\left(\overline{\tilde{A} \cap \tilde{B}}\right) = \overline{\tilde{A}} \cup \overline{\tilde{B}} \quad (3.4)$$

$$\left(\overline{\tilde{A} \cup \tilde{B}}\right) = \overline{\tilde{A}} \cap \overline{\tilde{B}} \quad (3.5)$$

Todas as propriedades para conjuntos clássicos também são aplicáveis para os conjuntos difusos, com exceção da **lei do meio excluído**, figura 3.9. Isto se deve ao fato de que os conjuntos difusos podem-se superpor, então, as lei do meio excluído para os conjuntos difusos são expressados desta forma:

$$\underline{A} \cup \overline{\underline{A}} \neq X \quad (3.6)$$

$$\underline{A} \cap \overline{\underline{A}} \neq \emptyset \quad (3.7)$$

### 3.5 Operações algébricas

#### 3.5.1 Produto Cartesiano

O Produto cartesiano do conjunto difuso é definido por: Se  $\underline{A}_1, \dots, \underline{A}_n$  são conjuntos difusos em  $X_1, \dots, X_n$ , o produto cartesiano é então um conjunto no espaço dos produtos  $X_1 \times \dots \times X_n$  com a seguinte função de pertinência:

$$\mu_{(\underline{A}_1 \times \dots \times \underline{A}_n)}(x) = \min_i \left\{ \mu_{\underline{A}_i}(x_i) \mid x = (x_1, \dots, x_n), x_i \in X_i \right\} \quad (3.8)$$

#### 3.5.2 Enésima potência

A enésima potência de um conjunto difuso  $\underline{A}$  é um conjunto difuso com a função de pertinência:

$$\mu_{\underline{R}^m}(x) = \left[ \mu_{\underline{A}}(x) \right]^m, \quad x \in X \quad (3.9)$$

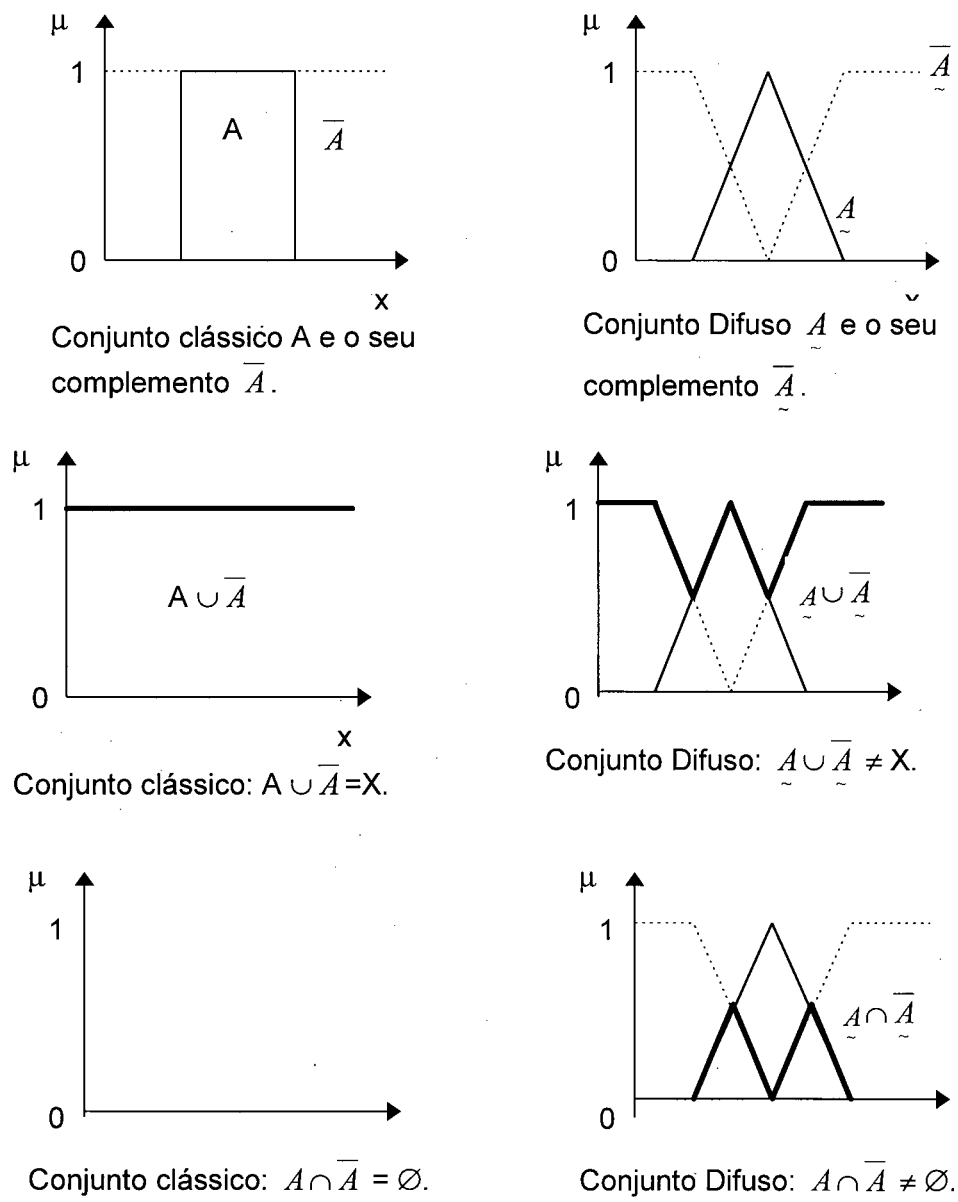


Figura 3.9. Representação gráfica da lei do meio excluído para conjuntos clássicos (esquerda) e conjuntos difusos (direita) [JAM93].

### 3.5.3 Soma algébrica

A soma algébrica  $\underline{C} = \underline{A} + \underline{B}$  está definida como:

$$\underline{C} = \left\{ \left( x, \underline{\mu}_{\underline{A} + \underline{B}}(x) \right) \mid x \in X \right\}$$

Onde,

$$\underline{\mu}_{\underline{A} + \underline{B}}(x) = \underline{\mu}_{\underline{A}}(x) + \underline{\mu}_{\underline{B}}(x) - \underline{\mu}_{\underline{A}}(x) \cdot \underline{\mu}_{\underline{B}}(x) \quad (3.10)$$

### 3.5.4 Soma limitada

A soma limitada  $\underline{C} = \underline{A} \oplus \underline{B}$  esta definida como:

$$\underline{C} = \left\{ \left( x, \underline{\mu}_{\underline{A} \oplus \underline{B}}(x) \right) \mid x \in X \right\}$$

Onde,

$$\underline{\mu}_{\underline{A} \oplus \underline{B}}(x) = \min \left\{ 1, \underline{\mu}_{\underline{A}}(x) + \underline{\mu}_{\underline{B}}(x) \right\} \quad (3.11)$$

Note-se que neste caso a pertinência da soma nunca será superior a um.

### 3.5.5 Diferença limitada

A diferença limitada  $\underline{C} = \underline{A} \ominus \underline{B}$  esta definida como:

$$\underline{C} = \left\{ \left( x, \underline{\mu}_{\underline{A} \ominus \underline{B}}(x) \right) \mid x \in X \right\}$$

Onde,

$$\underline{\mu}_{\underline{A} \ominus \underline{B}}(x) = \max \left\{ 0, \underline{\mu}_{\underline{A}}(x) + \underline{\mu}_{\underline{B}}(x) - 1 \right\} \quad (3.12)$$



Note-se que a pertinência da diferença nunca pode ser inferior a zero, ou seja, não pode tomar valores negativos.

### 3.5.6 Produto Algébrico

Produto Algébrico de dois conjuntos difusos  $\underline{C} = \underline{A} \bullet \underline{B}$  define-se como:

$$\underline{C} = \left\{ \left( x, \mu_{\underline{A}}(x) \bullet \mu_{\underline{B}}(x) \right) \mid x \in X \right\} \quad (3.13)$$

Exemplo da aplicação destes operadores: Considere os seguintes conjuntos:

$$\underline{A}(x) = \{(3, 0.5), (5, 1), (7, 0.6)\}$$

$$\underline{B}(x) = \{(3, 1), (5, 0.6)\}$$

Aplicando cada um dos operadores obtemos:

$$\underline{A} \times \underline{B} = \{[(3;3), 0.5], [(5;3), 1], [(7;3), 0.6], [(3;5), 0.5], [(5;5), 0.6], [(7;5), 0.6]\}$$

$$\underline{A}^2 = \{(3, 0.25), (5, 1), (7, 0.36)\}$$

$$\underline{A} + \underline{B} = \{(3, 1), (5, 1), (7, 0.6)\}$$

$$\underline{A} \oplus \underline{B} = \{(3, 1), (5, 1), (7, 0.6)\}$$

$$\underline{A} \ominus \underline{B} = \{(3, 0.5), (5, 0.6)\}$$

Lembre-se que, os valores que como resultado destas operações derem um grau de pertinência zero são eliminados do conjunto.

### 3.6 Classes de operadores

São duas as classes básicas de operadores [ZIM90]:

- ◆ Os operadores para interseção e união de conjuntos difusos chamados de **normas triangulares (t-norma) e conormas triangulares (t-conorma ou s-norma)**, e;
- ◆ Os operadores de termo médio que modelam conexões entre os conjuntos difusos t-norma e t-conorma.

Ambas classes tem operadores parametrizados como não parametrizados

**t-norma:** Pertencem à esta classe os seguintes operadores :

- ◆ min;
- ◆ produto, e
- ◆ soma limitada

Eles estão caracterizadas por serem funções bivaloradas de  $[0, 1] \times [0, 1]$  que satisfazem as seguintes condições:

1.  $t(0,0) = 0$ ;  $t(\mu_{\underline{A}}(x), 1) = t(1, \mu_{\underline{A}}(x)) = \mu_{\underline{A}}(x)$ ,  $x \in X$
2.  $t(\mu_{\underline{A}}(x), \mu_{\underline{B}}(x)) \leq t(\mu_{\underline{C}}(x), \mu_{\underline{D}}(x))$   
 Se  $\mu_{\underline{A}}(x) \leq \mu_{\underline{C}}(x)$  e  $\mu_{\underline{B}}(x) \leq \mu_{\underline{D}}(x)$  (monotonicidade)
3.  $t(\mu_{\underline{A}}(x), \mu_{\underline{B}}(x)) = t(\mu_{\underline{B}}(x), \mu_{\underline{A}}(x))$  (comutatividade)
4.  $t(\mu_{\underline{A}}(x), t(\mu_{\underline{B}}(x), \mu_{\underline{C}}(x))) = t(t(\mu_{\underline{A}}(x), \mu_{\underline{B}}(x)), \mu_{\underline{C}}(x))$  (associatividade)

**t-conorma (ou s-norma):** Modelam a união dos conjuntos difusos, pertencendo à ela os operadores:

- ◆ max;
- ◆ soma algébrica;
- ◆ soma limitada

É também uma função bivalorada  $s$  que mapeia de  $[0, 1] \times [0, 1]$  entre  $[0, 1]$  apresentando as seguintes propriedades:

1.  $s(1,1) = 1$ ;  $1(\underline{\mu}_A(x), 0) = t(0, \underline{\mu}_A(x)) = \underline{\mu}_A(x)$ ,  $x \in X$
2.  $s(\underline{\mu}_A(x), \underline{\mu}_B(x)) \leq s(\underline{\mu}_C(x), \underline{\mu}_D(x))$   
 Se  $\underline{\mu}_A(x) \leq \underline{\mu}_C(x)$  e  $\underline{\mu}_B(x) \leq \underline{\mu}_D(x)$  (monotonicidade)
3.  $s(\underline{\mu}_A(x), \underline{\mu}_B(x)) = s(\underline{\mu}_B(x), \underline{\mu}_A(x))$  (comutatividade)
4.  $s(\underline{\mu}_A(x), s(\underline{\mu}_B(x), \underline{\mu}_C(x))) = s(s(\underline{\mu}_A(x), \underline{\mu}_B(x)), \underline{\mu}_C(x))$  (associatividade)

### 3.7 Propriedades dos conjuntos difusos

Comutatividade:

$$\underline{A} \cup \underline{B} = \underline{B} \cup \underline{A} \quad (3.14)$$

$$\underline{A} \cap \underline{B} = \underline{B} \cap \underline{A} \quad (3.15)$$

Associatividade:

$$\underline{A} \cup (\underline{B} \cup \underline{C}) = (\underline{A} \cup \underline{B}) \cup \underline{C} \quad (3.16)$$

$$\underline{A} \cap (\underline{B} \cap \underline{C}) = (\underline{A} \cap \underline{B}) \cap \underline{C} \quad (3.17)$$

Distributividade:

$$\underline{A} \cup (\underline{B} \cap \underline{C}) = (\underline{A} \cup \underline{B}) \cap (\underline{A} \cup \underline{C}) \quad (3.18)$$

$$\underline{A} \cap (\underline{B} \cup \underline{C}) = (\underline{A} \cap \underline{B}) \cup (\underline{A} \cap \underline{C}) \quad (3.19)$$

Idempotência:

$$\underline{A} \cup \underline{A} = \underline{A} \quad (3.20)$$

$$\underline{A} \cap \underline{A} = \underline{A} \quad (3.21)$$

Identidade:

$$\underline{A} \cup \emptyset = \underline{A} \text{ e } \underline{A} \cup X = X \quad (3.22)$$

$$\underline{A} \cap X = \underline{A} \text{ e } \underline{A} \cap \emptyset = \emptyset \quad (3.23)$$

Transitividade:

$$\text{Se } \underline{A} \subseteq \underline{B} \subseteq \underline{C} \text{ então } \underline{A} \subseteq \underline{C} \quad (3.24)$$

Involução:

$$\overline{\overline{A}} = A \quad (3.25)$$

### 3.8 Tipos de conjuntos difusos

#### 3.8.1 Conjunto difuso TIPO 1

Um conjunto difuso  $\underline{A}$  do universo  $X$  é do tipo 1, se sua função de pertinência  $\underline{\mu}_A$ , é um mapeamento de  $X$  para  $[0, 1]$ .

$$\underline{A} = \left\{ \left( x, \underline{\mu}_A(x) \right) \mid x \in X, \underline{\mu}_A(x) \in [0,1] \right\}$$

#### 3.8.2 Conjunto difuso TIPO 2

É aquele conjunto difuso cujos valores de pertinência são conjuntos difusos tipo 1 para  $[0, 1]$ .

$$\underline{A} = \left\{ \left( x, \underline{\mu}_A(x) \right) \right\}$$

Onde,

$$\underline{\mu}_A(x) = \left\{ \left( v_i, \underline{\mu}_{v_i}(x) \right) \mid x \in X, v_i, \underline{\mu}_{v_i}(x) \in [0,1] \right\}$$

#### 3.8.3 Conjunto difuso TIPO m

É um conjunto difuso no universo  $X$  cujos valores de pertinência são conjuntos difusos tipo  $m-1$ ,  $m > 1$  para  $[0, 1]$ .

Desde o ponto de vista prático, os conjuntos difuso tipo  $m$  para  $m > 2$  são muito difíceis de tratar com eles ou inclusive impossível de visualizar e medir [ZIM90].

### 3.8.4 $\alpha$ -Cortes (Conjunto de nível $\alpha$ )

O conjunto  $A_\alpha$  é chamado de conjunto de nível  $\alpha$  e é definido como:

$$A_\alpha = \{x | \mu_A(x) > \alpha\}; \quad \alpha \in [0,1) \quad (3.26)$$

$$A_{\bar{\alpha}} = \{x | \mu_A(x) \geq \alpha\}; \quad \alpha \in (0,1] \quad (3.27)$$

O primeiro deles é chamado de  $\alpha$ -corte forte e o segundo de  $\alpha$ -corte fraco. A diferença entre eles é a presença ou ausência da igualdade.

Fazendo uso dos  $\alpha$ -cortes obtemos a seguinte equação que é denominada o **princípio de resolução** [TER94].

$$\mu_A(x) = \sup_{\alpha \in [0,1)} \min[\alpha, \chi_{A_\alpha}(x)] \quad (3.28)$$

$$\mu_A(x) = \sup_{\alpha \in (0,1]} \min[\alpha, \chi_{A_{\bar{\alpha}}}(x)] \quad (3.29)$$

Se o suporte do conjunto é um conjunto de números reais e a função de pertinência é contínua, o  $\alpha$ -corte fraco de um conjunto difuso convexo é um intervalo fechado como é mostrado na figura 3.10.

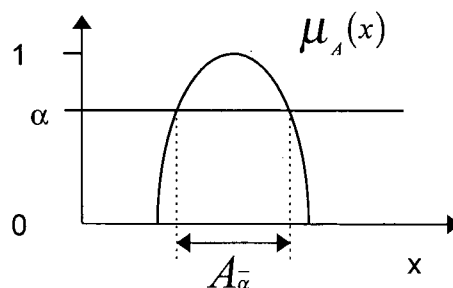


Figura 3.10.  $\alpha$ -corte fraco [TER94].

### 3.9 Relações difusas

#### 3.9.1 Introdução

Relações ambíguas como “x e y são quase iguais” ou “x e y são muito semelhantes” e “x é muito mais veloz que y” estão presentes na conversa do dia a dia, mas expressar estas relações em termos de relações ordinárias é muito difícil. Relações difusas permitem expressar este tipo de relações ambíguas.

Pode-se pensar nas Relações difusas como uma extensão das relações ordinárias e sua aplicação é muito ampla. Podem ser aplicadas em *clustering*, reconhecimento de padrões, sistemas de inferência e de controle. Podem, também, ser aplicados nas ciências chamadas “*soft sciences*” tais como psicologia, medicina, economia e sociologia.

#### 3.9.2 Relação difusa

A relação difusa R entre os conjuntos X e Y é um conjunto difuso resultante do produto  $X \times Y = \{ (x,y) \mid x \in X, y \in Y \}$  sendo caracterizada pela função de pertinência  $\mu_R$ ,

$$\mu_R : X \times Y \rightarrow [0, 1] \quad (3.30)$$

Especialmente quando  $X = Y$ , R é conhecido como uma relação difusa sobre X. Exemplo: Se x e y são pessoas, a relação como “x e y são muito parecidas” e “x é muito mais alto que y” são relações difusas. A relação difusa R é expressado da seguinte maneira utilizando a notação dos conjuntos difusos:

$$R = \int_{X \times Y} \frac{\mu_R(x, y)}{(x, y)} \quad x \in X, y \in Y \quad (3.31)$$

Generalizando, a relação n-ária difusa R em  $X_1 \times X_2 \times \dots \times X_n$  está dada por:

$$R = \int_{X_1 \times X_2 \times \dots \times X_n} \mu_R(x_1, x_2, \dots, x_n) / (x_1, x_2, \dots, x_n); \quad x_i \in X_i$$

(3.32)

e temos a seguinte função de pertinência:

$$\mu_R : X_1 \times X_2 \times \dots \times X_n \rightarrow [0, 1]$$

Outras formas de expressar uma relação difusa são matrizes e grafo.

### 3.9.3 Matrizes e grafos difusos

Dados conjuntos finitos  $X = \{x_1, x_2, \dots, x_m\}$  e  $Y = \{y_1, y_2, \dots, y_n\}$ , uma relação difusa entre  $X \times Y$  pode ser expressada por uma matriz  $m \times n$ , como mostra a figura 3.11.

Esta classe de matriz que expressa relações difusas é chamada de **matriz difusa**. Sendo que  $\mu_R$  toma valores no intervalo  $[0, 1]$ , os elementos de uma matriz difusa também tomam valores de  $[0, 1]$ .

$$R = \begin{bmatrix} \mu_R(x_1, y_1) & \mu_R(x_1, y_2) & \cdots & \mu_R(x_1, y_n) \\ \mu_R(x_2, y_1) & \mu_R(x_2, y_2) & \cdots & \mu_R(x_2, y_n) \\ \vdots & \vdots & & \vdots \\ \mu_R(x_m, y_1) & \mu_R(x_m, y_2) & \cdots & \mu_R(x_m, y_n) \end{bmatrix}$$

Figura 3.11. Matriz difusa representando uma relação difusa R.



Para representar uma relação difusa  $R$  em forma de grafo, para  $\mu_R(x_i, y_j)$ , nós fazemos  $x_i, y_j$  vértices e colocamos os grau  $\mu_R(x_i, y_j)$  na flecha que une  $x_i$  a  $y_j$ . A este tipo de grafo se chama de **grafos difusos**.

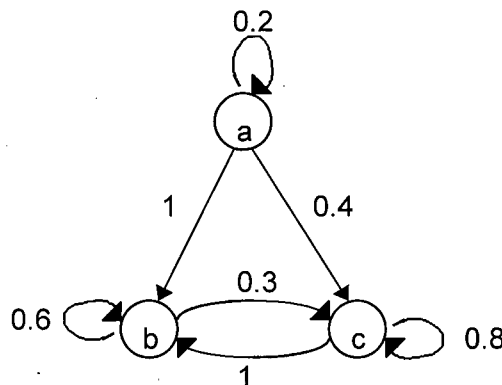
Exemplo: Sé a relação difusa  $R$  no universo  $X = \{a, b, c\}$  é:

$$R = 0.2/(a,a) + 1/(a,b) + 0.4/(a,c) + 0.6/(b,b) + 0.3/(b,c) + 1/(c,b) + 0.8/(c,c)$$

A matriz difusa resultante seria:

$$R = \begin{bmatrix} 0.2 & 1 & 0.4 \\ 0 & 0.6 & 0.3 \\ 0 & 1 & 0.8 \end{bmatrix}$$

E o grafo difuso seria o seguinte:



### 3.10 Operações com relações difusas

Se  $R$  e  $S$  são relações difusas em  $X \times Y$ , temos as seguintes operações:

$$\text{Inclusão} \quad R \subseteq S \leftrightarrow \mu_R(x, y) \leq \mu_S(x, y)$$

$$\text{União} \quad R \cup S \leftrightarrow \mu_{R \cup S}(x, y) = \max(\mu_R(x, y), \mu_S(x, y))$$

$$\text{Interseção} \quad R \cap S \leftrightarrow \mu_{R \cap S}(x, y) = \min(\mu_R(x, y), \mu_S(x, y))$$

$$\text{Conjunto complementar} \quad \bar{R} \leftrightarrow \mu_{\bar{R}}(x, y) = 1 - \mu_R(x, y)$$

### 3.11 Composição de uma relação difusa (RD)

Se  $R$  é uma RD em  $X \times Y$  e  $S$  é uma RD em  $Y \times Z$ , a composição de  $R$  e  $S$ ,  $R \circ S$ , é uma RD entre  $X \times Z$ . Existem muitos tipos de composição para as relações difusas. Consideraremos a seguir algumas delas:

**max-min composição:**

$$R \circ S \leftrightarrow \mu_{R \circ S}(x, z) = \max_y \left\{ \min(\mu_R(x, y), \mu_S(y, z)) \right\} \quad (3.33)$$

**min-max composição:**

$$R \circ S \leftrightarrow \mu_{R \circ S}(x, z) = \min_y \left\{ \max(\mu_R(x, y), \mu_S(y, z)) \right\} \quad (3.34)$$

**max-estrela composição:**

$$R * S \leftrightarrow \mu_{R * S}(x, z) = \max_y \left\{ (\mu_R(x, y) * \mu_S(y, z)) \right\} \quad (3.35)$$

### 3.7 Relação difusa complementar

Uma relação difusa inverça da relação difusa  $R$  é representado por  $R^c$  e definido como:

$$R^c \leftrightarrow \mu_{R^c}(y, x) = \mu_R(x, y) \quad (3.36)$$

A seguir apresenta-se as relações difusas mais básicas para  $x, y \in X$ ,

Relação de identidade:

$$I \leftrightarrow \mu_I(x, y) = \begin{cases} 1; & x = y \\ 0; & x \neq y \end{cases}$$

Relação zero:

$$O \leftrightarrow \mu_O(x, y) = 0$$

Relação inversa:

$$E \leftrightarrow \mu_E(x, y) = 1$$

Exemplos destes três tipos de relações são:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{e} \quad E = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

### 3.13 Propriedades básicas das relações difusas

Na tabela 3.2 representa-se as propriedades básicas das operações com relações difusas, sendo que as propriedades da união, interseção e complemento para as relações difusas são iguais ao dos conjuntos difusos, e não aparecem nesta tabela.

Para a composição ( $\circ$ )	Para a Relação inversa ( $^c$ )
(1) $R \circ I = I \circ R = R$	(11) $(R \cup S)^c = R^c \cup S^c$
(2) $R \circ O = O \circ R = O$	$(R \cap S)^c = R^c \cap S^c$
(3) Em geral, $R \circ S \neq S \circ R$	$(R \circ S)^c = S^c \circ R^c$
(4) $(R \circ S) \circ T = R \circ (S \circ T)$	(12) $(R^c)^c = R, \overline{R^c} = \overline{R}$
(5) $R^{m+1} = R^m \circ R, R^c = I$	(13) $R \subseteq S \rightarrow R^c \subseteq S^c, \overline{R} \supseteq \overline{S}$
(6) $R^m \circ R^n = R^{m+n}$	
(7) $(R^m)^n = R^{mn}$	
(8) $R \circ (S \cup T) = (R \circ S) \cup (R \circ T)$	
(9) $R \circ (S \cap T) \subseteq (R \circ S) \cap (R \circ T)$	
(10) $S \subseteq T \rightarrow R \circ S \subseteq R \circ T$	

Tabela 3.2. Propriedades básicas das Relações Difusas.

### 3.14 Proposição difusa

Uma proposição da lógica difusa,  $P$ , é uma declaração que envolve conceitos sem limites claros de definição. Sentenças lingüísticas que tendem a expressar idéias subjetivas e que podem ser interpretados em forma ligeiramente diferente por vários indivíduos envolve tipicamente uma proposição difusa. Sentenças descrevendo o peso, beleza, ou preferências de cores podem ser utilizadas como exemplos de uma proposição difusa; o valor de verdade determinado a  $P$  pode se qualquer valor no intervalo  $[0, 1]$ . Então a determinação do grau de

verdade de uma proposição é mapeado no intervalo  $[0, 1]$  no universo  $U$  dos valores de verdade  $T$ ,

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

As proposições difusas são determinados a conjuntos difusos, suponhamos que  $P$  esta determinado a um conjunto difuso  $A$ , então o valor de verdade da proposição denotado por  $T(P)$ , esta dada por:

$$T(P) = \mu_A(x); \quad \text{onde } 0 \leq \mu_A \leq 1$$

Que deve ser interpretado da seguinte maneira: o grau de verdade da proposição  $P: x \in A$  é igual ao grau de pertinência de  $x$  ao conjunto  $A$ .

Os conectores lógicos de negação, disjunção, conjunção e implicação são também definidos para a lógica difusa. Considere as proposições  $P$  e  $Q$  como definidas nos conjuntos difusos  $A$  e  $B$  respectivamente então a definição dos conectores seria a seguinte:

Negação  $T(\bar{P}) = 1 - T(P)$

Disjunção  $P \vee Q \Rightarrow x \text{ é } A \text{ ou } B$   
 $T(P \vee Q) = \max(T(P), T(Q))$

Conjunção  $P \wedge Q \Rightarrow x \text{ é } A \text{ e } B$   
 $T(P \wedge Q) = \min(T(P), T(Q))$

Implicação  $P \rightarrow Q \Rightarrow X \text{ é } A \text{ então } x \text{ é } B$   
 $T(P \rightarrow Q) = T(P \vee Q) = \max(T(P), T(Q))$

A implicação também pode ser modelada na forma de regra,

$$P \rightarrow Q \quad \text{SE } x \text{ é } A \text{ ENTÃO } y \text{ é } B$$

que equiivale a seguinte relação difusa, R,

$$R = (A \times B) \cup (\bar{A} \times Y)$$

cuja função de pertinência está expressada pela seguinte fórmula,

$$\mu_R(x, y) = \max\left[\left(\mu_A(x) \wedge \mu_b(y)\right), \left(1 - \mu_A(x)\right)\right] \quad (3.37)$$

Se a implicação fosse da forma:

SE  $x$  é A ENTÃO  $y$  é B CASO CONTRÁRIO  $y$  é C

então a relação difusa equivalente seria:

$$R = (A \times B) \cup (\bar{A} \times C)$$

cuja função de pertinência estaria expressada por,

$$\mu_R(x, y) = \max\left[\left(\mu_A(x) \wedge \mu_b(y)\right), \left(1 - \mu_A(x) \wedge \mu_A(x)\right)\right] \quad (3.38)$$

## CAPÍTULO IV

### SISTEMAS ESPECIALISTAS

#### 4 Sistemas especialistas

##### 4.1 Definição e história

Designa-se com o nome de **SISTEMA ESPECIALISTA (SE)**, aquele sistema que tem a suficiente especialização em alguma das artes humanas e que opera em escala real e não em uma aplicação simulada. Designa-se também deste modo um sistema que joga um papel de um consultor dando recomendações sobre uma tarefa específica [BUC 84].

SE são programas de computador que tratam de reproduzir alguns aspectos da inteligência humana, seja este reconhecimento de fala, visão artificial, raciocínio lógico ou aprendizado baseado em experiências [SHO 84].

O conhecimento contido numa especialidade pode ser:

**Conhecimento Público:** As definições públicas de fatos e teorias, tais como livros de textos e referências nos quais o domínio do estudo está representado.

**Conhecimento Privado:** Conhecimento que o especialista foi adquirindo mediante a experiência pessoal na área e que não é publicado na literatura. O conhecimento privado consiste em um grande número de regras intuitivas Heurísticas. A heurística permite ao especialista humano de ter uma suspeita, até se poderia dizer educada, nos casos que fossem necessários para reconhecer abordagem promissoras aos problemas e para tratar eficientemente com dados não muito precisos e incompletos. A elucidação e reprodução deste tipo de conhecimento é a tarefa principal na construção de sistemas especialistas [HAY83].

As justificativas dos pesquisadores para guiar a atenção mais para o conhecimento e não em métodos formais de raciocínio são:

- ◆ A maioria dos problemas mais difíceis e interessantes não tem uma solução algorítmica tratável. Isto se deve a que muitas tarefas importantes, originadas em contextos físicos ou sociais complexos, geralmente escapam a uma análise rigorosa e descrição precisa.
- ◆ O especialista humano alcança um desempenho destacado justamente devido a que ele continuamente pode se informar ou atualizar.
- ◆ O valor intrínseco do conhecimento é um recurso escasso, cujo refinamento e reprodução cria riquezas. Dito de outra forma, a extração dos conhecimentos humanos e sua colocação em forma computacional, diminui grandemente os gastos de reprodução do conhecimento.

Pode-se dizer de forma resumida que a performance de um especialista depende criticamente de seu conhecimento, e da mesma forma a especialização de um SE depende também da representação daquele conhecimento.

Os SE têm diferenças muito importante, tanto dos sistemas de processamentos de dados como dos outros sistemas desenvolvidos em outros ramos da IA, que em forma resumida são as seguintes:

- ◆ O SE realiza tarefas difíceis com uma performance muito parecida ao dos especialistas.
- ◆ A ênfase é nas estratégias de resolução de problemas específicos ao domínio.



- ◆ Utiliza o seu próprio conhecimento para raciocinar sobre seu próprio processo de inferência e dar explicações e justificativas para as conclusões a que chega após uma inferência.

Portanto SE representam uma área da IA que envolve paradigmas, ferramentas e estratégias de desenvolvimento de sistemas.

O primeiro sistema especialista foi o DENDRAL, que apareceu nos finais dos anos 60, e foi um sistema que identificava estruturas químicas utilizando os dados de um espectrômetro de massa [Lindsay, Buchanan, Feigenbaum e Lederberg, 1980]. Foi justamente esse mesmo grupo que logo formaria o *Stanford Heuristic Programming Project* (HPP) e que, aproveitando a experiência adquirida no DENDRAL para em meados da década de 70, projetou o MYCIN.

Não se deve também esquecer que outro dos precursores dos SE foi a dissertação de Joel Moses, que conduz ao projeto MACSYMA em matemática simbólica (Grupo Mathlab, 1977). O MACSYMA incorpora centenas de regras extraídas de especialistas em matemática aplicada.

Logo temos o EXPERT (Weiss, 1978) que é uma linguagem de construção de sistemas especialistas que foi utilizado primariamente na área de oftalmologia, endocrinologia e reumatologia.

CADUCEUS, desenvolvido na Universidade de Carnegie-Mellon em Pittsburgh (Pople e outros, 1975), consistia em uma extremamente grande rede semântica de relações entre doenças e sintomas de medicina interna.

O MYCIN, desenvolvido na Universidade de Stanford (1975), estava direcionado ao diagnóstico e tratamento de doenças da sangue e consistia em aproximadamente 400 regras que relacionava certas condições à sua interpretação [SH084].

O sucesso que tiveram os SE foi o que ajudou a despertar de novo o interesse na investigação em IA.

## 4.2 Conhecimento e técnica

O conhecimento em um determinado domínio pode-se dizer que consiste na descrição, relacionamentos e procedimentos em algum domínio de interesse (Bernstein 1977) .

**Descrição:** Numa base de conhecimento pode-se dizer que são frases em alguma linguagem cujos componentes elementares consistem de características primitivas ou conceitos. O sistema de descrição inclui geralmente regras ou procedimentos para aplicação e interpretação das descrições em aplicações específicas.

**Relações:** Dentro da base de conhecimento também temos casos especiais de descrição que são conhecimentos e relacionamentos que expressam as dependências e associações entre itens que podem ser taxonômica, definição e empírica.

**Procedimentos:** São operações específicas feitas durante o raciocínio ou resolução de problemas.

Deve-se entender a diferença entre conhecimento e técnica. As características que refletem o conhecimento e a engenharia são grande velocidade, erro reduzido, redução da carga cognitiva e uma crescente adaptabilidade e robustez, embora que Técnica significa que se tenha o conhecimento correto e utilize-o efetivamente.

## **4.3 Característica de um sistema especialista**

### **4.3.1 Corpo de conhecimento**

O centro de um SE é o poderoso corpo do conhecimento que é acumulado durante a construção do sistema. O conhecimento deve ser explícito, organizado e acessível de forma a simplificar a tomada de decisões. A codificação e aquisição do conhecimento é um dos mais importantes aspectos de um SE [WAT86].

### **4.3.2 Conhecimento de alto nível**

O conhecimento de alto nível é uma das características mais utilizadas, que permite ao SE atuar como um especialista e produzir resultados de alta qualidade num tempo mínimo, provendo ajuda na resolução de problemas. Isto se deve ao fato que o conhecimento representado no SE é a opinião de especialistas de alto nível, o que representa anos de experiência em uma determinada tarefa [WAT86].

### **4.3.3 Poder de modelamento preditivo**

O poder de modelamento preditivo permite ao sistema atuar, como uma teoria de processamento de informação ou modelo de resolução de problemas num determinado domínio, apresentando as respostas desejadas para uma determinada situação problema e mostrando como eles poderiam mudar para novas situações, podendo explicar, em detalhes, como a nova situação poderia ser alterada. Isto permite que o usuário avalie o efeito potencial de novos fatos ou dados e compreenda os relacionamentos para sua solução [WAT86].

#### 4.3.4 Memória Institucional

A memória institucional se deve ao fato que a base de conhecimento foi elaborado com pessoas chaves num escritório ou departamento, o que representa a política atual ou procedimento operacional de tal grupo. Esta compilação de conhecimentos representa os consensos de opiniões de alto nível e o registro permanente das melhores estratégias e os métodos utilizados pelo grupo de trabalho. Quando as pessoas chaves vão embora, ainda fica retido seu conhecimento [WAT86].

#### 4.3.5 Facilidade de treinamento

O SE pode ser projetado de tal forma a fornecer treinamento, sendo que ele já tem habilidade de raciocinar sobre seu próprio processo de raciocínio, mediante a habilidade de explica-lo. A figura 4.1. mostra as características gerais dos SE.

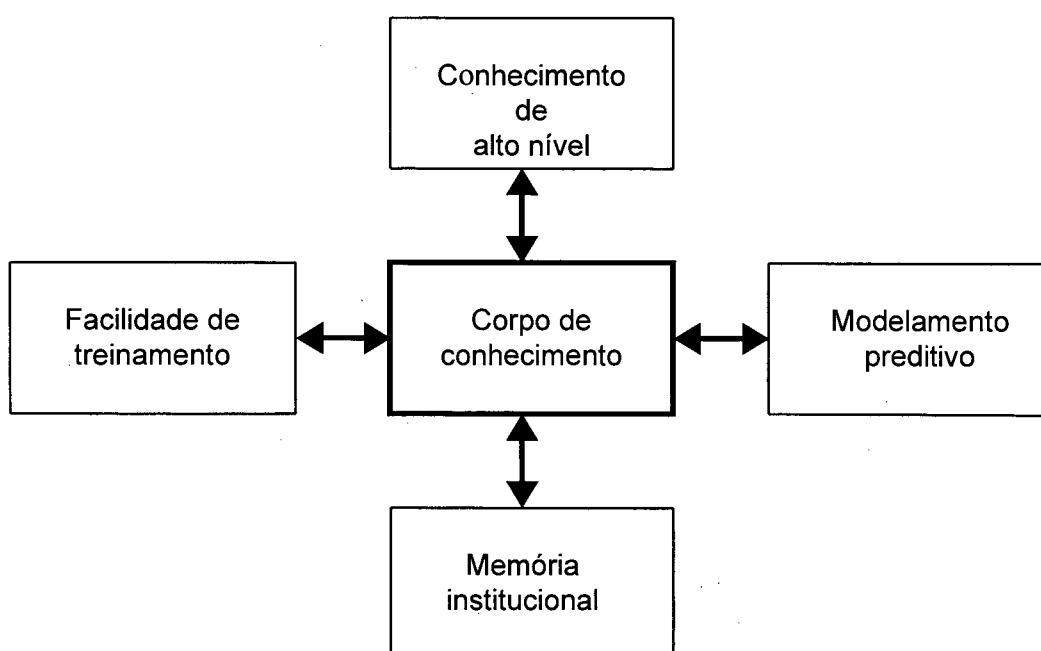


Figura 4.1. Características gerais de um sistema especialista [WAT86].

#### 4.4 Participantes na construção de um SE

Os participantes mais importantes na construção de um SE são: o sistema especialista em si, o especialista no domínio, o engenheiro do conhecimento e a ferramenta de construção do sistema especialista. Seus relacionamentos e funções podem ser encontrados em forma sumária na figura 4.2.

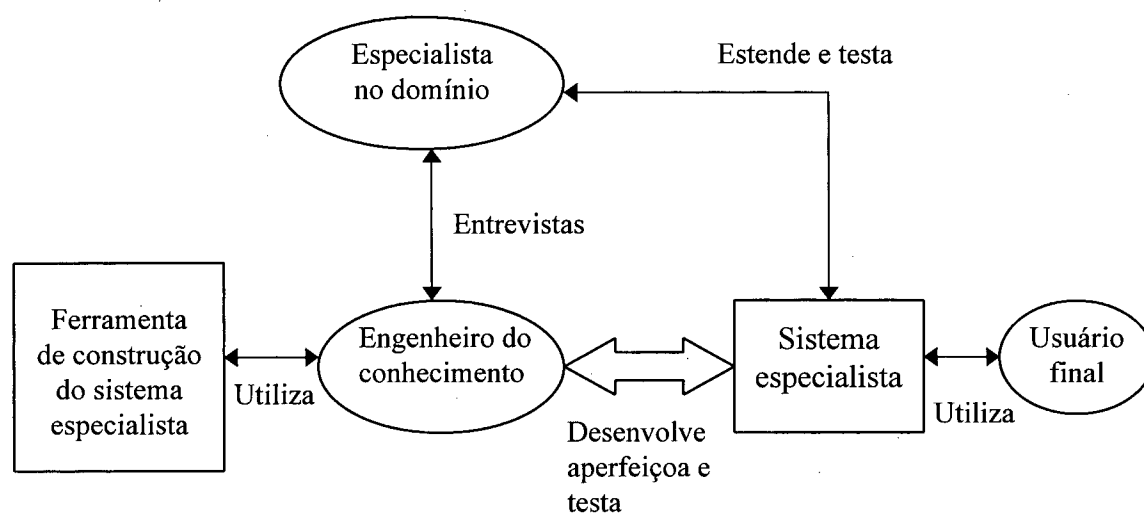


Figura 4.2. Participantes na construção de um sistema especialista [WAT86].

##### 4.4.1 Sistema especialista

Sistema especialista é a coleção de sistemas de computadores que resolvem problemas num domínio de interesse, tendo ele componentes para a resolução de problemas, componentes de suporte tais como ferramentas para a detecção de erros, facilidade de edição de dados, e dispositivos gráficos avançados, o que permite facilitar a entrada e leitura das informações pelo usuário.

#### **4.4.2 O Especialista no Domínio**

O especialista no domínio é uma pessoa informada e conhecida por produzir soluções boas a problemas particulares de uma área. Ele utiliza sua habilidade para fazer a busca de uma solução mais eficiente, isto é aproveitado pelo SE e introduz também os conhecimentos de outras fontes como livros e revistas especializadas.

#### **4.4.3 O Engenheiro de Conhecimento**

O engenheiro de conhecimento é uma pessoa geralmente com formação em Ciências da Computação e Inteligência Artificial e que conhece como construir um SE. Ele entrevista o especialista, organiza o conhecimento, e decide como representá-lo no SE.

#### **4.4.4 A ferramenta de construção do sistema especialista**

A ferramenta de construção do SE é a linguagem de programação utilizada pelo Engenheiro de Conhecimento ou programadores, para a construção do sistema especialista, devendo ele prover uma forma conveniente para a representação de complexos conceitos de alto nível. Deve-se sempre ter em mente que a ferramenta de construção do SE não é o SE, já que ele ainda não tem representado a base de conhecimento.

#### **4.4.5 Usuário Final**

Usuário final é a pessoa que utiliza o SE, podendo ser um cientista, um advogado, um estudante etc.

#### 4.5 Vantagens e desvantagens de um sistema especialista

As vantagens e desvantagens de um sistema especialista comparadas a um especialista humano são representadas em forma resumida nas tabelas 4.1 e 4.2.

<b>Especialista Humano</b>	<b>Sistema Especialista</b>
Percível	Permanente
Difícil de transferir	Fácil de transferir
Difícil de documentar	Fácil de documentar
Imprevisível	Consistente
Caro	Acessível

Tabela 4.1. Comparação das bondade de um SE frente a um especialista humano [WAT86].

<b>Especialista Humano</b>	<b>Sistema Especialista</b>
Criativo	Sem Criatividade
Adapta estratégias a novas situações.	Para adaptar suas estratégias precisa a introdução de novas regras.
Experiência sensorial	Experiência Simbólica
Amplia o enfoque ao problema	Enfoque restrito
Conhecimento de senso comum	Conhecimento técnico

Tabela 4.2. Comparação das desvantagens um SE frente a um especialista humano [WAT86].

#### 4.6 Estrutura de um sistema especialista

A organização do conhecimento em um SE é feita de um modo que separa o conhecimento relacionado à resolução do problema de outros conhecimentos do sistema, como por exemplo, imprimir caracteres no monitor ou modificar os dados em resposta ao comando do usuário.

A base de conhecimento num SE contém fatos e regras ou outras formas de representação que utilizam os fatos para alcançar uma decisão.

O motor de inferência tem um interpretador que decide como aplicar as regras para inferir um novo conhecimento e um escalonador (*scheduler*) que decide a ordem em que as regras devem ser aplicadas, como mostra a figura 4.3.

Para ter o conhecimento requerido, o SE deve ter a base de conhecimento contendo muitos conhecimentos de alto nível referente ao domínio do problema, e a máquina de inferência contém o conhecimento de como tornar efetivo o uso do domínio do conhecimento.

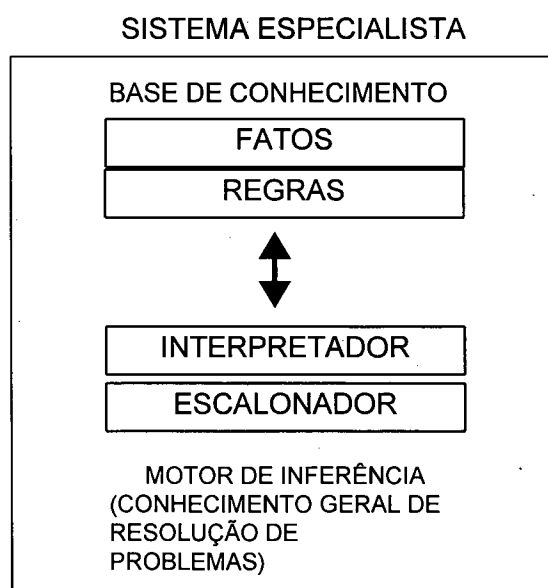


Figura 4.3. Estrutura de um sistema especialista [WAT86].



## **4.7 Etapas na construção de sistemas especialistas**

O desenvolvimento de um SE consta de cinco fases altamente interdependentes e superpostas: identificação, conceitualização, formalização, implementação e testes [BUC83].

### **4.7.1 Identificação**

Na etapa de identificação, o EC e o especialista determinam as características mais importantes do problema. Isto inclui identificação do problema propriamente dito (ex. tipo de problema e faixa de aplicação), os participantes no processo de desenvolvimento (ex. outros especialistas que participaram do projeto), os recursos necessários (ex. facilidade de tempo e facilidades computacionais) e as metas e objetivos para a construção do SE.

De todos eles, o que mais trabalho dá ao EC é a identificação do problema e sua faixa de aplicação. Geralmente, o problema inicialmente considerado é muito grande ou complexo e deve ser diminuído de tal forma a ter um tamanho gerenciável. O EC deve focar sua atenção em um pequeno e interessante subproblema e implementar roteiros para sua solução.

### **4.7.2 Conceitualização**

Na etapa de conceitualização, determinam-se os conceitos, relações e mecanismos de controle que são necessários para descrever o problema a ser solucionado. Neste momento se apresenta o problema de granularidade, ou seja, deve-se considerar a que nível de detalhe o conhecimento deve ser representado.

### **4.7.3 Formalização**

A formalização implica na expressão de conceitos importantes e relações de maneira formal. Neste momento o EC deve optar, tendo em conta as características do problema, se

representa-o em forma de *frames* ou em forma de regras de produção da forma de regras IF-THEN; e escolher a ferramenta de desenvolvimento de SE mais apropriado tendo em conta o método de representação escolhido.

#### 4.7.4 Implementação

Aqui na implementação o EC passa o conhecimento formalizado a um programa de computação. A construção do programa requer conteúdo, forma e integração.

O conteúdo vem do domínio do conhecimento, feito durante a conceitualização, isto é, a estrutura de dados, regras de inferência e estratégias de controle necessários para a solução do problema. O formalismo é especificado pela linguagem escolhida para desenvolver o sistema, e a interação tem a ver com a reorganização de vários blocos de conhecimentos, para eliminar erros globais entre a estrutura de dados e regras ou especificações de controle.

A implementação deve ser feita rapidamente, pois ela permite checar a efetividade do projeto de decisão e é muito possível que o código inicial logo deva ser revisado, reestruturado e talvez descartado.

#### 4.7.5 Testes

Na etapa de testes se avalia a performance e utilidade do protótipo do programa. Aqui participa ativamente o especialista, que ajuda ao EC a avaliar o protótipo uma vez que ele trabalhe com poucos exemplos, deve-se testar em vários problemas para avaliar sua performance e utilidade. Algumas das perguntas que se deve fazer na hora de testar a performance do protótipo são:

- ◆ O sistema toma decisões que o especialista concorda como apropriado?.

- ◆ São as regras de inferências corretas, consistentes e completas?.
  
- ◆ Permitirão as estratégias de controle que o sistema considere os itens numa ordem natural como o especialista prefere?.
  
- ◆ São as fundamentações do sistema adequadas para descrever como e porque uma conclusão foi alcançada? etc..

Na hora de avaliar a utilidade do sistema, surge outra bateria de perguntas tais como:

- Será a solução do problema útil ao usuário de um modo significativo?.
- São as conclusões do sistema apropriadamente organizadas, ordenadas e presentes com o nível correto de detalhe?.
- É o sistema suficientemente rápido para satisfazer ao usuário?.
- É a interface do sistema suficientemente amigável? etc.

#### **4.8 Adquirindo o conhecimento de um especialista**

No processo de aquisição de conhecimento, o EC usualmente obtém o conhecimento por meio da interação direta com o especialista. Estas interações são uma série de prolongadas entrevistas sistemáticas, durante as quais o EC apresenta para o especialista problemas, questionamentos, dados e casos reais de tal forma que estes sejam estudados e analisados, observando como é que ele avalia e resolve o caso. Por tanto, o especialista vai passando para o EC de forma direta ou indireta conhecimentos, conceitos, soluções etc. Como mostra a figura 4.4.

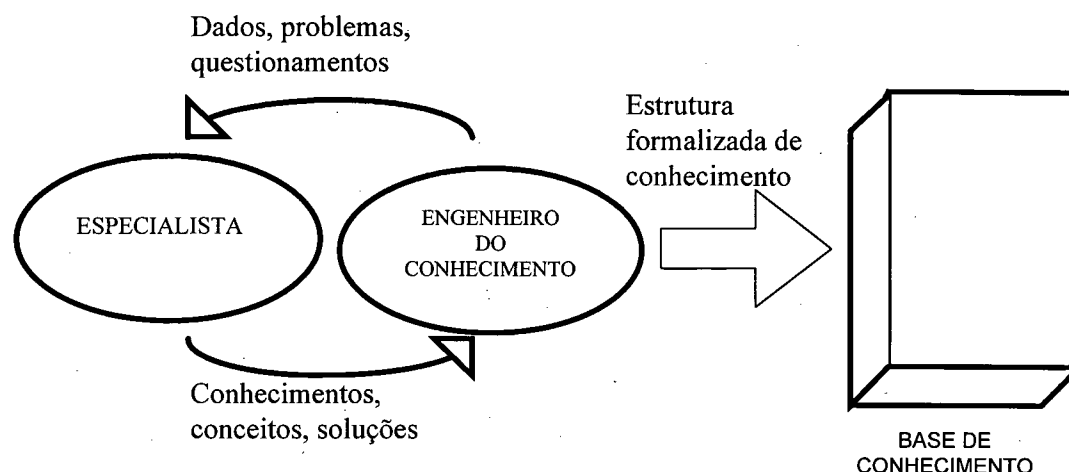


Figura 4.4. Processo típico de aquisição de conhecimento para a construção de SE [WAT86].

Raras vezes é efetivo perguntar diretamente ao especialista as regras e métodos para resolver um problema particular no domínio de estudo. Isto deve-se a que o especialista faz complexos julgamentos rápidos, combinando as peças básicas do conhecimento tão rapidamente que é difícil para ele descrever o processo. No momento de explicar seu raciocínio, ele leva em conta somente os passos maiores e esquece os menores, porque ditos passos são muito óbvios. Na tabela 4.3 apresenta-se em forma sumaria alguma das técnicas de extração do conhecimento do especialista [WAT86].

#### 4.9 Sistemas especialistas aplicados à área médica

A área médica foi um dos alvos principais para a construção de SE. Isto deve-se ao fato que nesta área a forma de manipulação do conhecimento é muito complexa, o que não permite fazer uma representação numérica, mas sim simbólica, relacionando as informações e orientando-as até chegar numa conclusão; o diagnóstico. Este relacionamento de informações é feito com regras de inferência que reproduzem de certa forma o raciocínio do médico mediante a representação do conhecimento dele em uma base de conhecimentos, aplicando as técnicas de aquisição de conhecimento, o que permite pensar que o SE ao fazer um diagnóstico tem um comportamento inteligente.

Método	Descrição
Observação no local	Observar o especialista resolvendo problemas reais no seu lugar de trabalho.
Discussão do problema	Explorar o tipo de dado, conhecimento e procedimento necessários para resolver o problema específico.
Descrição do problema	O especialista descreve um protótipo de problema para cada tipo de resposta no domínio em estudo.
Análise do problema	Apresenta-se ao especialista uma série de problemas reais para resolver em voz alta, sondando as explicações dos passos no processo de raciocínio.
Refinamento do sistema	Faz com que o especialista dê para você uma série de problemas para resolver utilizando as regras adquiridas por meio de entrevistas.
Exame do sistema	Faz com que o especialista examine e critique as regras do protótipo do sistema e a estrutura de controle.
Validação do sistema	Apresentar os casos resolvidos pelo especialista e o protótipo de sistema para outros especialistas fora do projeto.

Tabela 4.3. Algumas das técnicas de aquisição do conhecimento [BUC83].

O primeiro SE nesta área foi o MYCIN que, como já foi dito anteriormente, pode-se considerar como um dos SE pioneiros e que assistia no diagnóstico de pacientes com doenças infecciosas (bacteremia, meningitis e cistitis), e que sem dúvida marcou o desenvolvimento de SE em outras áreas, pelas técnicas que foram revolucionárias na época.

Seguidamente na tabela 4.4 se apresenta os nomes de alguns dos SE aplicados na área médica:

Nome do Sistema	Domínio de Ação	Equipe
ABEL	Diagnóstico de desordens em acido base e eletrólitos.	[PAT881b,82c,82D]
AI/RHEUM	Assiste no diagnóstico de enfermidades reumatológicas.	[LIN80a,82]
CASNET/GLAUCOMA	Diagnóstico de glaucomas aconselhando um plano de terapia.	[SZO78, WEI78, KUL80a] [WEI81c, KUL82, ALT84b]
EEG ANALYSIS SYSTEM	Realiza análises dos registros de eletroencefalograma de pacientes renais.	[BAA84]
DRUG INTERACTION CRITIC	Assiste na administração de drogas que interajam com outras.	[ROA84]
CENTAUR	Assiste na interpretação de provas funcionais pulmonares	[AIK83a]
CLOT	Diagnostica desordens no sistema de coagulação do sangue.	[BEN80, BEN84]
HEPAXPERT-I	Interpreta os resultados de estudos sorológicos para o diagnóstico de Hepatite A e B	[ADL94]

Tabela 4.4 Exemplo de SE aplicados na área médica.

## CAPITULO V

### SISTEMAS ESPECIALISTAS DIFUSOS

#### 5 Sistemas especialistas difusos (SED)

##### 5.1 Introdução

Na atualidade é indispensável que um sistema especialista modele a difusividade das informações, pois só deste modo poderá, de certa forma, imitar a capacidade do especialista, que tem habilidade para raciocinar ou inferir numa área determinada e é capaz de processar informações ambíguas.

Os SED iniciaram-se nos anos 70 com aplicações na área de controle, com muito bons resultados até hoje. Zadeh em 1968, num artigo intitulado “Algoritmos Difusos”, descreve os métodos de algoritmos de predicado na forma If/Then, lançando a idéia de expressar a ambigüidade por meio de algoritmos lingüísticos, sendo esta uma idéia muito avançada para a época.

Nos anos 1980, com a teoria de Dempster-Shafer, o valor da probabilidade que modela a incerteza em sistemas especialistas já não é mais um valor numérico simples, passando a ser expresso na forma de intervalo. Mas como foi colocado anteriormente, o fato de um sistema especialista ser um sistema que imita o modo como se expressa o conhecimento de um especialista, então o método para construir sistemas especialistas efetivos deve incorporar a forma em que o especialista realiza esta tarefa, devendo-se então prestar atenção ao processo mental do mesmo para realizar seu trabalho.

O pensamento humano envolve caracterizações por meio de todos os aspectos para obter informações incompletas e trabalhar com elas, tais como inferência de similaridade, avaliação de objetivos complexos e fazer o julgamento e tomada de decisão em circunstâncias incertas.

Pode-se dizer que o conhecimento humano do dia a dia deve-se à capacidade de pensar em palavras, sendo justamente a teoria difusa, que propõe um método para captar a ambigüidade da linguagem e os conceitos, a essência de um SE.

A quantidade de SE publicados recentemente referem-se a um valor perto dos 2.000, mas só 2% está sendo utilizado atualmente. Com relação à aplicabilidade, os sistemas especialista difusos superam os SE não difusos [TER94]. Isto se deve ao fato de que a teoria difusa ajuda a solucionar as três maiores dificuldades no desenvolvimento de um SE as quais são:

1. Expressão do conhecimento;
2. Utilização do conhecimento, e;
3. Aquisição do conhecimento.

Na expressão do conhecimento, por exemplo, a expressão lingüística é de nível superior, permitindo que o conhecimento real, o conhecimento em forma de regras e o conhecimento em forma de juízos possam ser expressos de tal forma que incluam a ambigüidade.

Quanto à utilização do conhecimento, a lógica difusa é particularmente efetiva mediante a utilização da inferência difusa, que permite que esta seja feita por uma base de regras difusas que utilizam informações ambíguas.

O terceiro problema, de extrair os conhecimentos experimentais do especialista, torna-se mais evidente porque o sistema faz uso dos mesmos termos que o especialista utiliza,



através de termos lingüísticos que adquirem valores que captam melhor a ação tomada pelo especialista em forma qualitativa. Exemplo “se x é pequeno, então y é grande”.

Os SED aplicados ao controle já demonstraram sua aplicabilidade, mas também estão demonstrando serem efetivos em outros campos, como análises de falhas, diagnóstico médico, programação de horários, sistemas de planejamento, previsão econômica, investimento em estoques, previsão de vendas, assistência no gerenciamento de decisões e assessoria.

## 5.2 Sistema de inferência difuso

O sistema de inferência difuso é também conhecido como sistema baseado em regras, modelos difusos, memória associativa difusa (FAM) ou controlador difuso quando aplicado a tarefas de controle.

Basicamente um sistema de inferência difuso está composto de cinco blocos funcionais, representado na figura 5.1.

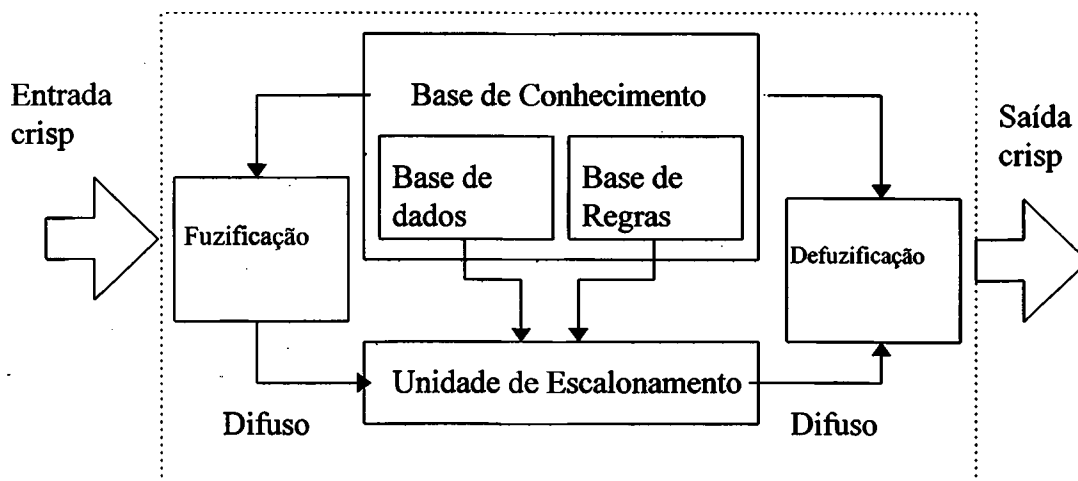


Figura 5.1. Sistema de Inferência Difuso [LEE90].

### 5.2.1 Base de dados

Na Base de dados estão definidas as funções de pertinência dos conjuntos difusos que são utilizados pela base de regras difusas. A figura 5.2 mostra um exemplo de conjuntos difusos definidos na base de dados.

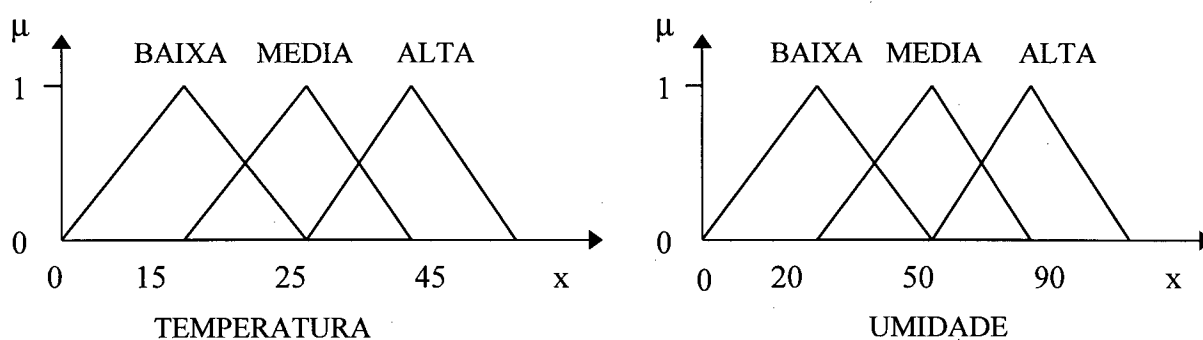


Figura 5.2. Exemplos de conjuntos difusos definidos na base de dados.

### 5.2.2 Fuzificação

A fuzificação é a etapa inicial da inferência na qual os valores numéricos são transformadas em graus de pertinência para um valor lingüístico determinado. Isto é realizado mediante a aplicação da função de pertinência correspondente. A figura 5.3 mostra a fuzificação da temperatura de entrada *crisp* de 40 °C.

Mediante a fuzificação, encontra-se que um valor numérico *crisp* de temperatura de 40 °C faz com que a variável lingüística TEMPERATURA assuma dois valores: MEDIA com um grau de pertinência 0.3 e ALTA com um grau de pertinência 0.6.

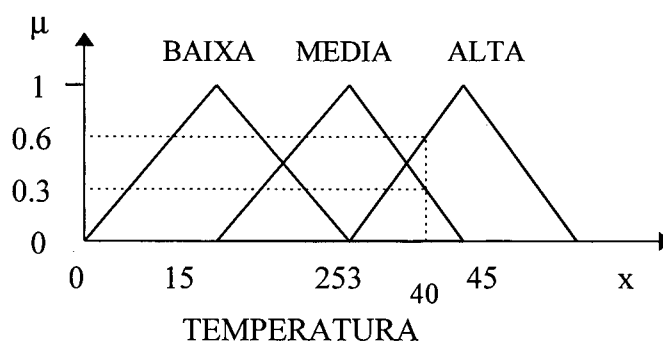


Figura 5.3. Fuzificação da temperatura de entrada 40°C.

### 5.2.3 Base de Regras

Na base de regras encontram-se os conjuntos de regras que podem ser aplicadas pelo sistema de inferência, expressas em forma canônica.

### 5.2.4 Unidade de escalonamento

A unidade de escalonamento realiza as tarefas de inferência das regras contidas na base de regras, determinando quais as regras a serem aplicadas e sua ordem, levando em consideração a entrada fuzificada.

### 5.2.5 Defuzificação

Na defuzificação transforma-se os valores difusos resultantes da inferência em valores de saídas numéricos, mediante a aplicação de um dos métodos de defuzificação. A base de regras, juntamente com a base de dados, são referenciados como base de conhecimento.

O algoritmo para o raciocínio difuso utilizado pelos sistemas de inferência difusos é o seguinte :

- a) Comparar os valores das variável de entrada com as funções de pertinência, da premissa para obter os valores de pertinência a cada termo lingüístico (fuzificação).
- b) Combinar os valores de pertinência da premissa para obter a força com que é disparada cada regra. Isto é feito mediante um operador T-norma, freqüentemente multiplicação ou mínimo.
- c) Gerar o conseqüente (difuso ou crisp) de cada regra dependendo da força com que a mesma é disparada.
- d) Fazer agregação dos conseqüentes para produzir uma saída crisp (defuzificação).

### 5.3 Tipos de raciocínio difuso

Existem vários tipos, mas dependendo do tipo de raciocínio difuso e as regras IF-THEN utilizadas pode-se classificá-las em três tipos. A figura 5.4 mostra os tipos comuns de regras difusas na forma IF-THEN e mecanismos difusos de raciocínio.

- a) TIPO 1: A saída total é uma média dos pesos das saídas crisp de cada regra induzida pela força com que a regra é disparada e a função de pertinência de saída. A função de pertinência de saída utilizada neste caso deve ser monotonicamente não decrescente [TSU79].
- b) TIPO 2: A saída difusa total é derivada mediante a aplicação do operador “max” às saídas difusas (cada uma delas é igual ao mínimo das forças com que foi disparada e a função de pertinência de saída de cada regra). Foram propostos vários métodos para escolher a saída crisp final baseados na saída difusa total; algumas delas são centróide, bissetriz, media das máximas, etc [LEE90].

- c) TIPO 3: São utilizadas as regras difusas em forma IF-THEN de Takagi e Sugeno. A saída de cada regra é uma combinação linear das variáveis de entradas mais um termo constante e a saída final é uma média dos pesos das saídas de cada regra [SUG88].

#### 5.4 Expressão das regras em forma canônica num sistema especialista difuso baseado em regras (SEDBR)

Se há sistema com  $n$  entradas e  $m$  saídas tem-se três espaços gerais presentes no sistema:

- a) O espaço das possíveis condições na entrada do sistema, que pode ser representado, em geral, por uma coleção de subconjuntos difusos  $A^k$ , para  $k = 1, 2, \dots$ , que são partições difusas do espaço  $X$ , representados por funções de pertinência,

$$A^k \in F(X); \mu_{A^k}(x) : X \rightarrow [0, 1], \text{ para } k = 1, 2, \dots$$

- b) O espaço de possíveis saídas conseqüentes, comandos de controle ou decisões recomendadas, no caso de tomada de decisões, baseados em algumas condições específicas nas entradas, e em geral pode ser representado por uma coleção de subconjuntos difusos  $B^k$ , para  $k = 1, 2, \dots$ , que são partições difusas de um espaço  $Y$  expressado mediante uma função de pertinência,

$$B^k \in F(Y); \mu_{B^k}(y) : Y \rightarrow [0, 1], \text{ para } k = 1, 2, \dots \quad (5.1)$$

- c) O espaço dos possíveis mapeamentos de relações a partir dos espaços de entrada  $X$  ao espaço de saída  $Y$ . Estas são representadas em geral por relações difusas  $R^k$ , para  $k = 1, 2, \dots$ , e expressas pela função de pertinência,

$$R^k \in F(X \times Y); \mu_{R^k}(x, y) : X \times Y \rightarrow [0, 1], \text{ para } k = 1, 2, \dots \quad (5.2)$$

Entenda-se por forma canônica de expressão das regras de produção, um conjunto de restrições incondicionais seguidas de um conjunto condicional de restrições.

Estas restrições tanto condicionais como incondicionais são expressas por conectores lingüísticos tais como “and” e “or”. A figura 5.5 mostra a forma canônica de um conjunto de regras.

$R^1$	Restrição $R^1$
$R^2$	Restrição $R^2$
$\vdots$	
$R^k$	Restrição $R^k$
$R^{k+1}$	IF condição $C^1$ , THEN restrição $R^{k+1}$
$R^{k+2}$	IF condição $C^2$ , THEN restrição $R^{k+2}$
$\vdots$	
$R^r$	IF condição $C^{r-k}$ , THEN restrição $R^r$

Figura 5.5. Forma canônica de um conjunto de regras.

Em geral pode-se encontrar três formas gerais de declaração em qualquer regra de produção:

a) Declaração determinante, ex.:

x = pequeno

estação = primavera

temperatura = quente

x é grande

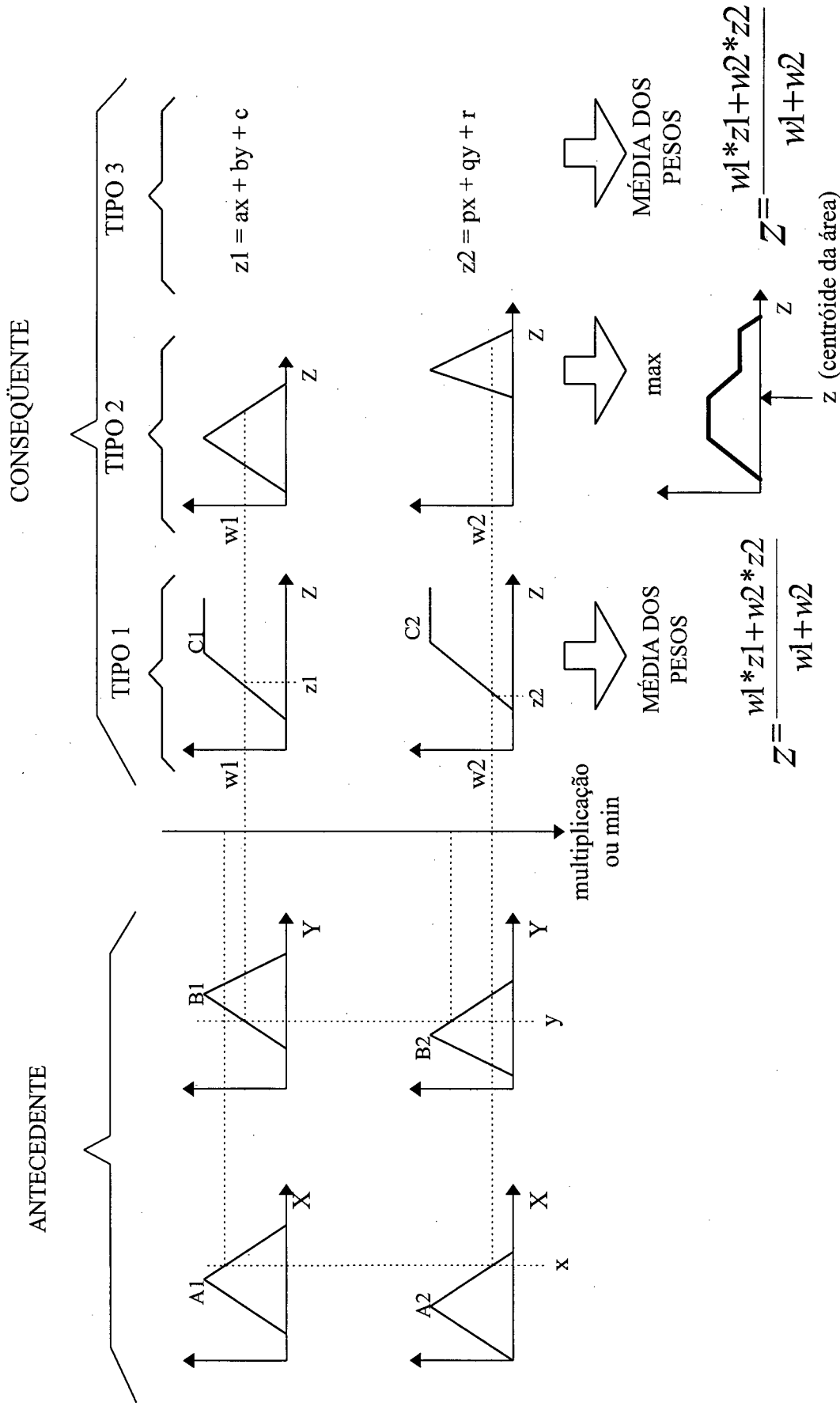


Figura 5.4. Tipos comuns de regras difusas na forma IF-THEN e mecanismos difusos de raciocínio [LEE90].

b) Declaração condicional, ex.:

IF x é pequeno THEN y é grande ELSE y não é grande

IF x é positivo THEN diminua y ligeiramente

IF o tomate está vermelho THEN o tomate esta maduro

c) Declaração de ações incondicionais. Ex.:

multiplique por x

diminua a temperatura

pare

#### 5.4.1 Múltiplos antecedentes e múltiplos conseqüentes unidos por “AND” e “OR”

Pode-se determinar um número de regras complexas com mais de um antecedente ou conseqüente e ilustrar como representar as mesmas em forma canônica simples, tais como:

a) IF x é  $A^1$  AND  $A^2$  ... AND  $A^L$  THEN y é  $B^S$

Assumindo um novo conjunto difuso  $A^S$  como

$$A^S = A^1 \cap A^2 \cap \dots \cap A^L \quad (5.3)$$

expressadas por meio da função de pertinência

$$\mu_{A^S}(x) = \text{Min} [ \mu_{A^1}(x), \mu_{A^2}(x), \dots, \mu_{A^L}(x) ] \quad (5.4)$$

então a regra deve ser escrita da seguinte maneira,

IF  $A^S$  THEN  $B^S$

b) IF x é  $A^1$  OR  $A^2$  ... OR  $A^L$  THEN y é  $B^S$

O novo conjunto  $A^S$  esta definido como

$$A^S = A^1 \cup A^2 \cup \dots \cup A^L \quad (5.5)$$

$$\mu_{A^S}(x) = \text{max} [ \mu_{A^1}(x), \mu_{A^2}(x), \dots, \mu_{A^L}(x) ] \quad (5.6)$$

podendo a regra ser escrita da seguinte maneira,

IF  $A^S$  THEN  $B^S$



### 5.4.2 Sentenças condicionais com “ELSE”

a) IF  $A^1$  THEN  $B^1$  ELSE  $B^2$

esta regra pode ser descomposta por duas regras em forma canônica,

IF  $A^1$  THEN  $B^1$

IF NOT( $A^1$ ) THEN  $B^2$

b) IF  $A^1$  THEN ( $B^1$  ELSE IF  $A^2$  THEN  $B^2$ )

pode ser representado em forma canônica, como,

IF  $A^1$  THEN  $B^1$

IF NOT( $A^1$ ) AND  $A^2$  THEN  $B^2$

### 5.4.3 Regras IF-THEN aninhadas

a) IF  $A^1$  THEN (IF  $A^2$  THEN  $B^1$ )

pode-se representar por,

IF  $A^1$  AND  $A^2$  THEN  $B^1$

b) IF  $A^1$  THEN (IF  $A^2$  THEN (IF ... ( $B^1$ ) ...))

é um caso geral que pode-se representar por,

IF  $A^1$  AND  $A^2$  AND ... THEN  $B^1$

c) IF  $A^1$  THEN IF  $A^2$  THEN IF  $A^3$  THEN  $B^1$  ELSE  $B^2$  ELSE  $B^3$

devesse transformar em,

IF  $A^1$  AND  $A^2$  AND  $A^3$  THEN  $B^1$

IF  $A^1$  AND  $A^2$  AND NOT( $A^3$ ) THEN  $B^2$

IF NOT( $A^1$ ) THEN  $B^3$

d) IF A<sup>1</sup> THEN B<sup>1</sup> AND B<sup>2</sup> IF A<sup>2</sup> THEN B<sup>3</sup> IF A<sup>3</sup> THEN B<sup>4</sup>

transforma-se em,

IF A<sup>1</sup> THEN B<sup>1</sup>

IF A<sup>1</sup> THEN B<sup>2</sup>

IF A<sup>1</sup> AND A<sup>2</sup> THEN B<sup>3</sup>

IF A<sup>1</sup> AND A<sup>2</sup> AND A<sup>3</sup> THEN B<sup>4</sup>

### 5.5 Agregação de regras

O problema de obter um conjunto difuso de saída  $y$  geral partindo da aplicação de regras individuais é conhecida como problema de agregação de regras. Para simplificar considera-se somente dois casos extremos simples:

a) **Sistemas conjuntivos de regras:** É utilizado no caso que o sistemas de regras devem ser satisfeitas juntas. Neste caso a saída agregada é calculada por meio da interseção difusa de todas as saídas das regras individuais,

$y = y^1 \cap y^2 \cap \dots \cap y^r$  que é definida por uma função de pertinência,

$$m_y(y) = \text{Min} [\mu_{y^1}(y), \mu_{y^2}(y), \dots, \mu_{y^r}(y)], \text{ para } y \in Y \quad (5.7)$$

b) **Sistemas disjuntos de regras:** Quando o sistema de regras onde a satisfação de ao menos uma regra é requerida, a agregação da saída é calculada por meio da união de todas as contribuições das regras individuais

$y = y^1 \cup y^2 \cup \dots \cup y^r$  que é definida por uma função de pertinência,

$$m_y(y) = \text{max} [\mu_{y^1}(y), \mu_{y^2}(y), \dots, \mu_{y^r}(y)], \text{ para } y \in Y \quad (5.8)$$

### 5.6 Classes especiais de modelos de sistemas especialistas difusos baseados em regras

a) A primeira classe são as que tem as restrições de entrada dadas na forma de conjuntos clássicos e as restrições de saída são dadas por *singleton*, ex.

$$R^1: \text{ IF } x_0 < x < x_1 \text{ THEN } y = f_1(x) \text{ ELSE}$$

$$R^2: \text{ IF } x_1 < x < x_2 \text{ THEN } y = f_2(x) \text{ ELSE}$$

$$\vdots$$

$$R^r: \text{ IF } x_{r-1} < x < x_r \text{ THEN } y = f_r(x)$$

- b) A segunda classe são aquelas nas quais as condições de entrada são conjuntos clássicos e a saída é expressa por meio de conjuntos difusos ou relações difusas, ex.:

$$R^1: \text{ IF } x_0 < x < x_1 \text{ THEN } y \text{ é } B^1 \text{ ELSE}$$

$$R^2: \text{ IF } x_1 < x < x_2 \text{ THEN } y \text{ é } B^2 \text{ ELSE}$$

$$\vdots$$

$$R^r: \text{ IF } x_{r-1} < x < x_r \text{ THEN } y \text{ é } B^r$$

Onde  $B^i$  para  $i = 1, 2, \dots$ , são conjuntos difusos definidos no espaço de saída, podendo-se obter saídas *crisp* mediante um processo de defuzificação.

- c) A terceira classe são aqueles cujas entradas são definidas em forma de conjuntos difusos definidos no universo de discurso, e as saídas são dadas na forma de *singletons* ou uma função *crisp* não linear, ex.:

$$R^1: \text{ IF } x \text{ é } A^1 \text{ THEN } y = f_1(x) \quad \text{ELSE}$$

$$R^2: \text{ IF } x \text{ é } A^2 \text{ THEN } y = f_2(x) \quad \text{ELSE}$$

$$R^r: \text{ IF } x \text{ é } A^r \text{ THEN } y = f_r(x).$$

- d) A quarta classe são aqueles cujas entradas e saídas são definidas em forma de conjuntos difusos definidos no universo do discurso, ex.:

$R^1$ : IF  $x$  é  $A^1$  THEN  $B^1$  ELSE  
 $R^2$ : IF  $x$  é  $A^2$  THEN  $B^2$  ELSE  
 $R^f$ : IF  $x$  é  $A^f$  THEN  $B^f$ .

### 5.7 Aplicação de sistemas especialistas difusos no diagnóstico médico

Quando se fala de enfermidade fala-se de um estado do corpo humano que se desvia da condição normal. Diagnóstico neste caso significará a detecção da parte do corpo que está afetada, grau e nome da enfermidade.

Quanto mais cedo tenta-se diagnosticar uma enfermidade, as informações e os indicadores são mais ambíguos, freqüentemente sensoriais, tais como ruído estranho em forma intermitente, cheiro de queimado ou um pequeno aumento na temperatura.

Outro caso de informação ambígua presente no diagnóstico médico é quando a diferença entre normal e anormal não é muito clara. Isto acontece na detecção precoce de doenças físicas e psicológicas.

Para o julgamento do diagnóstico de um paciente leva-se em conta os sintomas observados, resultados dos testes, queixas principais do paciente, sinais observados no paciente, história médica e circunstâncias do diagnóstico. O último tópico é importante, pois é evidente que o diagnóstico feito num *checkup* e outro feito frente a uma possível operação, os indicadores de uma possível enfermidade e a necessidade de testar novamente são importantes no primeiro caso, mas no segundo é mais importante a certeza com relação à possibilidade, pois normalmente uma operação não é feita sem ter certeza da existência de uma doença na parte afetada.

As ambigüidades que se encontram freqüentemente quando se estudam os sintomas são expressões de freqüência tais como, “na maioria dos casos”, “primeiro”, “algumas poucas”, e “quase”.

Também existem circunstâncias ambíguas na observação da condição do paciente, tais como, laboratórios de análises diferentes têm resultados diferentes na dosagem da concentração de uma substância, quando valores entre os intervalos normais e anormais são obtidos, ou quando estão faltando dados necessários para um diagnóstico mais preciso.

A relação entre sintomas e doenças está muito longe de ser *crisp* e sendo elas por natureza difusas, é comum encontrar as seguintes expressões ao falar com os especialistas ou nos livros de textos:

“Pancreatite aguda está **quase sempre** relacionada com debilitamento e vômitos”

“**Tipicamente** uma pancreatite aguda começa com uma repentina dor no abdome”

“**Freqüentemente** é observado a ocorrência simultânea de hepatite alcoólica e cirrose”

É evidente que a modelagem de palavras como quase sempre, tipicamente, freqüentemente, são difusas, devendo ser expressas na forma de termos lingüísticos.

### 5.8 Métodos utilizados para o tratamento de ambigüidade no diagnóstico médico

Existem muitos métodos propostos para o tratamento da ambigüidade no diagnóstico médico, apresentando-se, a seguir, os três métodos básicos mais importantes [TER94].

### 5.8.1 Diagnóstico utilizando relações difusas

Neste caso considera-se que os sintomas como indicadores de anormalidade, que é resultado de uma certa causa, e o diagnóstico é justamente tentar especificar esta causa.

Considere-se os conjuntos de sintomas como componentes do seguinte conjunto  $Y = \{ y_1, y_2, \dots, y_n \}$ , e se  $r_{ij}$  expressando a existência de uma conexão entre  $x_i$  e  $y_j$  e considerando uma matriz de  $m$  linhas e  $n$  colunas,  $R = \{r_{ij}\}$ . Então neste caso a relação difusa é expressa pela equação:

$$y = x \circ R \quad (5.9)$$

que implica as seguintes operações:

$$y_j = \bigvee_i \{x_i \wedge r_{ij}\}; \quad j = 1, 2, \dots, n \quad (5.10)$$

$\vee$  e  $\wedge$  representa max e min respectivamente.

Geralmente encontram-se  $x$  (causa) tendo  $y$  (os sintomas) e  $R$  (a relação), e este seria o processo de diagnóstico. Especialmente quando  $y_j$  e  $r_{ij}$  são valores no intervalo  $[0, 1]$ , este problema é chamado de **problema inverso** da equação relacional difusa.

O significado da última equação, considerando  $\vee$  e  $\wedge$  como operadores Booleanos *and* e *or*, para maior simplicidade seria:

Quando a causa primaria  $i$  ocorre,  $x_i = 1$ .

Quando o sintoma  $j$  aparece,  $y_j = 1$ .

Quando o causa primaria  $i$  ocorre e o sintoma  $j$  aparece,  $r_{ij} = 1$ .

Em todas as outras circunstâncias tudo é zero.

Então pode-se escrever,

$$y_j = (x_1 \text{ and } r_{1j}) \text{ or } (x_2 \text{ and } r_{2j}) \text{ or } \dots \text{ or } (x_m \text{ and } r_{mj}); \quad j = 1, 2, \dots, n$$

o significado desta expressão é a seguinte, quando o sintoma  $y_j$  aparece ao menos uma das causas conectadas a ela ocorre e da mesma forma se  $y_j$  esta ausente nenhuma das causas a ela

relacionada ocorrem. Em primeiro lugar define-se a composição  $\omega$  e  $\tilde{\omega}$

Definição: Dado  $p, q \in [0, 1]$

$$p \omega q = \begin{cases} q & \text{se } p > q \\ [q, 1] & \text{se } p = q \\ \emptyset & \text{se } p < q \end{cases}$$

$$p \tilde{\omega} q = \begin{cases} [0, q] & \text{se } p > q \\ [0, 1] & \text{se } p \leq q \end{cases}$$

nesta definição de composição  $\omega$  é a solução do problema inverso  $p$  e  $q$  são dados; encontre  $z$  de forma que  $p \wedge z = q$ ,  $\emptyset$  indica que não tem solução.

O algoritmo para a solução do problema inverso é a seguinte:

1. Encontrar a matriz  $U$  e  $\tilde{U}$ ,

$$U = \{ u_{ij} \} = \{ r_{ij} \omega y_j \}$$

$$\tilde{U} = \{ \tilde{u}_{ij} \} = \{ r_{ij} \tilde{\omega} y_j \}$$

2. Encontrar a matriz  $W = \{ w_{ij} \}$

$$W_{ij} = \begin{cases} u_{ij} & \text{para } \exists i \in \{i \mid u_{ij} \neq \emptyset\} \\ \sim & \text{para as outras } i \\ u_{ij} & \end{cases}$$

como podem existir diferentes formas de fazer esta seleção pode-se achar diferentes matrizes  $W$  e para distinguir utiliza-se,  $W^k = \{ W_{ij}^k \}$ .

3. Para cada  $k$ , calcular;

$$X_i^k = \bigcap_j W_{ij}^k ; \quad i = 1, 2, \dots, n,$$

tem-se nela a solução final.

### 5.8.2 Diagnóstico utilizando padrões de sintomas e graus de conformidade

Considera-se o caso no qual uma grupo específico de sintomas é indicado e então a doença é determinada. Neste caso o diagnóstico significa a investigação do grau de conformidade dos padrões de sintomas previamente especificados e a observação atual [SAN79].

Considere-se  $\{ A_j, j = 1, 2, \dots, n \}$  um grupo de nomes de atributos conectados com os sintomas ex: temperatura, concentração, etc. Se  $A_j$  adquire a forma de expressões lingüísticas tais como “muito alto” e o conjunto suporte deste conjunto difuso é expresso por  $U_j$ . E os conjuntos difusos caracterizados pela função



$\mu_{R_j} : U_j \rightarrow [0, 1]$  são expressos por  $R_j$ . A proposição difusa  $P_j$ , que é algo similar a “alta temperatura” pode ser representada da seguinte maneira:

$$P_j = \text{“ } A_j \text{ é } R_j \text{”}; \quad j = 1, \dots, n.$$

e a proposição composicional,

$$S = \text{“ } P_1 \text{ e } P_2 \text{ e } \dots \text{ e } P_n \text{”}$$

pode ser escrita simplesmente,

$$S = \text{“ } A \text{ é } R \text{”}$$

$A = \{ A_1, \dots, A_n \}$  e  $R$  é um produto  $n$ -dimensional dos conjuntos difusos de  $R_1$  a  $R_n$ .

Considerando a última equação como um padrão de sintoma que corresponde a uma doença particular e, por outro lado, o dado da observação atual dado por “ $A_j$  é  $Q_j$ ” para cada  $A_j$ , então;

$$O = \text{“ } A \text{ é } Q \text{”}$$

$Q$  é o produto direto dos conjuntos difusos de  $Q_1$  a  $Q_n$ .

Para diagnosticar se uma doença em questão está ocorrendo ou não, considera-se a coexistência de duas proposições composicionais  $S$  e  $Q$ , de outra forma, considera-se o grau de conformidade de  $R$  e  $Q$ . O grau de conformidade de dois conjuntos difusos é expresso por  $\gamma$  e se resolve a seguinte equação:

$$\gamma = \sup_{u \in U} (R \cap Q) \quad (5.10)$$

$U = U_1 \times U_2 \times \dots \times U_n$  é o produto direto. Define-se o produto direto dos conjuntos difusos como o  $\min$  e  $\gamma_j$  é determinado por,

$$\gamma_j = \sup_{u_j \in U_j} (R_j \cap Q_j) \quad (5.11)$$

A equação (5.10) pode se expressa como segue:

$$\gamma = \min_j \{ \gamma_j \} \quad (5.12)$$

Em outras palavras  $\gamma$  é o valor mínimo fora de conformidade com  $R_j$  e  $Q_j$  ( $j = 1, 2, \dots, n$ )

### 5.8.3 Aplicação do engenheiro do conhecimento no diagnóstico

A utilização do engenheiro do conhecimento no diagnóstico médico já vem desde Shortliffe [SHO 76] quando ele publica seu sistema chamado MYCIN. Tem-se duas grandes partes do que é chamado de sistemas de engenharia do conhecimento:

- ◆ A base de conhecimento representado na forma “IF A THEN B”.
- ◆ O mecanismo de inferência que utiliza o conhecimento.

Nesse sistema o conhecimento do especialista é colocado na memória do computador e utiliza-o. Isto é chamado de sistema especialista.

Tem-se casos nos quais A e B da regra “ IF A THEN B” são proposições difusas e casos nos quais a veracidade da regra antes citada não é completa. K.P.Adlassing *et al.* desenvolveram um sistema chamado CADIAG na qual o diagnóstico é feito tendo em conta a ambigüidade, nesse encontram-se os seguintes quatro itens:

- S: sintomas e resultados de laboratório,  
 D: nome da doença ou diagnóstico,  
 SC: combinação dos sintomas,  
 IC: combinações internas

Se representamos  $\emptyset$  como desconhecido ou indeterminado e T é determinado pela seguinte equação:

$$T = [0, 1] \cup \emptyset$$

Cada sintoma  $S_i$  é caracterizado por  $\mu_{S_i}$ , que tem um valor em T, e seu valor é considerado como o grau com o qual o sintoma aparece. T pode assumir valores  $\emptyset$  contrastando como as funções de pertinência; e a interseção, união e complemento dos conjuntos difusos são definidos como segue:

$$X_1 \wedge X_2 = \begin{cases} \min\{x_1, x_2\} & \text{se } X_1 \in [0,1] \text{ e } X_2 \in [0,1] \\ \emptyset & \text{se } X_1 = \emptyset \text{ e/ou } X_2 = \emptyset \end{cases}$$

$$X_1 \vee X_2 = \begin{cases} \max\{x_1, x_2\} & \text{se } X_1 \in [0,1] \text{ e } X_2 \in [0,1] \\ X_1 & \text{se } X_1 \in [0,1] \text{ e } X_2 = \emptyset \\ X_2 & \text{se } X_1 = \emptyset \text{ e } X_2 \in [0,1] \\ \emptyset & \text{se } X_1 = \emptyset \text{ e/ou } X_2 = \emptyset \end{cases}$$

$$\overline{X_1} = \begin{cases} 1 - X_1 & \text{se } X_1 \in [0,1] \\ \emptyset & \text{se } X_1 = \emptyset \end{cases}$$

Considera-se as seguintes relações difusas entre os itens:

SD:	sintoma - doença
((CS)D):	combinação de sintomas - doença
SS:	sintomas - sintomas
DD:	doença - doença

A força dessa relação está dado pela frequência e força da confirmação. Por exemplo, para fazer a relação SD escreve-se a regra que relata a relação sintoma - doença na seguinte forma:

SE A ENTÃO B COM (O, C)

A letra A representa os sintomas e B o diagnóstico; os valores determinados para O e C são valores lingüísticos, mas as relações difusas são construídas utilizando os números representados pelos mesmos. Se os dados de um paciente são determinados, então é feita uma composição com a relação difusa e os dados difusos. De outra forma, utilizando as regras composicionais de Zadeh [ZAD 83], pode-se alcançar a confirmação ou exclusão da enfermidade.

### 5.9 Exemplos de sistemas especialistas na área medica que aplicam lógica difusa

Na tabela 5.1 apresenta-se o nome do sistema, o domínio da aplicação e a equipe que o desenvolveu, a sigla *sn* na tabela indica que o sistema não tem um nome específico.

Nome do Sistema	Domínio de Ação	Equipe
CADIAG - 2	medicina interna	[ ADL 85 ]
EMERGE	análise de dor do peito	[ HUD 88 ]
EXPERT	reumatologia, oftalmologia	[ WEI 81 ]
IRIA	diagnóstico de patologias tireóideas	[ SAN 76 ]
W.H.O	classificação de dislipoproteinemias	[ SAN 82 ]
Sistema de investigação prediagnóstica	detecção cedo de doenças em adultos	[ TAZ 86 ] [ TAZ 88a ]
Sistema de suporte reconhecimento medico geral	medicina do trabalho	[ TAZ 88b ] [ TAZ 88c ]
PNEUMON-IA	Diagnóstico e tratamento de pneumonias	[ GOD 89 ]
RENIOR	Diagnóstico reumatológico	[ SIE 94 ]
sn	Diagnóstico de estados fisiopatológicos em tempo real	[ ESC 94 ]
PreTherapy	Sistema especialista para o treinamento de estudantes	[ CHR 94 ]

Tabela 5.1. Exemplo de SE da área médica que utilizam lógica difusa.

## CAPITULO VI

### ESTUDO DA FERRAMENTA

#### 6 Estudo da ferramenta

##### 6.1 Introdução

A shell que será utilizada é KAPPA-PC 2.3 da IntelliCorp Inc.. Ela permite escrever sistemas especialistas num ambiente gráfico de alto nível e gerar códigos com padrão ANSI C e GUI.

Para construir sistemas baseados em modelos, precisa-se ferramentas para representar os objetos e os processos. Em KAPPA, classes e instâncias são utilizadas para representar os objetos. Programação orientada a objetos e regras são utilizadas para representar processos.

A programação orientada a objetos (POO) é utilizada para representar o comportamento de objetos individuais, e as regras para representar o raciocínio em forma de regras IF-THEN . Além disso, o KAPPA fornece ferramentas para desenvolver e personalizar a interface do usuário. Outra vantagem é que as ferramentas estão bem integradas, o que permite que o usuário aprenda o básico e vá enriquecendo os conceitos básicos com extensão e especialização das idéias. KAPPA - PC integra tecnologias como:

1. Programação orientada a objetos.
2. Linguagem de aplicação de alto nível.
3. Raciocínio baseado em base de regras.
4. Ambiente de desenvolvimento gráfico.

5. Ferramenta de apresentação gráfica orientada a objetos.
6. Interface com fontes de dados externos e outros software.

## 6.2 Comparando KAPPA-PC a processadores convencionais de dados

As bases de dados (BD) só podem armazenar números e fatos simples, enquanto KAPPA pode armazenar também inter-relacionamentos entre objetos, conceitos e processos.

As BD são quantitativas por natureza; KAPPA agrega um modo mais qualitativo de computação na forma de heurísticas.

Enquanto os programas convencionais atuam como caixas pretas, KAPPA é transparente, permitindo que o usuário observe as trocas no modelo e na seqüência de passos que conduzem a um troca.

## 6.3 Programação orientada a objetos (POO)

A POO está composta de uma coleção de estruturas de dados chamadas objetos. Cada objeto tem dois tipos básicos de informação: informação que descreve características do objeto e informações que especificam o que ele pode fazer.

As características do objeto são representado em KAPPA por *Slots* e as ações são representadas por métodos.

### 6.3.1 Características da POO

Entre as características mais importantes da POO tem-se duas que o tornam conveniente para desenvolver uma aplicação:

- ◆ Os objetos podem herdar métodos na mesma forma que herda *slots*, o que permite que seja diretamente herdado por outros objetos. Também permite sobrepor o comportamento que foi herdado, dando um comportamento diferente no objeto filho.
- ◆ A especialização, que permite uma interface uniforme para classes diferentes de objetos, permite uma uniformidade de interface que mantém oculta as diferenças procedurais.

### 6.3.2 Vantagens da POO

Algumas vantagens da POO são:

- ◆ Representação normal: A POO provê uma forma natural de representar o mundo real, criando entidades que têm comportamento. Isto é possível graças aos métodos, que são funções associadas aos objetos e que permitem uma representação mais clara do conhecimento.
- ◆ Modularidade: A modularidade permite que uma aplicação seja composta de partes ou módulos relativamente independentes. Então, ao modificar um módulo, não é necessário fazer qualquer troca em outro lugar.
- ◆ Reuzabilidade: Sendo que a aplicação está organizada em módulos, então eles podem ser reutilizados em outras aplicações.

### 6.4 Representação dos objetos em KAPPA

Os blocos de construção da base de dados em KAPPA-PC são os objetos que podem ser relacionados entre eles para capturar o relacionamento entre conceitos e coisas que eles representam.



### 6.4.1 Classes e instâncias

O mundo pode ser visto como uma coleção de objetos que têm certas características, partes, habilidades e/ou relações. Ex.: carro é um objeto que tem atributos como cor, preço, velocidade máxima, etc. e tem partes componentes como portas, motor, pneus, etc.. Os objetos em KAPPA são definidos como classes ou instâncias.

Classe é um objeto mais geral, podendo ser um grupo ou coleção de instâncias particulares. Instância é um objeto mais específico, sendo ele um item em particular ou um evento. Na figura 6.2 Carros é uma classe da qual CarroDeJoão é uma instância.

Subclasse é uma classe que é parte de outra. Ela é representada dentro de outra classe na hierarquia. Ex.: Sedã e Van são subclasses de Carros.

Classe Pai é a classe que está diretamente acima de uma subclasse na hierarquia. Por exemplo: na figura 6.2 Carros é a Classe Pai de Sedã.

Antecessor e descendente descrevem uma relação indireta entre classes, subclasses e instâncias. Ex.: Carros é o antecessor de CarroDeMaria, então CarroDeMaria é descendente de Carros. Como mostra a figura 6.2.

*Slots* representam todas as propriedades importantes de um objeto particular. Ele permite adicionar detalhe, estrutura, lista de atributos ou propriedades. Cada *slot* descreve uma característica do objeto. Ele permite também especificar a característica na descrição individual de cada objeto, quando é preenchido por valores. Em KAPPA-PC os *slots* apresentam uma série de opções que são:

- ◆ Especificação do número de valores que ele pode tomar;
- ◆ Especificação do tipo de valores que ele pode tomar;
- ◆ Especificação do conjunto de valores permitidos, e;
- ◆ Especificação de outras instâncias como possíveis valores.

Exemplo de *slots* da instância CarroDeMaria é mostrado na figura 6.1.

Instância:	CarroDeMaria	
	<b><u>Slots</u></b>	<b><u>Valor</u></b>
	Dono	Maria Oliveira
	Registrado	Não
	Modelo	1983
	Cor	Preto

Figura 6.1. Exemplo de *slots* contidos na instancia CarroDeMaria.

### 6.5 Vantagens em utilizar classes na representação de grupos de instâncias

- ◆ O raciocínio pode ser feito na classe considerando-a como um todo. Por exemplo, sabendo-se que um carro é pesado e vai rápido pode-se chegar à conclusão que é perigoso ser atropelado por este tipo de carro.
- ◆ Quando reconhece-se que um objeto é uma instância de uma classe, pode-se fazer afirmativas sobre ela sem conhecer detalhes particulares do objeto. Ex.: Mesmo que não tenha nunca visto um Sedã Chevy 1986, você pode assumir que ele é pesado, é rápido e é perigoso ser atropelado por ele, pois ele é membro de uma classe com a qual você está familiarizado.
- ◆ Quando aprende-se alguma coisa nova com respeito a uma classe, na verdade aprendeu-se algo sobre todos os membros da classe. Ex.: Se toma-se conhecimento que todos os carros têm cilindros, então pode-se assumir que o Chevy 1986 também tem um.

## 6.6 Organizando os objetos com herança

Tendo instâncias, classes e classes mais gerais, cria-se uma hierarquia de objetos nas quais as propriedades mais gerais são representadas só uma vez na classe mais apropriada.

As vantagens da herança são:

- Maior facilidade para criar as aplicações devido às propriedades gerais só precisarem ser definidas uma vez só.
- A aplicação é guardada e mantida numa forma mais eficiente, devido a que as propriedades gerais podem ser armazenados numa classe objeto, mas são aplicáveis a todos os descendentes automaticamente.

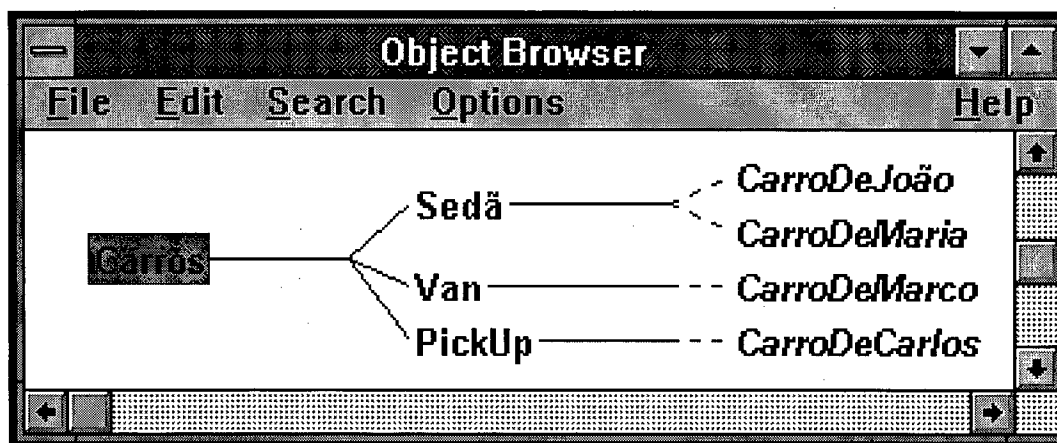


Figura 6.2. Representação de uma hierarquia, pelo *Object Browser* do KAPPA, na qual estão representadas as classes e instâncias.

## 6.7 Representação dos processos

Em KAPPA-PC, a representação dos processo do domínio são representados em três modos diferentes, sendo eles em forma de métodos, funções e regras.

### 6.7.1 Métodos

Os métodos em KAPPA permitem potencializar os objetos de tal forma que eles possam representar o comportamento das coisas ao qual eles correspondem. Os comportamentos dos *slots* são especificados por meio da criação de métodos. Eles são escritos na linguagem de aplicação do KAPPA - KAL (*Kappa Application Language*), especificando como os objetos devem se comportar. Esta técnica para guardar objetos com comportamento é chamada de **programação orientada a objetos**. Uma das vantagens dos métodos é que a informação que ele provê não precisa ser armazenada num *slots*, mas simplesmente é fornecida quando dita informação é solicitada.

Exemplo: Poderia ser criado um método que adicionado à classe Carros calcule quantos litros por quilômetro consome um carro, tendo como dados a distância percorrida e os litros de combustíveis consumidos.

Os métodos em KAPPA podem ser herdados, e métodos locais podem substituir métodos herdados. Eles são ativados por mensagens enviados aos objetos que os contêm. É importante salientar que em KAPPA é possível definir também MONITORES, que são métodos que são ativados quando acontece uma troca no valor do *slot*.

Em KAPPA tem-se diferentes tipos de monitores, dependendo do momento que eles são ativados:

- ◆ Antes da troca do valor do *slot*.
- ◆ Quando é acessado o *slot*.
- ◆ Depois da troca do valor do *slot*, e
- ◆ Se fosse necessário.

### 6.7.2 Funções

As funções são outra maneira que KAPPA usa para representar os processos, apresentando aproximadamente 300 funções que permitem a manipulação da base de conhecimento. Estas funções variam de simples operadores numéricos como “+”, “-”, lógicas como *Let* e funções específicas complexas de KAPPA como *PostInputForm* e *ForwardChain*. É importante destacar que novas funções podem ser criadas utilizando KAL ou linguagem C.

Os grandes grupos de funções apresentados por KAPPA são:

- ◆ Funções de Acesso a base de dados;
- ◆ Funções para DDE;
- ◆ Funções para acesso de arquivos;
- ◆ Funções de acesso DLL;
- ◆ Funções para imagens;
- ◆ Funções de operações lógicas;
- ◆ Funções matemáticas;
- ◆ Funções de manipulação de *strings*, e;
- ◆ Funções para manipulação de janelas.

### 6.7.3 Regras

As regras são utilizadas por KAPPA para a representação das relações entre causas e efeitos. Elas determinam as condições para as quais ocorre uma ação particular ou inferência. KAPPA representa os fatos que estão envoltos nas premissas e as conclusões referem-se a valores determinados aos *slots* em um objeto. Exemplo:

```
IF      a ignição esta acesa, e o sistema elétrico está bom,  
        e as condições das velas estão boas,  
THEN   as velas dão faísca.
```

## 6.8 Regras versus programação convencional

A primeira impressão que dá ao examinar a estrutura das regras é que ela é a mesma que na programação convencional, mas a diferença é que as regras vêm com um mecanismo de inferência, que não é outra coisa que um programa especial para o gerenciamento das regras e aplicá-las quando é apropriado. Na programação convencional sempre deve ser especificado quando uma condição deve ser disparada.

Alem disso, as regras com variável são um modo mais eficiente de codificar as informações. Uma regra pode representar um número de objetos, porém na programação convencional devem ser representados cada objeto com uma entidade individual.

## 6.9 Tipos de raciocínio em KAPPA-PC

Existem dois tipos de raciocínio baseado em regras: o raciocínio para frente (*Forward Chaining*) e raciocínio para trás (*Backward Chaining*). A seguir, faremos uma pequena descrição destes tipo de raciocínio apresentado por KAPPA.

### 6.9.1 Raciocínio para frente (*Forward Chaining*)

No raciocínio para frente sempre são avaliadas primeiro primeiro as premissas e se chega a uma conclusão que é o conseqüente da regra. Ele sempre é desencadeado pela entrada de um fato novo na aplicação e quando uma nova conclusão é alcançado ela pode disparar outra regra. Exemplo do raciocínio para frente é a seguinte. Se temos duas regras:

R01

IF     se as velas estão boas,  
       e a regulação do tempo está sincronizada,  
       e a bateria está carregada

THEN o sistema elétrico esta bom.

R02

IF se as velas estão ruins,  
e a regulação do tempo esta fora de sincronismo,  
e a bateria está baixa

THEN o sistema elétrico está ruim.

Assuma-se que os *slots* Condição da Vela tem o valor boa e regulação de tempo está em sincronia e coloca-se um novo fato na qual afirma-se que a bateria do CarroDeMaria está carregada na base de fatos. Então o mecanismo de inferência de KAPPA detecta que uma premissa da regra R0I encaixa com este novo fato, ele checa as outras duas premissas e aplica a regra que coloca a conclusão boa no *slot* Sistema Elétrico.

### 6.9.2 Raciocínio para trás (*Backward Chaining*)

No raciocínio para trás inicia-se perguntado ao mecanismo de inferência se um fato pode ser estabelecido. Ex.: Pode-se perguntar se o Sistema Elétrico do CarroDeMaria está bom.

O objetivo neste caso é detectar a regra cuja conclusão coincida com a pergunta, e dita pergunta é representada em forma de meta (*goal*), por isso é que este tipo de inferência é chamada de raciocínio dirigido por metas.

Uma vez feita a pergunta meta, o *Backward Chaining* inicia colocando na agenda de regras todas as regras cujas conclusões coincidem com a meta e analisa as premissas dos mesmos. No caso em que um dos valores da premissa da regra é desconhecido, o mecanismo de inferência pergunta ao usuário por dito valor, e se o mesmo coincide com o valor da regra, então a meta é afirmada. No exemplo anterior seria o preenchimento do *slot* Sistema Elétrico com bom ou ruim, dependendo da regra que foi aplicada.

### 6.10 Escolhendo entre raciocínio para frente ou raciocínio para trás

Geralmente o raciocínio para frente faz mais sentido de ser utilizado quando queremos introduzir novos fatos e achar suas conseqüências. Isto é típico em simulações: se as condições da bateria mudam, precisa-se saber qual será a sua influência na potência das luzes, sistema de arranque etc..

Quando informações específicas são necessárias, é mais apropriado a utilização de um raciocínio mais direto. Aqui é quando o raciocínio para trás é utilizado, freqüentemente, em diagnóstico, estabelecendo por exemplo uma *meta* que, no exemplo, anterior poderia ser “ A bateria do carro esta ruim”.

### 6.11 Determinando quando regras devem ser utilizados

Se o processo de raciocínio requer poucas condições, mas chama a uma serie de passos predeterminados, então neste caso as regras não são propícias.

As regras são apropriadas se tem-se uma série de condições que devem ser tratadas, se tais condições podem ser quebradas em regras menores e se o mecanismo de controle provido pelo sistema de inferência é o adequado.

Se é necessário micro-gerenciar a seqüência de eventos, então é melhor programar a estrutura detalhada de controle.

As regras são mais fáceis de entender do que o código de um programa, pois ele é escrito com uma sintaxes mais estruturada.



Finalmente, outra vantagem das regras é que em KAPPA foi introduzido ferramentas especializadas de busca de erros e explicação dos resultados, tanto para o programador, como para o usuário final.

### 6.12 Considerações importantes na escrita de regras em KAPPA

A seguir tem-se considerações que devem-se fazer quando se utiliza regras com KAPPA-PC.

- ◆ Determinar as regras que são consideradas relevantes, em que ordem e em que momento devem ser postas na lista de regras.
- ◆ Determinar os pares de Objeto:Slots e determinar a ordem em que estes aparecem na agenda para ser processado pelo raciocínio para frente.
- ◆ Determinar se todas as regras da base de conhecimentos serão aplicadas ou um subconjunto delas.
- ◆ Determinar se o processo de raciocínio será executado até o final ou será detido num determinado ponto, p. ex.: quando um determinado Objeto:Slot foi acessado ou quando uma regra será disparada.
- ◆ Se é utilizado o raciocínio para frente, determine que modelo de raciocínio será utilizado: seletivo, busca em profundidade, busca em amplitude ou *best-first*.
- ◆ Se duas regras criam conflitos, qual é a regra que deseja aplicar. Faça esta decisão estabelecendo a prioridade da regra.
- ◆ Se é utilizado o raciocínio para trás, determinar quais serão suas metas.

### 6.13 A agenda

A agenda é uma fila de pares de objeto:*slot* que serão processados pelo raciocínio para frente. Os *slots* cujos valores foram alterados como resultado da aplicação de uma regra são automaticamente agregados na agenda, podendo também ser utilizada a função *Assert* para agregar os pares objeto:*slot* na agenda. Quando é feito o raciocínio para frente existem dois modos de agenda em KAPPA-PC:

Modo IGNORE indica que um item na agenda será ignorado se existe uma versão mais recente do mesmo.

Modo NOIGNORE significa que um item na agenda será processado sem importar se ele ocorre de novo na agenda.

### 6.14 Resolução de conflitos no raciocínio para frente

Em KAPPA existem quatro diferentes estratégias para determinar que regra será testada primeiro, quando mais de uma regra se encaixa com um item na agenda, e apresenta-se os mesmos em forma resumida.

#### 6.14.1 Avaliação seletiva

Esta estratégia é a mais eficiente devido a que segue só um caminho de raciocínio. Durante o raciocínio, novas regras identificadas como importantes são adicionados na lista de regras de acordo com sua prioridade. Quando uma das regras é aplicada, então o resto delas na lista são removidas antes de considerar o seguinte item na agenda.

#### **6.14.2 Avaliação em profundidade (*Depth First*)**

É uma estratégia exaustiva que cobre todos as possíveis implicações do novo par objeto:*slot*. Difere do anterior pois ele não limpa a lista de regras quando uma regra é avaliada como verdadeira. Neste tipo de avaliação, as novas regras que podem ser aplicáveis são ordenadas, dependendo de sua prioridade, e colocadas no início da lista de regras.

No caso em que existam itens na lista de regra e na agenda, a prioridade é a de avaliar o próximo item da agenda.

#### **6.14.3 Avaliação em amplitude (*Breadth First*)**

Ele é também um método exaustivo. Durante o raciocínio neste modo as novas regras identificadas como aplicáveis são ordenadas de acordo com sua prioridade e então colocados no final da lista de regras.

No caso que exista itens tanto na agenda como na lista de regras a prioridade é a aplicação do seguinte item na lista de regras.

#### **6.14.4 Avaliação *Best First***

É similar à estratégia de avaliação em profundidade, sendo portanto também exaustivo. Nela as novas regras que aparecem como aplicáveis são misturadas com as regras que já estão na lista de regras dependendo de sua prioridade.

Se existem itens tanto na agenda como na lista de regras a prioridade é o de avaliar o próximo item da agenda.

## 6.15 Ferramentas do desenvolvedor de sistemas

O ambiente de desenvolvimento em KAPPA consiste em varias janelas que tem ferramentas para desenvolver uma aplicação, como mostra a figura 6.3.

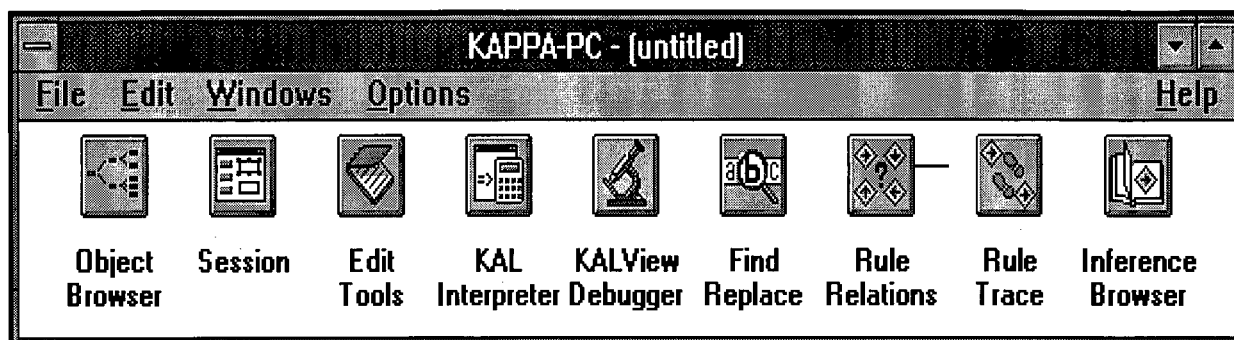


Figura 6.3. Janela principal do KAPPA-PC.

As ferramentas de desenvolvimento são:



**Object Browser** (Examinador de Objetos): Ele apresenta uma representação gráfica da hierarquia de objetos permitindo a modificação dos objetos e suas relações. A hierarquia é representada da direita à esquerda, representado por todas as subclasses da raiz. As instâncias, como não podem ter outras instâncias ou subclasses, representam o final da hierarquia.



**Session** (Sessão): A janela de seções é a interface principal para o usuário final da aplicação em KAPPA. Ela permite que o desenvolvedor da aplicação personalize as janelas com suas próprias opções gráficas criando, uma interface que simplifique a interação da aplicação com o usuário final.

A janela de seções apresenta dois modos, o de *Layout* que manipula as imagens gráficas e o de *runtime* que é o modo de apresentação da interface ao usuário final.



**Editors** (Editores): O KAPPA provê ao desenvolvedor do sistema sete editores:

1. Editor de classe;
2. Editor de instância;
3. Editor de *Slot*;
4. Editor de Métodos;
5. Editor de Funções;
6. Editor de Regras, e;
7. Editor de Metas.

A figura 6.4 mostra como é apresentado a janela que contém os editores no KAPPA. Eles podem ser utilizados para modificar os objetos e seus *slot* e métodos associados, nele são escritas as regras com o editor de regras, as funções com o editor de funções e o as metas com o editor de metas. Em todos os editores é utilizado o linguagem KAL, o que facilita o trabalho de desenvolvedor do sistema, pois a sintaxe para todos eles é a mesma.

Outra vantagem dos editores é que antes de salvar seu conteúdo ele detecta automaticamente erro na escrita de comandos ou detectam as chaves ou as vírgulas que não foram colocadas. Isto é muito importante pois evita ter de voltar a editar um método por erros de sintaxe no momento da execução.

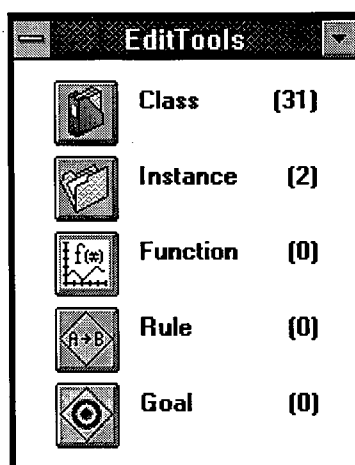


Figura 6.4. Ferramentas de edição do KAPPA-PC.



**KAL Interpreter Window** (Janela Interpretadora do KAL): Esta janela permite a escrita e interpretação de expressões em KAL. Ele tem dois objetivos:

- ◆ Um ambiente de desenvolvimento alternativo ao editor de conhecimento e o Browser de objetos.
- ◆ Um mecanismo para testar as expressões em KAL por meio de sua avaliação.

Exemplo: Pode-se escrever a seguinte expressão no KAL Interpreter para criar uma classe:

```
=>MakeClass(Autos, Root);  
Autos
```



**KAL View Debugger**: Ele permite detectar erros nos códigos das funções ou métodos. A janela de KAL View Debugger esta dividido em três:

- ◆ A janela do código fonte, que mostra o código da função ou método que está sendo examinado.
- ◆ A janela de valores, que mostra os valores de cada elemento quando é feita a execução passo a passo.
- ◆ A janela de *Watches*, que mostra os valores de várias variáveis ou *slots* que estão sendo observados.



**Find/Replace (encontrar/substituir):** Ele permite encontrar e substituir átomos que aparecem em qualquer lugar dentro do programa. Todas as ocorrências do átomo são mostrados na janela.



**Rule Relation Windows (Janela de Relações de Regras):** Provê uma representação gráfica das relações entre as regras que é muito importante para o encadeamento das regras. Nesta janela deve-se determinar em que regra esta interessado e as outras regras cujas conclusões são premissas desta regra são mostradas como dependências IF; regras cujas premissas são a conclusão da regra especificada são indicadas como dependências THEN.



**Rule Trace Windows (Janela de Seguimento de Regras):** Ele permite observar a regra que o mecanismo de inferência está chamando, permitindo um seguimento do impacto do raciocínio num *slot* particular da base de conhecimento.

Neste seguimento é possível observar como o sistema gera novas conclusões e permite detectar erros na base de conhecimento da aplicação. O mesmo permite fazer um seguimento ao processo no raciocínio, seja este raciocínio para frente ou raciocínio para trás.



**Inference Browser (Exame da Inferência):** Ele permite observar a regra que o mecanismo de inferência chama em forma gráfica. Nela pode-se observar como o sistema alcança uma conclusão examinando as linhas de raciocínio uma vez que o processo de raciocínio está completo. Ele permite detectar as fontes de erros na base de conhecimento da aplicação.

Destas ferramentas que KAPPA provê, as que estão relacionadas diretamente com a manipulação de regras são:

- ◆ Editor de regras
- ◆ Janela de relação de regras
- ◆ Janela de seguimento de regras
- ◆ Editor de metas (*goal*)
- ◆ O *browser* de inferência

### 6.16 Utilizando KAL, a Linguagem de aplicação de KAPPA-PC

KAL é a linguagem utilizado em KAPPA para escrever as regras, métodos e funções. Os nomes em inglês nesta linguagem fazem lembrar muito o que eles fazem. Exemplo para obter a lista de instâncias que fazem parte de uma classe, simplesmente utiliza a função *GetInstanceList(NOME DA CLASSE)* ou para criar um *slot* é só escrever *MakeSlot(NOME DO NOVO SLOT)* e pronto, isto facilita para o desenvolvedor do sistema ir aprendendo rapidamente esta linguagem.

O KAL permite fazer o seguinte:

- ◆ Colocar novas informações na base de conhecimento, criando objetos, *slots*, colocando novos valores aos *slot*, criar vínculos entre os objetos etc.
- ◆ Excluir informações da base de conhecimento, excluindo valores de um *slot* ou eliminado vínculos entre objetos.
- ◆ Fazer questionamentos sobre as informações armazenadas na base de conhecimento, recuperando valores dos *slot*, ou recuperando os antecessores de uma classe ou instância, etc.



### 6.16.1 Sintaxe do KAL

A sintaxe do KAL está dividida em seis tipos principais:

1. **Átomos:** uma palavra só ou um grupo de palavras entre aspas. Ex.: CarroDeMaria ou “roxo branco preto”.
2. **Pares:** Um nome de um objeto e um *slot* separados por dois pontos. Ex.: CarroDeMaria:Color.
3. **Expressões binárias:** Eles são utilizadas para testar e modificar valores numéricos, lógicos ou valores de string. Exemplo destas expressões são +, -, ≤, >, And, Or etc.
4. **Expressões especiais:** Elas são utilizadas por funções e tipicamente associadas com expressões que iniciam com o nome da função seguida de parâmetros específicos e acabando com uma expressão. Ex.: For, ForAll, While, etc.
5. **Expressões em Bloco:** São grupo de expressões separados por ponto e virgula que iniciam com uma chave aberta e terminam com uma chave fechada seguido de ponto e virgula. Ex.: { OpenWriteFile(arquivo.txt); WriteLine(“Dados dos Clientes”); CloseWriteFile(); };
6. **Chamada de Funções:** Como as ferramentas mais poderosas que o KAL tem, elas criam, acessam e modificam a base de conhecimentos, manipulam listas, arquivos, controlam a interface gráfica do usuário final etc. A sintaxe das funções são o nome delas seguidas dos argumentos entre parêntesis. Ex.: GetSlotList(Carros, Global:Temp).

### 5.17 Utilizando dados de origem externa

KAPPA suporta uma variedade de técnicas para interagir com outros software, permite importar dados de diferentes origens externas diretamente na aplicação, podendo-se trazer registros específicos de uma base de dados ou de uma planilha.

Qualquer arquivo que pode ser convertido em ASCII pode ser interpretada em KAPPA mediante a utilização de funções predefinidas.

Pode-se também executar outros programas utilizando a função *Execute*, que toma como argumento texto ou expressões. Ex.: `Execute("C:\WINDOWS\Write.exe");`.

Os dados de Planilhas e Base de dados de Excel, Lotus 1-2-3, dBASE III, dBASE III Plus, dBASE IV podem ser acessadas e modificadas por KAPPA-PC.

E por último a característica de troca dinâmica de dados do Windows DDE (*Dinamic Data Exchange*) podem ser utilizados em KAPPA, permitindo que ele envie e receba dados de outras aplicações. KAPPA prove funções KAL para diretamente registrar e chamar funções com DLL (*Dinamic Link Libraries*), o que é muito útil para se desenvolver funções genéricas com C que logo serão chamados pelo KAPPA sem necessidade de modificar os arquivos executáveis dele.

## CAPÍTULO VII

### APLICAÇÃO DE UM SISTEMA ESPECIALISTA DIFUSO PARA O DIAGNÓSTICO DE HEPATOPATIAS CRÔNICAS

#### 7.1 Introdução

No mundo real existem muitos conhecimentos difusos, isto é, conhecimentos que por natureza são vagos, imprecisos, incertos, ambíguos ou inexatos. O pensamento e o raciocínio humano envolvem freqüentemente informações difusas, possivelmente originadas de conceitos humanos inerentes a experiências imprecisas, ao invés de experiências exatas.

Em sistemas baseadas na teoria clássica de conjuntos e na lógica bivalorada, é muito difícil de responder algumas questões, especialmente quando elas não são inteiramente certas. Então, os sistemas especialistas devem ser capazes de enfrentar os problemas com informações imprecisas, incompletas e com diferentes opiniões de especialistas, e isto é possível mediante a utilização da lógica difusa, que permite a utilização de uma linguagem natural ambígua nas regras, de forma a expressar os conhecimentos adquiridos dos especialistas da mesma forma que este está acostumado [TER94].

Na área de Gastro-Enterologia, os sistemas especialistas têm sido desenvolvidos principalmente para facilitar o diagnóstico, como é o caso da primeira aplicação nesta área que data de trinta anos atrás [RIM63], que fazia uma análise matemática para tentar identificar doenças epigástricas, e para prover diretrizes na tomada de decisão na prática clínica e no tratamento [LUC89].

Neste capítulo, foi desenvolvida uma aplicação que possibilita a compreensão de como os sistemas especialistas difusos podem ser aplicados em problemas reais dentro da área médica, especificamente na área de Gastro-Enterologia. O problema escolhido é o do diagnóstico de

hepatopatias crônicas. A solução é o desenvolvimento de um sistema especialista difuso que possa assistir ao diagnóstico deste tipo de doenças, orientado principalmente para o treinamento de pessoal médico, para-médico e enfermeiras.

## 7.2 Descrição do Problema

O fígado é o maior sintetizador de enzimas e outros compostos do organismo, e tem participação ativa na desintoxicação do mesmo. Um mau funcionamento do fígado produz uma série de alterações no organismo, que quando aparecem, são relacionados com uma possível disfunção hepática devido a algum tipo de agressão.

Estas agressões podem ser agudas ou crônicas. A primeira é uma agressão dos hepatócitos, que são as células componentes do fígado, por algum agente que evolui muito rápido. Agressões crônicas ocorrem quando a agressão aos hepatócitos é mais leve e a inflamação hepática não é resolvida no prazo de seis meses.

Dentro da especialidade de Gastro-Enterologia temos o estudo dos distúrbios que acontecem com o fígado e o sistema biliar, que são considerados como doenças individuais que devem ser descritas pelas mudanças funcionais e morfológicas que eles produzem.

No gerenciamento clínico de um paciente com doença do fígado, é importante avaliar o grau de distúrbio dos quatro componentes funcionais e morfológicos do fígado: as células hepáticas, o sistema vascular, os dutos biliares e o sistema retículo endotelial. Para isto, nos métodos clínicos e laboratoriais de avaliação, cada um destes componentes deve ser considerado. Neste caso o problema será delimitado ao estudo da detecção de Hepatopatias Crônicas e o diagnóstico de algumas das etiologias das mesmas.

No exame clínico, o médico passa a identificar o paciente, para fazer logo a anamnese onde o paciente conta as queixas tais como dor, febre, distúrbios mentais, hemorragias etc.. Logo estuda-se os antecedentes do paciente, por exemplo, tenta-se saber se o paciente consome

drogas hepatotóxicas, utiliza drogas injetáveis, e se ele ingere álcool em doses cirrotizantes. Depois é feito o exame físico no qual tenta-se detectar sinais de desnutrição, ascite, icterícia, pressão arterial, hálito, presença de estigmas, ginecomastia, hepatomegalia etc. Essas informações auxiliam no fornecimento de pistas diagnósticas ao médico, da possível presença ou não de doença hepática. Porém, como se pode observar, os valores utilizados são geralmente vagos e incompletos. Cada um dos dados mencionados deve ser estruturado e montado para orientar as suspeitas de uma hepatopatia crônica ou alguma outra doença em particular.

Os exames laboratoriais são importantes para tentar detectar doenças, para direcionar o processo de diagnóstico, estimar a severidade de uma doença, calcular prognóstico e avaliar a terapia. Na avaliação da funcionalidade hepática existe uma série de estudos laboratoriais que devem ser feitos para tentar demonstrar deficiências bioquímicas, mostradas na tabela 7.1. Os valores normais desta tabela são difusos, pois o limite entre o normal e anormal vai se deteriorando lentamente e não em forma absoluta, dependendo do contexto da situação.

Os estudos laboratoriais podem também ajudar a determinar a etiologia de uma doença e isto é especialmente certo na lesão do fígado por álcool. Deve-se sempre ter em conta que os dados clínicos junto com os dados laboratoriais são os que determinam a etiologia. A interpretação dos dados laboratoriais é muito difícil se eles não são interpretados no contexto da clínica, e no caso de se proceder assim até poderia conduzir a erro no diagnóstico.

### **7.3 Solução proposta**

Devido à complexidade que envolve o diagnóstico médico de hepatopatias crônicas, e à ambigüidade que está presente nela, tanto para determinar os valores normais ou anormais, como na interpretação de dados clínicos e laboratoriais, propõe-se o desenvolvimento de um SISTEMA ESPECIALISTA DIFUSO para *Windows* que capte a ambigüidades do diagnóstico mediante a expressão das mesmas em forma de variáveis lingüísticas, utilizando a lógica difusa, de tal forma a assistir no diagnóstico aos profissionais da saúde.

Teste	Valor Normal	Possível Interpretação
<b>Laboratorial</b>		
<b>Bilirrubinas</b>		
Total	5-17 $\mu\text{mol}$	Diagnostica icterícia. Avalia a severidade da
Conjugada	< 5 $\mu\text{mol}$	doença de Gilbert, hemolises.
Fosfatase Alcalina	35 - 130 iu/l	Diagnostica colestases, infiltrações hepáticas
GOT	5 - 40 iu/l	Diagnóstico precoce de doenças hepatocelulares, seguimento de progressos
GPT	5 - 35 iu/l	Esta ligeiramente embaixo da GOT em etiologias alcóolicas.
$\gamma$ GT	10 - 48 iu/l	Diagnostica abuso alcóolico, marcador biliar de colestases.
Albumina	35 - 50 g/l	Avalia a severidade
$\gamma$ Globulina	5 - 15 g/l	Assiste no diagnóstico de hepatites crônica e cirroses
PT	12 - 16 seg.	Avalia severidade

Figura 7.1. Estudos laboratoriais essenciais para o diagnóstico de doenças Hepato-Biliares.

Na aplicação foi considerada a utilização da *Shell* para o desenvolvimento de sistemas especialistas o KAPPA-PC 2.3 da IntelliCorp Inc.. Aproveitando-se as ferramentas que ele tem para fazer uma *interface* com o usuário num ambiente gráfico de alto nível, as características de programação orientada a objetos, e o raciocínio baseado em regras. A inclusão da lógica difusa foi feita mediante a implementação de funções e métodos para a definição dos conjuntos difusos, fuzzificação (ver Anexo I), e controle do mecanismo de inferência, incorporando-se uma base de conhecimento.

### 7.3.1 Função de fuzzificação

Na função de fuzzificação foram escolhidas as funções de pertinência dos estudos laboratoriais que estão definidos por quatro parâmetros **a**, **b**, **c** e **d**, da seguinte forma:

$$\mu_j(x) \stackrel{\text{def}}{=} \begin{cases} \frac{x - a_j}{b_j - a_j} & \text{se } x \in [a_j, b_j] \\ 1 & \text{se } x \in [b_j, c_j] \\ \frac{d_j - x}{d_j - c_j} & \text{se } x \in [c_j, d_j] \\ 0 & \text{caso contrario} \end{cases}$$

Na figura 7.1 é representada em forma gráfica a função de pertinência como os quatro parâmetros.

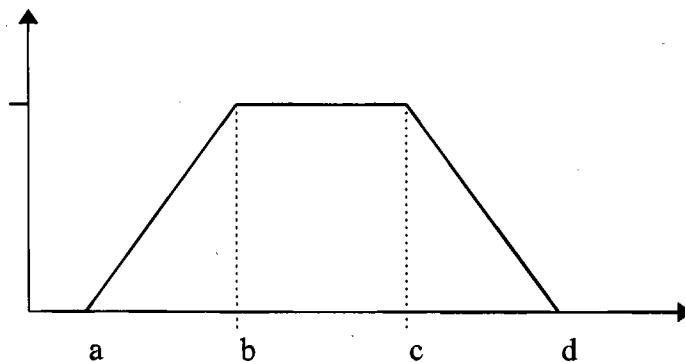


Figura 7.1 Representação da função de pertinência e seus quatro parâmetros.

### 7.3.2 Definição dos conjuntos difusos

As variáveis lingüísticas definidas para os estudos laboratoriais foram as seguintes:

TGO (transaminase glutâmico-oxalacética), TGP (transaminase glutâmico-pirúvica), GGT (gama-glutamyltranspeptidase), FAlc (fosfatase alcalina), BT (bilirrubina total), BD (bilirrubina direta), BI (bilirrubina indireta), TAP (tempo de protrombina), ALB (albumina), ALFA1, ALFA2, BETA, GG (gama globulina).

O universo do discurso sobre o qual os conjuntos difusos foram definidos, foi obtido através de consulta a especialistas, chegando-se a valores consensuais para cada um dos estudos laboratoriais. Os conjuntos difusos que foram adotados para eles, também constituem termos que os médicos especialistas na área estão acostumados a utilizar na hora de interpretar os estudos laboratoriais, e são os seguintes:

Muito elevado	MUE
Moderadamente elevado	MOE
Ligeiramente elevado	LIE
Normal	NOR
Ligeiramente baixo	LIB
Moderadamente baixo	MOB
Muito baixo	MUB

Na figura 7.2 descreve-se os conjuntos difusos da variável lingüística TGO, de tal forma a ilustrar como é que as variáveis lingüísticas laboratoriais estão definidas.



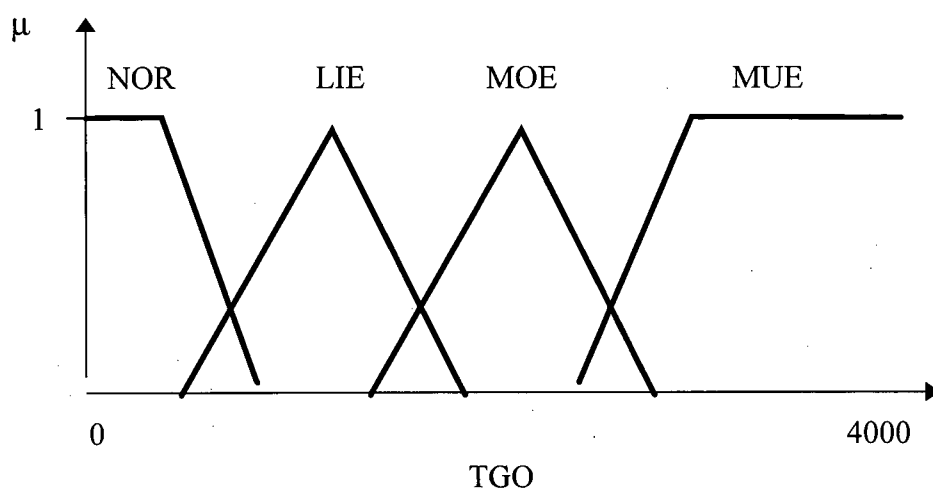


Figura 7.2 Conjuntos difusos definidos para a variável linguística TGO.

### 7.3.3 Base de Conhecimento

A base de conhecimento no sistema ARANDÚ é formada por fatos, regras e módulos. Os fatos são os objetos mais elementares do sistema e eles são de três diferentes tipos: Booleanos (sim, não), difusos (variáveis linguísticas com termos linguísticos), numéricos.

O conjunto de regras, selecionado para formar a base de conhecimento do sistema especialista, é do tipo *if-then* que relaciona dados clínicos e laboratoriais para diagnosticar as hepatopatias crônicas e suas possíveis etiologias. As regras foram adquiridos dos especialistas na área de Gastro-Enterologia do Hospital Universitário da Universidade Federal de Santa Catarina.

Os módulos são agrupamentos de regras, que facilitam a correção e atualização da base de regras. Na formação dos módulos foram adotados critérios, como regras com a mesma conclusão, regras com antecedentes similares, regras para interpretar determinados estudos de laboratório (funcional hepático, eletroforese) etc.

Para a avaliação das regras são utilizados conjuntos difusos, obtidos mediante o processo de fuzzificação, nos antecedentes e conseqüentes das regras. Quando uma regra é disparada, assinala-se à conclusão da mesma um valor de certeza lingüístico que está definida na tabela 7.1. A figura 7.3 mostra a função de pertinência do valor de certeza lingüístico.

VALOR DE CERTEZA LINGÜÍSTICA	ABREVIATURA	INTERVALO
Impossível	I	[0.00 , 0.08]
Quase impossível	QI	[0.05 , 0.20]
Ligeiramente possível	LP	[0.15 , 0.35]
Moderadamente possível	MOP	[0.25 , 0.50]
Possível	P	[0.45 , 0.70]
Muito possível	MUP	[0.65 , 0.85]
Quase Certeza	QC	[0.80 , 0.95]
Certeza	C	[0.90 , 1.00]

Tabela 7.1. Valores de certeza lingüísticos utilizados e os intervalos dos mesmos.

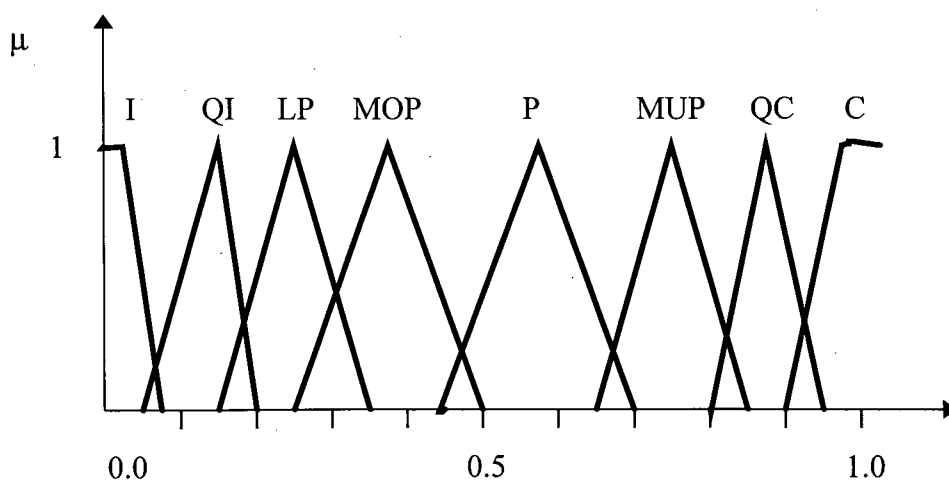


Figura 7.3 Função de pertinência do valor de certeza lingüístico.

A seguir apresenta-se alguns exemplos de regras que estão contidos na base de conhecimento:

EH02

If Bradipsiquismo = PRESENTE AND Asterixis = PRESENTE  
AND Hálito = HEPÁTICO

Then [QUASE CERTEZA] A presença de uma ENCEFALOPATIA HEPÁTICA

ES03

If Eritema = PRESENTE AND Nevus = PRESENTE  
AND Ginecomastia = AUSENTE

Then [QUASE CERTEZA] A presença de ESTIGMAS

HC07

If Encefalopatia Hepática = QC AND Hipertensão Portal = MUP  
AND Ascite = MODERADA AND Hepatomegalia = ACENTUADA  
AND Estigmas = MUP

Then [MUITO POSSÍVEL]  
A presença de uma HEPATOPATIA CRÔNICA DESCOMPENSADA,  
fazer as provas da função hepática.

FH01

If TGO = MOE AND TGP = MOE AND GGT = MUE  
AND BT = LIE AND TAP = MOB

Then [POSSÍVEL]  
Presença de HEPATOPATIA

#### **7.3.4 Interface com o Usuário**

A *interface* com o usuário esta projetada de tal forma que as telas se apresentam numa ordem natural, como os especialistas estão acostumados a considerar, de modo ergonômico por

facilitar a interação homem-máquina. As informações que devem ser introduzidas, aproximadamente trinta questões, foram escolhidas pelos especialistas como as mais fundamentais para o diagnóstico, em alguns casos já tem uma lista de opções de onde a usuário do sistema ARANDÚ vai selecionando com o *mouse*, de modo a eliminar possíveis erros de digitação na hora de preencher as informações. Nas telas também são incluídas imagens digitalizadas de alguns sintomas de modo que possa ser consultado pelo usuário na hora de utilizar o sistema.

As telas apresentadas para entrada de dados são as seguintes: histórico do paciente, anamnese e estudos laboratoriais. A figura 7.4 mostra a tela de anamnese, com as informações que devem ser preenchidas.

ANAMNESE	
Ictericia	ACENTUADA
Nevus Aracneiformes	PRESENTE
Ascite	LEVE
Asterixis	AUSENTE
O Fígado é Palpável ?	ACENTUADA
Hálito	AUSENTE
Esplenomegalia	PRESENTE
Ginecomastia	PRESENTE
Eritema Palamar	AUSENTE
Volumem da Parótida Aumentada?	NAO

**IMAGENS**

- Eritema Palmar
- Nevus Aracneiformes
- Esquema Nervus
- Ginecomastia

**ANAMNESE CONTINUAÇÃO**

Figura 7.4. Tela de Anamnese para entrada de dados no sistema ARANDÚ.

#### 7.4 Resultados Obtidos

Quando da implementação do sistema foram apresentados a ele casos de pacientes com hepatopatias crônicas, que foram selecionados das fichas da Clínica Médica do Hospital Universitário (HU), e casos hipotéticos sugeridos pelos médicos especialistas na área, analisando-se um total de vinte casos.

Foi possível observar que a interação do usuário (o especialista) e o sistema para a entrada de dados foi muito amigável devido à estruturação da *interface* homem-computador.

Com relação às etiologias que o sistema diagnostica, que são: viral (B e C), alcoólica, drogas, metabólicas (doença de Wilson e Hemocromatose) e idiopáticas, observa-se que em 75% dos casos (15 amostras) o sistema fez o mesmo diagnóstico dos especialistas e nos 25% restantes (5 amostras) o sistema deu erro ou não chegou a um diagnóstico da etiologia. Destaca-se que dos dados apresentados, a grande maioria deles se deviam a etiologia alcoólica, por serem os casos mais frequentemente atendidos no HU.

Uma das facilidades que foi implementado no sistema, foi a de salvar os registros das consultas numa planilha *EXCEL*, de forma a ter uma base de dados dos pacientes que poderia ser consultada, posteriormente, para fazer um acompanhamento do mesmo. Isto é muito importante com certos dados laboratoriais, porque permite verificar se a doença está evoluindo positivamente ou não.

Na parte explicativa do sistema, ele apresenta comentários dos diagnósticos emitidos além de fazer sugestões dos estudos laboratoriais e métodos operacionais a serem feitos para confirmar o mesmo. Sendo que o sistema está direcionado para o ensino de estudantes e recém formados em medicina e outros profissionais da saúde, as imagens digitalizadas dos sintomas característicos na clínica hepática permite assistir aos usuários na aquisição de novos conhecimentos.

## CAPITULO VIII

### CONCLUSÕES E RECOMENDAÇÕES

#### 8.1 Conclusões

A lógica difusa é a que permite modelar a difusividade e ambigüidade das informações e expressa-las lingüísticamente. Além disso, melhora a utilização dos conhecimentos representados na forma de regras, que são processadas por um mecanismo de inferência nos sistemas especialistas difusos.

Neste trabalho aborda-se um estudo dos sistemas especialistas difusos aplicados na área médica. Como exemplo de aplicação foi estudado o diagnóstico de hepatopatias crônicas dentro da especialidade de Gastro-Enterologia, na qual os fatos, nosologia e critérios de diagnóstico são caracterizadas por alto nível de incerteza associada.

Comprova-se, ademais, que mediante a aplicação da lógica difusa, a aquisição do conhecimento dos especialistas ficava facilitada, por serem utilizadas na representação do mesmo termos que eles estão acostumados a utilizar, o que permite também uma maior facilidade para o refinamento da base de conhecimento.

Observa-se também que, uma vez definidos os conjuntos difusos, pode-se passar imediatamente a utilizá-as na construção das regras, mesmo que ainda não se tenha definido muito bem os intervalos de cada conjunto. Isto foi principalmente observado na elaboração das regras de interpretação dos dados laboratoriais, onde foi montada no sistema a base de conhecimento e após, mediante consulta a especialistas, chegou-se a um consenso na definição dos intervalos apropriados para cada conjunto.

Para o desenvolvimento do sistema foi utilizado o KAPPA-PC 2.3 da *IntelliCorp Inc.* e foram utilizados métodos e funções para nele introduzir a lógica difusa.

No trabalho demonstra-se a aplicabilidade da lógica difusa para a elaboração dos sistemas especialistas, por diminuir a complexidade e tamanho da base de conhecimento, melhorando o tempo de resposta do mesmo, além das vantagens anteriormente mencionadas, atingindo-se assim os objetivos gerais propostos no início.

## 8.2 Recomendações

Recomenda-se para trabalhos futuros:

- ◆ Ampliar o número de etiologia existentes, além de introduzir o diagnóstico das complicações e a terapêutica, o que permitirá aumentar o espectro de doenças que podem ser diagnosticadas.
- ◆ Fazer um estudo comparativo do desempenho de um sistema especialista difuso cuja base de conhecimento foi adquirido mediante entrevistas com os especialistas e outra na qual as regras foram extraídas de uma rede neuronal difusa treinada com dados de pacientes.
- ◆ Estudar melhores métodos de avaliação dos sistemas especialistas aplicadas na área médica, para determinar as diferenças entre os especialistas humanos e eles, nos casos nos quais um diagnóstico preciso é dificilmente alcançado.

## Referências Bibliográficas:

[ADL85] ADLASSNIG, K.P., KOLARZ, G., and SCHEITHAUER, W., Present State of the Medical Expert System CADIAG-2, **Method of Information in Medicine**, v. 24, n. 1, p. 13 - 20, 1985.

[ADL95] ADLASSNIG, K., P., HORAK, W., Development and retrospective evaluation of HEPAXPERT-I: a routinely-used expert system for interpretive analysis of hepatitis A and B serologic findings, **Artificial Intelligence in Medicine**, v. 7, p. 1 - 24, 1995.

[AIK83a] AIKINS, J. S., Prototypical knowledge for expert systems, **Artificial Intelligence**, v. 20, p. 163 - 210, 1983.

[ALT84b] ALTY, J. L. and COOMBS, M. J., Associative and causal approaches to diagnosis - INTERNIST and CASNET, **Expert Systems, Concepts and Examples**, Manchester, England: NCC Publications, 1984.

[BAA84] BAAS, L. and BOURNE, J. R., A rule-based microcomputer system for electroencephalogram evaluation, **IEEE Transactions on Biomedical Engineering**, v. BME-31, n. 10, October 1984.

[BAR81] BARR, A., and FEIGENBAUM, E., A., **The Handbook of Artificial Intelligence**, Los Altos, CA, 1981.

[BEN84] BENNETT, J. S. and ENGELMORE, Robert S., Experience using EMYCIN, in BUCHANAN, B. and SHORTLIFFE, E., **Rule-Based Expert Systems**, Mass.: Addison-Wesley, 1984, p. 314 - 328.



- [BON92] BONFA, I., MAIOLI, C., et al., Development project for an integrate expert system in the prognosis of chronic liver disease., in LUN, K., C., et al. , **Medinfo 92**, IFIP, P. 579 - 583, 1992.
- [BON94] BONFA, I., MAIOLI, C., et al., HERMES: an Expert System for the Prognosis of Hepatic Diseases, **WCES'94**, 1994.
- [BUC84a] BUCHANAN, B. and SHORTLIFFE, E., The problem of evaluation, in BUCHANAN and SHORTLIFE, **Rule-Based Expert Systems**, Mass.: Addison-Wesley, 1984, p. 571 - 596.
- [BUC84c] BUCHANAN, B. and SHORTLIFFE, Use of MYCIN inference engine, in BUCHANAN and SHORTLIFFE, **Rule-Based Expert Systems**, Mass.: Addison-Wesley, 1984, p. 295 - 301.
- [CHR94] CRISTENSEN, Larry, HAYS, Bill and CLYDE, Stephen, PreTherapy: An Expert System for Training Studen Therapists, **WCES'94**, 1994.
- [DUB80] DUBOIS, D., PRADE, H., **Fuzzy Sets And Systems**, New York: Academic Press, 1980.
- [ESC94] ESCALADA-IMAZ, G., et al., Medical Reasoning Model for Diagnosing Physiopathological States in a Real Time Environment, **WCES'94**, 1994.
- [GAS83] GASTON, L W., LINDBERG, D. A. B., VANKER, A. D., et al., AI/COAG, A Knowledge-based surrogate for the human hemostasis expert, **Missouri Medicine**, v. 80, n. 4, p. 185 - 188, April 1983.

- [HAY83] HAYES-ROTH, Frederick, WATERMAN, D., A. and LENAT, D., **Building Expert Systems**, Mass.: Addison-Wesley Publishing Company, Inc.
- [INT94a] INTELLICORP, **KAPPA-PC User's Guide**, IntelliCorp, Inc., January 1994.
- [INT94b] INTELLICORP, **KAPPA-PC Reference Manual**, IntelliCorp, Inc., January 1994.
- [INT94c] INTELLICORP, **KAPPA-PC Advanced Topic**, IntelliCorp, Inc., January 1994.
- [JAM93] JAMSHIDI, M., **Fuzzy Logic and Control: Software and Hardware Applications**, Prentice Hall Series on Environmental and Intelligent Manufacturing Systems, v. 1, 1993.
- [KOS94] KOSKO, Bart, **Fuzzy Thinking The New Science of Fuzzy Logic**, Hyperion, New York, 1994.
- [KUL80a] KULIKOWSKI, C. A., Artificial intelligence methods and systems for medical consultation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. PAMI-2, n. 5, p. 464 -476, September 1980.
- [KUL82] KULIKOWSKI, C. A. and WEISS, S. M., Representation of expert knowledge for consultation: the CASNET and EXPERT projects, in SZOLOVITS, P., **Artificial Intelligence in Medicine**, Boulder, Colo.: Westview Press, p. 21 - 55, 1982.
- [LEE90] C. LEE, Fuzzy Logic in control systems: fuzzy logic controller, **IEEE Trans. on Systems, Man, and Cybernetics**, v. 20, P. 404 - 418, 1990.
- [LES83] LESMO, L. et al., Fuzzy Production Rules: a learning methodology, in WANG, P., P., **Advances in fuzzy set theory and applications**, Plenum Press, 1983.

- [LEU88] LEUNG, K., S., and LAM, W., Fuzzy Concepts in Expert Systems, **IEEE**, p. 43 - 56, September 1988.
- [LIN80a] LINDBERG, D. A. B., SHAP, G. C., KINGSLAND, L. C. et al., Computer based rheumatology consultant, **Proceedings of the Third World Conference on Medical Informatics**, p. 1311 - 1315, 1980.
- [LIN82] LINDBERG, D. A. B., KINGSLAND, L. C., ROESELER, G.C. et al., A new knowledge representation for diagnosis in rheumatology, **Proceedings of AMIA Congress 82**, New York: Masson Publishing Co., p. 299 - 303, 1982.
- [LUC89] LUCAS, P., et al., HEPAR: an expert system for the diagnosis of disorders of the liver and biliary tract, **Liver**, v. 9, p. 266 - 275, 1989.
- [MAM82] MANDARI, E., H., and EFSTATHIOU, J., Fuzzy Logic, **Proceedings ACM Symp. on Expert systems**, Brunel University, 1982.
- [MAR82] MARTIN-CLOUAIRE, R., **A computer-aided medical diagnosis method based on a fuzzy set theoretical approach**, MSc Thesis, University of Saskatchewan, 1982.
- [MIL83] MILLER, P.L., Medical plan-analysis: the ATTENDING system, **Proceedings IJCAI-83**, p. 239 - 241, 1983.
- [MIZ79] MIZUMOTO, M., FUKAMI, S. and TANAKA, K., Some Methods of Fuzzy Reasoning, in GUPTA, M., M., RAGADE, R., K. and YAGER, R., R., **Advances in Fuzzy Set Theory and Applications**, North-Holland, Amsterdam, p. 117 - 136, 1979.
- [PAT81b] PATIL, Ramesh, SZOLOVITS, Peter and SCHWARTZ, William, Causal understanding of patient illness in medical diagnosis, **Proceedings IJCAI-81**, P. 893 - 899, 1981.

- [PAT82c] PATIL, Ramesh, SZOLOVITS, Peter and SCHWARTZ, Willian, Modeling knowledge of the patient in acid-base and electrolyte disorders, in SZOLOVITS, P., **Artificial Intelligence in Medicine**, AAAS Symposium Series, Boulder, Colo.: Westview Press, p. 191 - 226, 1982.
- [PAT90] PATTERSON, D. W., **Introduction to artificial intelligence and expert systems**, Prentice-Hall, inc., 1990.
- [RIC93] RICH, Elaine, KNIGHT, Kevin, **Inteligência Artificial**, Segunda Edição, São Paulo: MAKRON Books do Brasil Editora Ltda., 1993.
- [ROC88] ROCHA, A., F., THEOTO, M. and TORASSO, P., Heuristic Learning Expert Systems, **Fuzzy Logic in Knowledge-Based Sys. Dec. and Control**, 1988.
- [SAN79] SANCHEZ, E., Medical Diagnosis and Composite Fuzzy Relations, in **Advances in Fuzzy Set Theory and Application**, GUPTA, M. M., et al., eds., North-Holland, Amsterdam, 1979, p. 437 - 444.
- [SAN82] SANCHEZ, E., et al., Linguistic Approach in Fuzzy Logic of W.H.O. Classification of Dyslipoproteinemias, in **Fuzzy Set and Possibility Theory - Recent Developments**, p. 582 - 588, Pergamon Press, 1982.
- [SHE93] SHERLOCK, Sheila and DOOLEY, James, **Deseases of the Liver and Biliary System**, Nith Edition, London: Blackwel Scientific Publications, 1993.
- [SHO79] SHORTLIFFE, E. H., BUCHANAN, B. G. and FEIGENBAUM, E. A., Knowledge engineering for medical decisionmaking: a review of computer-based clinical decision aids, **Proceedingas of the IEEE**, v. 67, n. 9, p. 1207 - 1224, September 1979.

- [SHO84] SHORTLIFFE, Edward H., CARLISLE, Scott A., BISCHOFF, Miriam B. et al., in BUCHANAN, B. and SHORTLIFFE, E., An expert system for oncology protocol management. **Rule-Based Expert Systems**, Reading, Mass.: Addison-Wesley, 1984.
- [SHO90] SHORTLIFFE, E., H., Clinical Decision-Support Systems, Medical Informatics, in **Computer Application in the Health Care**, Addison Wesley, p. 466 - 502, 1990.
- [SHO91] SHORTLIFFE E., H., Knowledge - Based Systems in Medicine, **Medical Informatics Europe 1991.**, Berlin, Springer - Verlag, p. 5 - 9, 1991.
- [SIM88] SIMONS, G., T., **Introdução a Inteligência Artificial**, Editora Classe, 1988.
- [SUG88] SUGENO, M. and KANG, G.T., Structure identification of fuzzy model, **Fuzzy Sets and Systems**, v.28, P. 15 - 33, 1988.
- [TAZ86] TAZAKI, E., et al., Medical Diagnosis Using Simplified Multidimensional Fuzzy Reasoning, **Proc. IEEE Int. Conf. on SMC**, 1988.
- [TAZ88a] TAZAKI, E., et al. Medical Data Base System with Inference System: Artificial Intelligence in Health Checkup Assessments, **Proceedings of the 7th Joint Conference on Medical Informatics**, p. 377 - 380, 1988.
- [TAZ88b] TAZAKI, E., et al. Medical Data Base System with Inference System: Artificial Intelligence in Health Risk Assessments for Health Maintenance Information, **Proceedingas of the 7th Joint Conference on Medical Informatics**, p. 385 - 388, 1988.
- [TER94] TERANO, Toshiro, ASAI, Kiyoji and SUGENO, Michio, **Fuzzy Systems Theory and its Applications**, London: Academic Press, Inc., 1994.

- [TSU79] TSUKAMOTO, Y., An approach to fuzzy reasoning metod. In Madan M. Gupta, Rammohan K., Ragade, and Ronald R. Yager, editors, **Advances in Fuzzy Set Theory and Applications**, P. 137 - 149, North - Holland, Amsterdam, 1979.
- [UCK93] UCKUN, S., DAWANT, B., M. and LINDSTROM, D. , P., Model-Based Diagnosis in Intensive Care Monitoring: the YAQ Approach, **Artificial Intelligence in Medicine**, v. 5, p. 31 - 48, 1993.
- [VIO93] VOIT, Greg, Fuzzy Logic in C, **Dr. Dobb's Journal**, p. 40 - 49, February 1993.
- [WAT86] WATERMAN, Donald, A., **A Guide to Expert Systems**, Mass.: Addison-Wesley Publishing Company, 1986.
- [WEI78] WEISS, S. M. et al., A model-based method for computer-aided medical decision-making, **Artificial Inntelligence**, v. 11, p. 145 - 172, 1978.
- [WEI81c] WEISS, S. M. and KULIKOWSKI, C. A., Expert consultation systems: the EXPERT and CASNET projects, in BOND, A. H., **Machine Intelligence**, Infotech State of the Art Report, Series 9, n. 3, Pergamon Infotech Limited, p. 339 - 353, 1981.
- [ZAD65] ZADEH, L., A., Fuzzy Sets, **Information and Control**, v. 8, p. 338 - 383, 1965.
- [ZAD73] ZADEH, L., A., The Concept of a Linguistic Variable and its Application to Approximate Reasoning, New York, 1973.
- [ZAD83] ZADEH, L. A., Linguistic Variables, Aproximate Reasoning and Dispositions, **Medical Information**, v. 8, n. 3, p. 173 -186, 1983.
- [ZIM90] ZIMMERMANN, H., J., **Fuzzy Set Theory and Its Applications**, Massachusetts: Kluwer Academic Publishers, 1990.

## ANEXO I

### LISTAGEM DE ALGUMAS FUNÇÕES DO SISTEMA

```

/*****
**      Função para a definição dos conjuntos difusos      **
*****/

/*****
**** FUNCTION: DefConj
*****/
MakeFunction( DefConj, [],
{
OpenWriteFile("C:\kappa\conjunto");
While( ASCIICode(PostMenu("DEFINIÇÃO DOS CONJUNTOS DIFUSOS",
CONTINUAR, SAIR)) < 70)
{
Let [var 0] [nro 0] [min 0] [max 0]
    [cont1 0] [ter 0] [a 0] [aa 0] [b 0] [bb 0] [c 0]
    [cc 0] [d 0] [dd 0]
    {
PostInputForm("DEFINIÇÃO DOS CONJUNTOS DIFUSOS", var, "Nome da Variável
Linguística: ",
nro, "Numero de Conjuntos: ", min, "Valor Mínimo: ", max, "Valor Máximo: ");
WriteLine(var);
WriteLine(nro,min,max);
SetValue(cont1,0);
While(cont1 < nro)
{
PostInputForm("Conjunto Nro " # GetValue(contador+1),
ter,"Termo Linguístico: ", a,"Parâmetro (a): ",
aa,"Grão de Pertença {0,1}: ", b, "Parâmetro (b): ",
bb,"Grão de Pertença {0,1}: ", c, "Parâmetro (c): ",
cc,"Grão de Pertença {0,1}: ", d, "Parâmetro (d): ",
dd,"Grão de Pertença {0,1}: ");
WriteLine(ter,a,aa,b,bb,c,cc,d,dd);
SetValue(cont1,cont1+1);
};
};
};
CloseWriteFile();
});

```

```

/*****
**** Carrega os conjuntos difusos no sistema
*****/

/*****
**** FUNCTION: LoadSets
*****/

MakeFunction( LoadSets, [],
{
ClearList(Global:TempSet);
ClearList(Global:Conjunto);
GetSlotList(FunHepa, Global:TempSet);
GetInstanceList(FunHepa, Global:Conjunto);
EnumList(Global:Conjunto, x,{
    EnumList(Global:TempSet,y,{
        If GetSlotOption(x,y, MULTIPLE)
        Then {
            ClearList(x,y);
            AppendToList(x:y,0);
        }
        Else ResetValue(x,y);
    });
});
OpenReadFile("C:\kappa\fuzzy.set");
SetFileCursor(0,BEGIN);
While(Not(EndOfFile?()))
{
    Global:Lab = ReadWord();
    Global:NroLin = ReadWord();
    If Null?(Global:Lab)
    Then Break(Global:Lab);
    Global:Lab:min = ReadWord();
    Global:Lab:max = ReadWord();
    While(Global:NroLin > 0)
    {
        ClearList(Global:TempSet);
        AppendToList(Global:TempSet, ReadWord(), ReadWord(), ReadWord(), ReadWord(),
ReadWord(), ReadWord(), ReadWord(), ReadWord(), ReadWord());
        Let [a a#GetNthElem(Global:TempSet, 1)]
            [b b#GetNthElem(Global:TempSet, 1)]
            [c c#GetNthElem(Global:TempSet, 1)]
            [d d#GetNthElem(Global:TempSet, 1)]
        {

```



```

    ClearList(Global:Lab:a);
    ClearList(Global:Lab:b);
    ClearList(Global:Lab:c);
    ClearList(Global:Lab:d);
    AppendToList(Global:Lab:a, GetNthElem(Global:TempSet,
2),GetNthElem(Global:TempSet, 3));
    AppendToList(Global:Lab:b, GetNthElem(Global:TempSet,
4),GetNthElem(Global:TempSet, 5));
    AppendToList(Global:Lab:c, GetNthElem(Global:TempSet,
6),GetNthElem(Global:TempSet, 7));
    AppendToList(Global:Lab:d, GetNthElem(Global:TempSet,
8),GetNthElem(Global:TempSet, 9));

};
    Global:NroLin = Global:NroLin -1;
};
};
CloseReadFile();
PostMessage("Os conjuntos difusos foram todos carregados.");
} );

```

```

/*****
**      Função de fusificação          **
*****/

```

```

/*****
**** FUNCTION: Fuzzy
*****/

```

```

MakeFunction( Fuzzy, [objeto valor],
{

```

```

Let [f #objeto]
    [g #objeto]
    {
        ClearList(Global:LCod:f);
        ClearList(Global:LCod:g);
    };

```

```

If (valor >= GetNthElem(objeto:aMUB,1) And valor <= GetNthElem(objeto:dMUB, 1))
Then Miu(objeto,MUB,valor);
If (valor >= GetNthElem(objeto:aMOB, 1) And valor <= GetNthElem(objeto:dMOB, 1))
Then Miu(objeto,MOB,valor);
If (valor >= GetNthElem(objeto:aLIB, 1) And valor <= GetNthElem(objeto:dLIB, 1))
Then Miu(objeto,LIB,valor);
If (valor >= GetNthElem(objeto:aNOR, 1) And valor <= GetNthElem(objeto:dNOR, 1))
Then Miu(objeto,NOR,valor);

```

```

If (valor >= GetNthElem(objeto:aLIE, 1) And valor <= GetNthElem(objeto:dLIE, 1))
Then Miu(objeto,LIE,valor);
If (valor >= GetNthElem(objeto:aMOE, 1) And valor <= GetNthElem(objeto:dMOE, 1))
Then Miu(objeto,MOE,valor);
If (valor >= GetNthElem(objeto:aMUE, 1) And valor <= GetNthElem(objeto:dMUE, 1))
Then Miu(objeto,MUE,valor);
} );

```

```

/*****

```

```

**** FUNCTION: Miu

```

```

*****/

```

```

MakeFunction( Miu, [comp conj valor],

```

```

{

```

```

Let [a a#conj]

```

```

    [b b#conj]

```

```

    [c c#conj]

```

```

    [d d#conj]

```

```

    [f f#comp]

```

```

    [g g#comp]

```

```

{

```

```

If GetNthElem(comp:b, 1) == GetNthElem(comp:c, 1)

```

```

Then{

```

```

    If GetNthElem(comp:a, 2) == 1

```

```

    Then{

```

```

        If valor >= GetNthElem(comp:a,1) And valor <= GetNthElem(comp:b,1)

```

```

        Then{

```

```

            AppendToList(Global:LCod:f, conj);

```

```

            AppendToList(Global:LCod:g, 1.00);

```

```

        }

```

```

    Else{

```

```

        AppendToList(Global:LCod:f, conj);

```

```

        AppendToList(Global:LCod:g, FormatValue("%1.2f", (GetNthElem(comp:d,1) -
valor)/(GetNthElem( comp:d,1) - GetNthElem(comp:c,1))));

```

```

    };

```

```

}

```

```

Else{

```

```

    If GetNthElem(comp:d, 2) == 1

```

```

    Then{

```

```

        If valor >= GetNthElem(comp:c,1) And valor <= GetNthElem(comp: d,1)

```

```

        Then{AppendToList(Global:LCod:f, conj);

```

```

            AppendToList(Global:LCod:g, 1.00);

```

```

        }

```

```

    Else{AppendToList(Global:LCod:f, conj);

```

```

        AppendToList(Global:LCod:g, FormatValue("%1.2f", (valor -
GetNthElem(comp:a,1))/( GetNthElem(comp:b,1) - GetNthElem(comp:a,1))));
    };
}
Else{
    If valor >= GetNthElem(comp:c,1) And valor <= GetNthElem(comp:d,1)
    Then{AppendToList(Global:LCod:f, conj);
        AppendToList(Global:LCod:g, FormatValue("%1.2f", ( GetNthElem(comp:d,1) -
valor)/(GetNthElem(comp:d,1) - GetNthElem(comp:c,1))));
    }

    Else{AppendToList(Global:LCod:f, conj);
        AppendToList(Global:LCod:g, FormatValue("%1.2f", (valor -
GetNthElem(comp:a,1))/( GetNthElem(comp:b,1) - GetNthElem(comp:a,1))));
    };
};
};
}
Else{
    If valor >= GetNthElem(comp: b,1) And valor <= GetNthElem(comp: c,1)
    Then{AppendToList(Global:LCod:f, conj);
        AppendToList(Global:LCod:g, 1.00);
    }
    Else{
        If valor >= GetNthElem(comp: a,1) And valor <= GetNthElem(comp: b,1)
        Then{AppendToList(Global:LCod:f, conj);
            AppendToList(Global:LCod:g, FormatValue("%1.2f", (valor -
GetNthElem(comp:a,1))/ (GetNthElem(comp:b,1) - GetNthElem(comp:a,1))));
        }
        Else{AppendToList(Global:LCod:f, conj);
            AppendToList(Global:LCod:g, FormatValue("%1.2f", ( GetNthElem(comp:d,1) -
valor)/(GetNthElem(comp:d,1) - GetNthElem(comp:c,1))));
        };
    };
};
};
});

```

```

/*****
**** FUNCTION: SalvarPlanilha
*****/

```

```

MakeFunction( SalvarPlanilha, [],
{
ClearList(Global:TempSet);
ClearList(Global:Conjunto);

```

```
GetSlotList(Clinica, Global:TempSet);
GetSlotList(Laboratorio, Global:Conjunto);
Let [LC1 L # Global:ExL # "C1:" # L # Global:ExL # C # LengthList(Global:TempSet)]
    [LC2 L # Global:ExL # C # LengthList(Global:TempSet)#":L" # Global:ExL # C #
    (LengthList(Global:Conjunto)+ LengthList(Global:TempSet))]
    {
        RemoteSet(LC1, Global:TempSet, Excel,"paciente.xls");
        RemoteSet(LC2, Global:Conjunto, Excel,"paciente.xls");
    };
PostMessage("Os dados foram salvados na planilha PACIENTE.XLS do EXCEL.");

});
```