

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

# **Representação de Conhecimento e Redes de Decisão**

por  
Marcelo Ladeira

EQ-11 CPGCC - UFRGS  
Exame de Qualificação

Monografia submetida como requisito parcial (exame de qualificação) para a obtenção  
do grau de Doutor em Ciência da Computação

Área de abrangência:  
**Representação de conhecimento**

Tema de profundidade:  
**Redes de decisão**

Dr<sup>a</sup> Rosa Maria Viccari  
Orientadora

Porto Alegre, março de 1997

## CIP - Catalogação na Publicação

Ladeira, Marcelo

Representação de Conhecimento e Redes de Decisão / Marcelo Ladeira. -  
Porto Alegre: CPGCC da UFRGS, 1997.

p. 150: il. - (EQ-11)

Trabalho orientado pela Dr<sup>a</sup> Rosa Maria Viccari.

1. Representação de Conhecimento. 2. Esquemas de Representação de Conhecimento Certo. 3. Esquemas de Representação de Conhecimento Incerto. 4. Redes Bayesianas. 5. Teoria da Utilidade. 6. Redes de Decisão. 7. Inteligência Artificial. I. Viccari, Rosa Maria. II. Título. III. Série.

## Sumário

<b>Lista de Abreviaturas</b>	<b>5</b>
<b>Lista de Figuras</b>	<b>6</b>
<b>Lista de Tabelas</b>	<b>7</b>
<b>Resumo</b>	<b>9</b>
<b>Abstract</b>	<b>10</b>
<b>Parte I Representação de Conhecimento Certo</b>	<b>11</b>
<b>1 Introdução</b>	<b>13</b>
1.1 Inteligência e Inteligência Artificial	13
1.2 Conhecimento	18
1.3 Representação do Conhecimento	22
1.3.1 Fundamentos	22
1.3.2 Sistemas de Representação de Conhecimento.	25
1.3.3 Fatores para Seleção de um Esquema de Representação de Conhecimento	27
1.4 Raciocínio	28
1.5 Conclusão	32
<b>2. Esquema de Representação Baseado na Lógica</b>	<b>33</b>
2.1 Vantagens e Desvantagens	34
<b>3 Regras de Produção</b>	<b>36</b>
3.1 Regras de Produção	36
3.2 Linguagem Prolog	41
3.3 Vantagens e Desvantagens	43
<b>4 Redes Semânticas e Sistemas de Frames</b>	<b>45</b>
4.1 Redes Semânticas	45
4.2 Sistemas de <i>Frames</i>	48
4.1 Vantagens e Desvantagens	53
<b>5 Dependência Conceitual, Scripts e CYC</b>	<b>54</b>
5.1 Dependência Conceitual	54
5.1.1 Vantagens e Desvantagens no uso de dependência conceitual	57
5.2 Scripts	58
5.2.1 Vantagens e Desvantagens com o uso de Scripts	61
5.3 CYC	62
5.3.1 Vantagens e Desvantagens com o uso de CYC	65
<b>6 Esquemas de Representação de Propósito Específico</b>	<b>67</b>
6.1 Espaço de Estados	67
6.2 Representação Direta ou Analógica	68
6.2.1 Vantagens e Desvantagens da Representação analógica	69
<b>Parte II Representação de Conhecimento Incerto</b>	<b>71</b>

<b>7 Técnicas Baseadas no Raciocínio Não-Monotônico</b>	<b>74</b>
7.1 Lógica Não-Monotônica	75
7.2 Lógica Default	76
7.3 Aplicações do Raciocínio Não-Monotônico	78
7.3.1 Sistemas de Manutenção da Verdade	82
7.4 Vantagens e Desvantagens	88
<b>8 Teoria do Endosso</b>	<b>90</b>
8.1 Vantagens e Desvantagens	93
<b>9 Teoria de Probabilidades e Fatores de Certeza</b>	<b>94</b>
9.1 Teoria de Probabilidades	94
9.1.1 Vantagens e Desvantagens	96
9.2 Representação de incerteza	96
9.3 Meta-interpretador com incerteza	97
9.4 Combinações Heurísticas	100
9.5 Fatores de Certeza do MYCIN	101
9.5.1 Vantagens e Desvantagens	104
<b>10 Redes Bayesianas</b>	<b>106</b>
10.1 Vantagens e Desvantagens	110
<b>11 Teoria de Dempster-Shafer</b>	<b>113</b>
11.1 Vantagens e Desvantagens	116
<b>12 Técnicas Baseadas em Conjuntos Fuzzy</b>	<b>117</b>
12.1 Sistema Fuzzy	117
12.1.1 Teoria dos Conjuntos Fuzzy	118
12.1.2 Raciocínio Aproximado	121
12.1.3 Vantagens e Desvantagens	123
12.2 Teoria de Possibilidades	124
12.2.2 Distribuição de Possibilidades de Proposições Complexas	126
12.3 Normas Triangulares	128
12.3.1 Camada de Representação	128
12.3.2 Camada de Inferência	129
12.3.3 Camada de Controle	130
12.3.4 Vantagens e Desvantagens	131
<b>Parte III Redes de Decisão</b>	<b>133</b>
<b>13. Teoria da Decisão</b>	<b>134</b>
13.1 Decisão Racional	134
13.2 Teoria da Utilidade	137
<b>14 Redes de Decisões</b>	<b>141</b>
14.1 Redes de Crenças	145
<b>15 Conclusões</b>	<b>149</b>
<b>Bibliografia</b>	<b>152</b>

## Lista de Abreviaturas

AA	Modificadores (atributos) de ações ( <i>Action Aiders</i> ) em DC
ACT	Ações ( <i>ACTs</i> ) em DC
ATMS	TMS baseado em Suposições ( <i>Assumption-based TMS</i> )
CPT	Tabela de Probabilidades Condicionais ( <i>Conditional Probability Table</i> )
CYC	Teoria de representação de conhecimento de senso comum
CYCL	Linguagem de representação da CYC
DC	Dependência Conceitual
EMV	Valor Monetário Esperado ( <i>Expected Monetary Value</i> )
fbf	fórmula bem formada
FC	Fator de certeza
HMF	Hipótese do Mundo Fechado
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
i.é	isto é
JTMS	TMS baseado em Justificativas ( <i>Justification-based TMS</i> )
LD	Lógica <i>Default</i>
LNM	Lógica Não-Monotônica
LTMS	TMS baseado em Lógica ( <i>Logic-based TMS</i> )
MC	Medida de Crença
MD	Medida de Descrença
MEU	Máxima Utilidade Esperada ( <i>Maximum Expected Utility</i> )
MIMO	Multiplus Input Multiplus Output (Sistema Fuzzy)
MISO	Multiplus Input Single Output (Sistema Fuzzy)
MPG	Modus Ponens Generalizado (Lógica Fuzzy)
MTG	Modus Tolens Generalizado (Lógica Fuzzy)
NE	Nível epistemológico de representação em CYC
NH	Nível heurístico de representação em CYC
p.ex.	por exemplo
PA	Modificadores (atributos) de PP ( <i>Picture Aiders</i> ) em DC
PP	Atores, objetos físicos, forças naturais ou as divisões da memória humana: processador do consciente, memória intermediária ou de longo termo ( <i>Picture Producers</i> ) em DC
sse	se e somente se
TMS	Sistema de Manutenção da Verdade ( <i>Truth Maintenance System</i> )

## Lista de Figuras

<b>Quadro 1.1</b> - Definições de IA (Abordagens Centrada no Homem e Racional) _____	14
<b>Quadro 1.2</b> - Fases Históricas da IA Simbólica _____	17
<b>Figura 3.1</b> - Exemplo de Raciocínios (a) para Frente e (b) para Trás _____	39
<b>Figura 3.2</b> - Exemplo de Notações Lógica, Cláusulas de Horn e Prolog _____	41
<b>Figura 3.3</b> - Exemplo de Execução de um Programa Prolog _____	43
<b>Figura 4.1</b> - Exemplo de uma Rede Semântica _____	46
<b>Figura 4.2</b> - Rede Semântica Particionada _____	47
<b>Figura 4.3</b> - Herança de Propriedades Monovaloradas _____	52
<b>Figura 5.2</b> - Exemplo de Representação usando DC _____	56
<b>Figura 5.3</b> - Exemplo de Script _____	60
<b>Figura 7.1</b> - Exemplo de Rede de Dependências em JTMS _____	86
<b>Figura 9.1</b> - Exemplo de Incerteza Simbólica em Regras _____	97
<b>Figura 9.2</b> - Combinando Regras Incertas _____	98
<b>Figura 9.3</b> - Meta-interpretador Prolog com Incerteza _____	98
<b>Figura 9.4</b> - Avaliação de Incerteza com Heurística Inclusão _____	101
<b>Figura 9.5</b> - Propagação de Fatores de Certeza _____	104
<b>Figura 10.1</b> - Rede Bayesiana Típica _____	107
<b>Figura 10.2</b> - Exemplo de Clusterização e Condicionantes de Redes Bayesianas _____	110
<b>Figura 10.3</b> - Padrões de Inferências em Redes Bayesianas _____	111
<b>Figura 12.1</b> - Função Trapézio _____	118
<b>Figura 12.2</b> - Normalização com Granularidade 7 _____	121
<b>Figura 12.3</b> - Combinando Regras Incertas _____	129
<b>Figura 13.1</b> - Árvore de Decisão _____	137
<b>Figura 13.2</b> - Tipos de Dominância _____	139
<b>Figura 14.1</b> - Rede de Decisão Simples _____	142
<b>Figura 14.2</b> - Exemplo de Rede de Decisão _____	144
<b>Figura 14.3</b> - Transformação t1: Inversão de Arco em Rede de Decisão _____	144
<b>Figura 14.4</b> - Aplicações das Transformações t2 e t3: Absorção de Nós _____	145

## Lista de Tabelas

<b>Tabela 3.1</b> - Objeto-Atributo-Valor _____	37
<b>Tabela 4.1</b> - Algoritmo Herança de Propriedades _____	51
<b>Tabela 7.1</b> - Base de Conhecimento da História do Assassinato ABC _____	84
<b>Tabela 8.1</b> - Exemplo de Aplicação de Endossos em Reconhecimento de Planos ____	92
<b>Tabela 9.1</b> - Combinação de Incerteza Simbólica _____	99
<b>Tabela 9.2</b> - Combinações Heurísticas de Incerteza Numérica _____	100
<b>Tabela 9.3</b> - Predicados Heurísticas de Incerteza Numérica _____	101
<b>Tabela 11.1</b> - Combinação Simples de Dempster _____	115
<b>Tabela 11.2</b> - Exemplo de Combinação de Dempster com Conjunto Vazio _____	116
<b>Tabela 11.3</b> - Intervalo de Crença e Plausibilidade _____	116
<b>Tabela 12.1</b> - Axiomas de T-normas e de T-conormas _____	119
<b>Tabela 12.2</b> - Principais T-normas e T-conormas _____	119
<b>Tabela 12.3</b> - Principais Operadores de Implicação Fuzzy _____	122
<b>Tabela 12.4</b> - Exemplo de Distribuições de Possibilidade e Probabilidades _____	124
<b>Tabela 12.5</b> - Probabilidades Lingüísticas _____	129
<b>Tabela 12.6</b> - Família de T-normas de Schweizer e Sklar _____	130

“Cada homem vale o que valer o seu ideal. Ideais grandiosos geram homens magnânimos. Ideais pequenos geram homens mesquinhos.

No grande desfile da mesquinhez oferecido pelos nossos tempos, parece que vemos passar diante dos nossos olhos o rosto daqueles que são incapazes de comprometer-se com uma causa que ultrapasse o seu egoísmo: os *falsamente prudentes* e *‘bem-comportados’*; os *acomodados*, que trocam os ideais mais elevados pelos mais exequíveis; os *comodistas*, que substituem o cume pelo declive que passa à beira de todos os destinos; os *preguiçosos*, que subordinam o melhor ao mais fácil, que trocam o esforço construtivo pela praia e pela festinha, um amor grande e profundo pelos amoricos de fim de semana .. Todos eles se agrupam ao lado dessa variadíssima multidão dos que *vivem mais de desejos e sensações do que de ideais*, dos que se alimentam mais do colorido mundo das imagens televisivas do que da leitura, do estudo ou da reflexão que cultiva a inteligência e forma critérios”

Rafael Llano Cifuentes  
(Bispo Auxiliar da Arquidiocese do Rio de Janeiro),  
*Grandeza de Coração*, São Paulo: Quadrante, 1996, p.10-11

“A vida é uma oportunidade; aproveita-a.  
A vida é beleza; admira-a.  
A vida é felicidade; saboreia-a.  
A vida é um sonho; torna-o realidade.  
A vida é um desafio; enfrenta-o.  
A vida é um jogo; joga-o.  
A vida é preciosa; proteja-a.  
A vida é riqueza; conserva-a.  
A vida é amor; desfruta-o.  
A vida é mistério; desvenda-o.  
A vida é promessa; cumpra-a.  
A vida é tristeza; supera-a.  
A vida é um hino; canta-o.  
A vida é um combate; aceita-o.  
A vida é uma tragédia; domina-a.  
A vida é uma aventura; encara-a.  
A vida é um gozo; mereça-o.  
A vida é vida; defenda-a.”

Madre Teresa de Calcutá.



## Resumo

Essa monografia constitui a atividade final dos exames de qualificação, do programa de doutorado do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal do Rio Grande do sul. Esses exames compreendem o **exame de abrangência**, onde o candidato deve demonstrar conhecimento nos fundamentos, metodologias e técnicas em um domínio específico e o **exame de profundidade**, onde deve demonstrar conhecimento do estado da arte, análise de problemas relevantes e comparação crítica de soluções alternativas propostas na literatura para um tema de pesquisa específico, dentro da área de abrangência escolhida.

Esse trabalho apresenta-se dividido em três partes.

As duas primeiras correspondem ao exame de abrangência em **representação de conhecimento**. Na Parte I são apresentados os conceitos de Inteligência Artificial e conhecimento (certo e incerto) e discutida a importância da representação de conhecimento e do processo de raciocínio em sistemas inteligentes. A seguir são apresentados os esquemas de representação lógica, regras de produção, redes semânticas e *frames* e outros, com maior orientação semântica (dependência conceitual, scripts e CYC), para representar conhecimento certo. Na Parte II são discutidas as principais abordagens simbólicas (baseadas no raciocínio não-monotônico e a teoria do endosso de Cohen) e numéricas (probabilidades, regras com fatores de certeza associados, redes bayesianas, teoria da evidência de Dempster-Shafer e técnicas baseadas na teoria dos conjuntos fuzzy) para representar conhecimento incerto e raciocinar com base nele.

A Parte III corresponde ao exame de profundidade em **redes de decisão**. Inicialmente são apresentados os princípios da teoria da decisão e da teoria da utilidade e como eles podem ser combinados com redes bayesianas para produzir o formalismo de redes de decisão. O estado da arte em redes de decisão é apresentado e discutido.

**PALAVRAS CHAVES:** representação do conhecimento, esquemas de representação de conhecimento certo, esquemas de representação de conhecimento incerto, redes bayesianas, teoria da utilidade, redes de decisão, inteligência artificial.

## Abstract

This document is the final activity of the doctoral program's qualifying examination of the Post-Graduation Program in Computer Science at The Federal University of the Rio Grande do Sul, Brazil. The qualifying examination is forked into two examinations. The former is the **wide-ranging examination** where the candidate must show that he/she knows the main principles, methodologies and techniques of a specific domain. The later is the **depth examination** where the candidate must show enough knowledge about a specific research subject state-of-the-art, major problems analyze and critical comparison of alternative solutions proposed in the literature in that field. The candidate's major - defined by the depth examination - must be into his/her wide-ranging examination area.

This monograph is compounded of three parts. Both the first and the second one correspond to wide-ranging examination on **knowledge representation**. The last part correspond to depth examination on **decision networks**. The first part presents the concepts of artificial intelligence and certain and uncertain knowledge and discuss the importance of both knowledge representation and reasoning on intelligence systems. The logic, rule production, semantic networks and frames knowledge representation schemes are presented as well as other more semantic orientation schemes (i.e. conceptual dependency, scripts and CYC).

The second part describes the main symbolic and numeric approaches to represent uncertain knowledge and for reasoning based on it. The symbolic approach are those based on the non-monotonic logics and the Cohen's theory of endorsements. The formalisms that use numeric approach are probability theory, certainty factors associated with rule production, Bayesian networks, Dempster-Shafer evidence theory and a few fuzzy-set based techniques.

The third part presents the principles of the decision theory and of the utility theory, and how they can be combined with Bayesian networks to yield the decision network formalism. The decision networks state-of-the is also presented and discussed.

**KEYWORDS:** knowledge representation, certain knowledge representation schemes, uncertain knowledge representation schemes, Bayesian networks, utility theory, decision networks, artificial intelligence.

## Parte I Representação de Conhecimento Certo

Inicialmente são introduzidos os conceitos de Inteligência Artificial (IA) e conhecimento (certo e incerto) e discutida a importância da representação de conhecimento e do processo de raciocínio em sistemas inteligentes. A seguir são apresentados os esquemas de representação lógica, regras de produção, redes semânticas e *frames* e outros com maior orientação semântica (dependência conceitual, scripts e CYC), todos eles para representar conhecimento certo.

Sistemas convencionais de IA simbólica utilizam raciocínio monotônico. O uso do raciocínio monotônico em um domínio específico só é possível quando se dispõe de um modelo completo, consistente e imutável do domínio em questão. As informações que tais sistemas utilizam são completas com relação ao domínio e a adição de novos fatos consistentes à base de conhecimentos não implica a revisão da crença nos fatos existentes, os quais são sabidamente verdadeiros. É sempre possível afirmar que, em um dado momento, um fato é tido como verdadeiro ou tido como falso. Nestas condições o uso dos esquemas para representação de conhecimento certo é adequado.

Os esquemas de representação abordados são os utilizados na área conhecida como IA simbólica. Nela se acredita que “um programa de computador necessita de uma representação geral do mundo em termos da qual são interpretadas suas entradas, para ser capaz de agir inteligentemente” [BIT 96, p.24]. Logo a simulação da inteligência em computador requer conhecimento (declarativo) e alguma capacidade de raciocínio (manipulação do conhecimento visando gerar novos conhecimentos) [KIR 91].

Não são abordadas outras linhas de pesquisa de IA tais como *conexionismo*, *computação evolutiva* ou *moboticismo*.

O Conexionismo (ou redes neurais) simula comportamento inteligente através do desenvolvimento de programas baseados em modelos inspirados na estrutura e funcionamento do cérebro humano. Esses modelos não utilizam representação explícita de conhecimento, sendo que o conhecimento pode ser considerado codificado nos pesos sinápticos associados às diversas sinapses dos neurônios que compõem uma rede neural.

Já a computação evolutiva utiliza a metáfora da evolução natural. Para essa linha de pesquisa não é necessário ter um conhecimento prévio de uma forma (geralmente um algoritmo e uma representação do mundo) de encontrar uma solução para um problema.

Parte-se de um conjunto inicial - *população* - de candidatos a solução (o qual pode ser obtido através de um sorteio aleatório), uma *função de avaliação* capaz de atribuir um valor ao desempenho de cada membro da população (como solução para o problema a resolver), *critérios para seleção* dos indivíduos (membros da população) que serão utilizados para reprodução através da aplicação de *operadores* (baseados na evolução natural) tais como “*crossover*” (recombinação) e *mutação*.

São selecionados alguns membros da população (p. ex. os melhores avaliados) para reprodução. Os novos indivíduos são avaliados através da função de avaliação e, se superiores aos existentes, são eliminados indivíduos da população antiga para ceder lugar

para os novos, compondo uma nova geração. Assim, a cada nova geração, se caminha na direção da solução do problema ou se mantêm a melhor aproximação anterior.

Os moboticistas lidam com agentes reativos, i.é artefatos de software (e/ou hardware) que possuem *sensores* para monitorar o ambiente (mundo) onde atuam e *atuadores* para agirem (normalmente baseados em regras do tipo condição-ação), em resposta ao estado atual captado via os sensores.

Brooks [BRO 91] construiu diversos *mobot - mobile robots* - os quais operam sem supervisão, interagindo diretamente com o mundo através de percepção e ação e apresentando comportamento inteligente. Em sua pesquisa Brooks busca construir, de forma incremental, criaturas inteligentes completas, através da decomposição de um sistema inteligente em produtores de atividades independentes e paralelas, os quais interagem diretamente com o mundo (no qual atuam) através de percepção e ação. Desta forma advoga que é possível obter comportamento inteligente *sem representação*, i.é sem a necessidade de adotar a abordagem clássica da IA simbólica: decomposição de um sistema inteligente em unidades de processamento de informação independentes, as quais interagem entre si via representações.

# 1 Introdução

Neste capítulo são conceituados IA, conhecimento, representação de conhecimento e raciocínio em sistemas de IA. Também é descrito o enquadramento de *representação de conhecimento* (área de exame de qualificação em abrangência) em *inteligência artificial* (linha de pesquisa do doutorando).

## 1.1 Inteligência e Inteligência Artificial

A dificuldade em conceituar inteligência artificial está associada a dificuldade em definir precisamente o termo inteligência.

John McCarthy cunhou o termo IA, em 1956, como “*a capacidade de uma máquina realizar funções que, se realizadas pelo ser humano, seriam consideradas inteligentes*”, em uma conferência de verão realizada durante dois meses em Dartmouth College. Essa conferência foi proposta por McCarthy, Marvin Minsky (Harvard), Nathaniel Rochester (IBM) e Claude Shannon (Bell Laboratories) [BIT 96, P.19-20].

Para Charniak e McDermott [CHA 85], inteligência artificial *é o ramo da Ciência da Computação voltado para o estudo das faculdades mentais através do uso de modelos computacionais*. Esses modelos são sistemas de computação inteligentes, i.e. apresentam características associadas à inteligência no comportamento humano, como compreensão de linguagem, aprendizado, raciocínio, resolução de problemas, etc. [BAR 86, v. I, p.3].

Russell e Norvig [RUS 95, p.5] apresentam uma interessante análise das abordagens utilizadas em algumas definições de IA, reproduzidas no Quadro 1.1. Essas definições estão organizadas em quatro categorias e foram obtidas em oito livros-textos publicados de 1985 a 1993. As primeiras dizem respeito a maneira como os sistemas inteligentes simulam os seus *processos de pensamento e raciocínio: como humanos ou racionalmente*.

As duas últimas categorias estão relacionadas a forma de *comportamento* dos sistemas inteligentes: *como humanos ou racionalmente*.

Atuação *como humanos* permite aferição de performance em termos da performance humana na realização da mesma tarefa. Atuar *racionalmente* permite aferição de performance em termos do conceito ideal de inteligência denotado racionalidade, o qual implica que se o sistema é racional, então ele sempre realiza a coisa certa. Obviamente o comportamento humano pode ser racional, mas as pessoas freqüentemente cometem erros sistemáticos durante seus processos de raciocínio.

Com essas categorias podemos classificar os sistemas inteligentes em sistemas que: a) pensam como os humanos; b) pensam racionalmente; c) atuam como humanos; e d) atuam racionalmente.

A abordagem centrada no homem pode ser considerada uma ciência empírica, envolvendo elaboração de hipóteses e confirmação experimental das mesmas. A abordagem racional é mais embasada na matemática e engenharia. Ambas trouxeram progressos para a área de IA, como veremos a seguir.

	Como os Humanos	Racionalmente
Pensamento e Raciocínio	<p>“O excitante esforço novo para fazer computadores pensarem ... <i>máquinas com mentes</i>, no sentido amplo e literal” (Haugeland, 1985)</p> <p>“[A automação das] atividades que associamos com pensamento humano, tais como tomar decisão, resolver problemas, aprender ...” (Bellman, 1978)</p>	<p>“O estudo das faculdades mentais através do uso de modelos computacionais” (Charniak e McDermott, 1985)</p> <p>“O estudo das computações que tornam possível perceber, raciocinar e agir” (Wiston, 1992)</p>
	<p>“A arte de criar máquinas que executam funções que requerem inteligência quando executadas por pessoas” (Kurzweil, 1990)</p> <p>“O estudo de como fazer os computadores realizarem coisas que, no momento, pessoas fazem melhor” (Rich e Knight, 1991)</p>	<p>“Um campo de estudo que procura explicar e emular o comportamento inteligente em termos de processos computacionais” (Schalkoff, 1990)</p> <p>“O ramo da Ciência da Computação voltado para a automação do comportamento inteligente” (Luger e Stubblefield, 1993)</p>

**Quadro 1.1** - Definições de IA (Abordagens Centrada no Homem e Racional)

Fonte: Russel e Norvig, 1995. Artificial Intelligence: A Modern Approach. p.5

*Atuar como humano* é motivada principalmente pelo fato de que programas de IA têm que interagir com pessoas, em uma maneira em que possam ser entendidos, principalmente em atividades tais como: sistemas especialistas explicarem como chegaram aos seus diagnósticos ou mesmo um sistema de processamento de linguagem natural que esteja dialogando com o seu usuário.

Alan Turing propôs, em 1950, uma definição operacional para inteligência e um teste para verificar se um sistema (executando em um computador) possuía comportamento inteligente ou não, sendo *comportamento inteligente* entendido como a habilidade de atingir nível de performance similar ao humano, em todas as tarefas cognitivas, suficiente para iludir um interrogador. O teste proposto foi: um sistema deverá ser interrogado por uma pessoa, via um teletipo (em uma linha de comunicação), e possuirá comportamento inteligente se o interrogador não poder dizer se está se comunicando com um computador ou com uma pessoa. Observe que é evitada uma interação física direta entre o interrogador e o computador, visto que Turing considera a simulação física de uma pessoa desnecessária para a inteligência.

Sistema que passe no teste de Turing deve possuir capacidades de: processar linguagem natural; representar, armazenar e posteriormente recuperar informações providas antes ou durante o diálogo com o interrogador; raciocinar com base nas informações para responder questões e inferir novos fatos; e de aprendizagem para se adaptar às novas circunstâncias.

Para *pensar como um humano*, um sistema inteligente deve possuir alguma forma de determinar como as pessoas pensam, i.é como a mente humana trabalha. Uma vez que se tenha uma teoria suficientemente precisa da mente, é possível expressá-la como um programa de computador. Essa é a abordagem de *modelagem cognitiva*. No início do desenvolvimento das pesquisas em IA se dedicou muito esforço para a construção de programas gerais de solução de problemas, sendo bastante conhecido o GPS - General Problem Solver de Newell e Simon, implementado em 1961. Newell e Simon não tinha apenas o objetivo de solucionar problemas de forma correta, utilizando o GPS, mas sim comparar os passos durante os processos de raciocínio do programa e das pessoas, ao resolverem o mesmo problema.

Atualmente há uma simbiose entre IA e a *ciência cognitiva*, a qual utiliza modelos computacionais da IA em conjunto com técnicas experimentais da psicologia para construir teorias precisas e testáveis de como trabalha a mente humana. Essas teorias têm sido utilizadas, principalmente nas áreas de visão computacional, processamento de linguagem natural e aprendizado.

Sistemas que *pensam racionalmente*, i.é que desenvolvem inferência corretas (processos de raciocínio irrefutáveis), são baseados no uso da *lógica clássica*. A lógica provê uma notação precisa para construir declarações sobre todos os tipos de objetos do mundo e relacionamentos entre eles. Em meados dos anos 60 foram desenvolvidos programas que a partir de uma notação lógica descrevendo um problema, encontravam uma solução se existisse uma, dados tempo de processamento e memória suficientes. Se não existisse qualquer solução o programa poderia não terminar o seu processamento.

A abordagem de se construir programas inteligentes, em IA, utilizando notação lógica pode apresentar as seguintes dificuldades: a) ser difícil representar conhecimento informal através de uma notação lógica, principalmente se existir alguma incerteza associada à ele; b) os requisitos de tempo de processamento e de memória serem tão elevados que tornem impraticável a obtenção de uma solução na prática, a menos que se utilizem algumas informações de controle tais como que passos do processo de raciocínio tentar primeiro, etc.

A categoria dos sistemas que *atuam racionalmente* utiliza a abordagem *agente racional*. Um *agente* é um artefato (de hardware e/ou de software) que percebe (sinais do meio ambiente onde atuam e/ou de outros agentes) e atua. Um agente atua racionalmente quando atua para atingir os seus objetivos, face as suas crenças (sobre o mundo no qual atuam e/ou sobre outros agentes).

Nessa abordagem, IA é vista como o estudo e construção de agentes racionais. Ela é mais geral do que a abordagem da tradição lógica, a qual apenas enfatiza a realização de inferências corretas. Inferir corretamente, algumas vezes faz parte do comportamento de um agente racional, principalmente quando ele conclui que uma ação levará à obtenção do seu objetivo e então, executa aquela ação. Por outro lado algumas vezes o agente tem que tomar uma decisão mesmo que nenhuma das decisões possíveis seja correta. Em outras situações o agente pode agir de forma reativa, sem envolver qualquer processo de inferência, face a algum sinal que percebe. Por exemplo, retirar rapidamente a mão de um forno quente é uma ação reflexiva que não demanda nenhum processo de raciocínio.

Observe que essa abordagem de agente racional também requer todas as habilidades cognitivas necessárias para um programa passar no teste de Turing, as quais permitem

tomar decisões racionais. Por outro lado é mais fácil para um pesquisador desenvolver sistemas que atuam racionalmente, visto que o padrão de racionalidade é geral e está bem estabelecido, do que sistemas baseados no comportamento humano, o qual ainda não é completamente conhecido.

Observe que a *racionalidade perfeita* - fazer sempre a coisa certa - não é possível, em ambiente complexos, por razões de restrição de recursos computacionais e tempo. Nesses casos pode-se trabalhar com *racionalidade limitada*, ou seja atuar de forma “apropriada” quando não existe tempo suficiente para fazer os cálculos que se quer.

De forma geral, as metas principais da pesquisa em IA são: a) a construção de máquinas que executam tarefas em que normalmente a inteligência humana é necessária, e b) entender os princípios que tornam a inteligência possível, ou seja, o desenvolvimento de modelos computacionais dotados de comportamento inteligente.

As principais áreas de pesquisa da IA atual são: sistemas especialistas, aprendizagem, **representação do conhecimento**, aquisição do conhecimento, tratamento de informação incerta, visão computacional, robótica, controle inteligente, inteligência artificial distribuída, modelagem cognitiva, arquiteturas para sistemas inteligentes, processamento de linguagem natural e interfaces inteligentes.

Kirsh [KIR 91] identifica as seguintes suposições (pontos focais de debates e divisores de posições), todas elas no cerne da discussão sobre como a inteligência pode ser capturada por um mecanismo artificial:

- a) *importância do conhecimento e conceitualização*: inteligência (transcendendo o nível de inteligência de inseto) requer conhecimento declarativo e alguma forma de mecanismo tipo raciocínio. Denomina-se **cognição** ao *processo computacional de raciocínio* (i.e junção do conceito clássico de raciocínio e de processos periféricos tais como reconhecimento de linguagem e identificação de objetos). O núcleo da IA é o estudo das conceitualizações do mundo, pressupostas e usadas pelos sistemas inteligentes durante a cognição.
- b) *independência*: inteligência e o conhecimento que ela pressupõe podem ser estudados abstraindo-se os detalhes de percepção e controle motor (ação);
- c) *evolução dos estados da cognição e linguagem*: é possível descrever a evolução dos estados de conhecimento ou informacionais criados durante os processos de cognição usando um vocabulário muito semelhante a linguagem natural ou alguma versão lógico-matemática da linguagem natural;
- d) *aprendizado pode ser adicionado mais tarde*: a evolução dos estados da cognição (e dos conhecimentos do domínio necessários para a cognição) pode ser estudada de forma separada do estudo dos processos de aprendizagem, desenvolvimento psicológico e mudança evolutiva;
- e) *arquitetura uniforme*: existe um única arquitetura na qual qualquer tipo de cognição pode ser simulado.

Diferentes linhas de pesquisa na IA são baseadas na aceitação ou não de uma mistura dessas proposições e de seus corolários. Kirsh apresenta as seguintes linhas:

- lógicos: adotam todas as suposições, exceto a última, sobre a qual são neutros.
- conexionistas: rejeitam a importância do conhecimento e do raciocínio na cognição e que a evolução dos estados da cognição possa ser expressa por uma linguagem. Enfatizam a importância do treinamento no estudo da cognição, são



neutros com relação a existência de uma arquitetura única e estão divididos sobre a independência do estudo da cognição dos detalhes de percepção e ação.

- moboticistas: rejeitam a importância do raciocínio, conceitualização, descrição da evolução dos estados da cognição por linguagem, independência dos detalhes de percepção e ação, uniformidade de arquitetura para simulação da cognição e a separabilidade de conhecimento e aprendizagem.

Segundo Bittencourt [BIT 96], existem duas grandes linhas de pesquisa em IA: a simbólica e a conexionista.

A IA simbólica trabalha com a manipulação simbólica e segue a tradição lógica, sendo Newell e McCarthy seus principais defensores. O Quadro 1.2 [BIT 96, p.20], baseado em relatório interno do MIT (Massachusetts Institute of Technology), apresenta fases da histórica da IA simbólica.

ÉPOCA	OBJETIVOS	MÉTODOS	FRACASSO
Clássica (1956-70)	simular a inteligência humana.	solucionadores gerais de problemas e lógica.	subestimação da complexidade de problemas.
Romântica (1970-80)	simular a inteligência humana em situações pré-determinadas.	formalismos de representação de conhecimento adaptados ao tipo de problema.	subestimação da quantidade de conhecimento necessária para tratar mesmo o mais banal problema de senso comum.
Moderna (1980-90)	simular o comportamento de um especialista humano ao resolver problemas em um domínio específico.	sistemas de regras, representação da incerteza, conexionismo.	subestimação da complexidade do problema de aquisição de conhecimento.

**Quadro 1.2 - Fases Históricas da IA Simbólica**

Observa-se um amadurecimento da área com a conseqüente adoção de metas bem mais modestas, “*até os objetivos atuais de tornar os computadores mais úteis através de ferramentas que auxiliem as atividades intelectuais dos seres humanos*”, o que coloca a IA na perspectiva de uma atividade que praticamente caracteriza a espécie humana: a capacidade de utilizar representações externas, seja na forma de linguagem seja através de outros meios” [HIL 89 Apud BIT 96, p.22].

A linha conexionista visa a modelagem da inteligência humana através da simulação da estrutura e funcionamento do cérebro, em especial dos neurônios e das suas ligações. A IA conexionista trabalha com a propagação de sinais através de redes de neurônios artificiais. A metodologia de redes neurais apresentam as seguintes características:

- capacidade de aprendizagem através de treinamento com base em exemplos e de generalizar este aprendizado de maneira a reconhecer casos que não haviam sido apresentados como exemplo.
- não requer conhecimento a respeito de eventuais modelos matemáticos dos domínios de aplicação.

- elevada imunidade ao ruído, com o desempenho da rede neural piorando de maneira gradativa, em presença de informações (sinais de entrada para os neurônios artificiais) falsas ou ausentes.
- bom desempenho em tarefas mal definidas, onde falta o conhecimento explícito sobre como encontrar uma solução.

Os primeiros trabalhos datam de 1943 quando o neurofisiologista Warrem McCulloch e o lógico Walter Pitts desenvolveram o primeiro modelo matemático de um neurônio. A área nasceu praticamente junta com a IA simbólica mas passou por um período de ostracismo após a publicação do livro *Perceptrons: An Introduction to Computational Geometry* por Marvin Minsky e S. A. Papert, ambos professores MIT, no qual as propriedades de uma rede de neurônios artificiais (perceptrons) foram analisadas e suas limitações apontadas. Na década de 80 houve um renascimento da área, motivado principalmente por melhores conhecimentos da estrutura real do cérebro e melhores algoritmos de redes neurais e disponibilidade de computadores mais poderosos, inclusive paralelos.

As áreas de aplicações atuais da IA conexionista incluem reconhecimento de padrões em geral (visão computacional, reconhecimento de voz, etc.) processamento de sinais, diagnóstico de falhas, controle de processos, etc.

Também existe um novo paradigma, baseado na metáfora da teoria da evolução natural, denominado computação evolutiva, que constitui uma linha paralela a IA simbólica e a IA conexionista.

Como vimos (na introdução da Parte I), a computação evolutiva não exige um conhecimento prévio de uma maneira de encontrar uma solução para um problema. Parte-se de uma *população inicial* (de candidatos a solução), uma *função de avaliação*, *critérios para seleção* dos membros da população que serão utilizados para reprodução (através da aplicação de *operadores* tais como *crossover* e/ou *mutação*) e gera-se uma nova geração. Por construção, a cada nova geração, se caminha na direção da solução do problema ou se mantém a melhor aproximação anterior.

## 1.2 Conhecimento

A partir deste ponto em diante todas as referências a IA devem ser entendidas como a IA simbólica a menos que seja explicitada outra linha de pesquisa.

Um dos resultados que surgiu nas três primeiras décadas da pesquisa em IA é que inteligência requer conhecimento [RIC 94, p.9]. Assim, para Barr [BAR 86, p. 143] o comportamento inteligente de uma pessoa ou de um programa pode ser descrito em termos do conhecimento que demonstram ter, ao executarem a tarefa que requer inteligência.

Para resolver problemas, os programas de IA precisam de conhecimento e mecanismos para a sua manipulação. O conhecimento deve ser representado de uma forma que o programa possa manipula-lo. Lidamos com dois tipos diferentes de entidades:

- a) conhecimento (fato), verdades em algum mundo relevante - que queremos representar;
- b) representações do conhecimento (fato) em algum formalismo escolhido - o que realmente pode ser manipulado por programas de computador.

Para Rich e Knight [RIC 94], além da característica de *indispensabilidade*, o conhecimento em IA possui propriedades menos desejáveis, pois é volumoso, de difícil caracterização precisa, está mudando constantemente e difere de simples dados por organizar-se de uma maneira que corresponde ao modo como será utilizado. As técnicas de IA são métodos que exploram o conhecimento representado de forma que:

- a) capture generalizações (através do agrupamento de situações que compartilham propriedades importantes);
- b) seja compreendido pelas pessoas que o fornecem;
- c) possa ser facilmente modificado para corrigir erros ou refletir mudanças do mundo e da nossa visão do mundo;
- d) permita o seu uso, mesmo quando seja impreciso ou incompleto;
- e) possa ser usado para limitar as várias possibilidades que em geral têm que ser analisadas, evitando aumento da quantidade de conhecimento a armazenar.

A noção de *dado* está associada aos valores não interpretados que o sistema possui para processamento. A noção de *informação* consiste em dados selecionados e organizados para um fim determinado. *Conhecimento* é informação estruturada. A estruturação permite construir procedimentos para explorar as relações entre dados, atribuindo-lhes semântica [VIC 90]. O conhecimento pode aparecer sob a forma de objetos, proposições e definições, conceitos e relações, teoremas e regras, algoritmos, estratégias e táticas, e metaconhecimento.

Como Barr e Feigenbaum [BARR 86] destacam, são os procedimentos que permitem uma manipulação inteligente das estruturas de dados. Portanto uma estrutura de dados, *per si* não é um conhecimento, assim como uma enciclopédia não representa um conhecimento. Embora se possa dizer, de forma figurada, que um livro é uma fonte de conhecimento, sem o leitor o livro é apenas tinta em um papel!

Uma *base de conhecimentos* (convencional) é constituída de fatos e de um conjunto de regras e heurísticas (expressas através de estratégias) capazes de permitir a derivação de outros fatos, em um domínio específico. Nela há mais descrições do que dados, a informação está associada à função de interpretação existente no sistema, os fatos e as regras estão organizados em unidades que facilitam a inferência de novos fatos, e fatos são pontos de partida para a aplicação de regras.

O conhecimento pode ser classificado em: do domínio, genérico, básico e formal.

O *conhecimento do domínio* situa-se ao nível da aplicação, sendo constituído de conceitos (representação declarativa de objetos do domínio da aplicação) e dos termos básicos do domínio, podendo ser obtido em livros.

O *conhecimento genérico* situa-se ao nível da tarefa, sendo constituído de regras (estabelecendo associações empíricas entre causas e efeitos, graus de crenças e probabilidades de conclusão, e situações e ações a realizar), fornecidas por um especialista, com base na experiência (perícia) que possui.

O *conhecimento básico* situa-se ao nível da capacidade, sendo constituído de modelos (coleção de regras interrelacionadas, usualmente associadas a hipótese particular ou a uma conclusão global de diagnóstico).

O *conhecimento formal* situa-se ao nível da tecnologia, sendo constituído de estratégias (regras e procedimentos que ajudam a explorar o conhecimento) ou heurísticas.

Os sistemas inteligentes convencionais dispõem de um modelo completo, consistente e imutável do domínio em questão. As informações que tais sistemas utilizam são completas com relação ao domínio e a adição de novos fatos consistentes à base de conhecimentos não implica a revisão da crença nos fatos existentes (os quais são verdadeiros). É sempre possível afirmar que, em um dado momento, um fato é tido como verdadeiro ou tido como falso ou nem verdadeiro ou falso. Como não existe dúvida (incerteza) com relação a esse tipo de conhecimento, ele é dito *conhecimento certo*.

Muitos domínios realísticos não são tão estruturados assim. Para raciocínio neles é necessário fazer **simplicificações**, tais como deixar fatos desconhecidos (*ignorância teórica* sobre o domínio), não ditos (*ignorância prática* devido a falta de evidências sugestivas) ou grosseiramente resumidos (devido ao *esforço excessivo* necessário para detalhá-los) [PEA 88]. Por exemplo, um médico pode diagnosticar uma doença, com base em alguns sintomas, no entanto esse diagnóstico é uma hipótese que pode ser incorreta. Nesse caso as simplificações adotadas podem ter origem em conhecimento médico incompleto sobre a patologia em questão (*ignorância teórica*), sintomas determinantes da patologia não detectados, devido a fase de evolução da doença (*ignorância prática*) ou não realização de exames complementares (dados sintomáticos grosseiramente resumidos). Entretanto o diagnóstico obtido tem uma confiabilidade maior do que um simples palpite, sendo que o médico que o realizou tem uma noção da validade da hipótese formulada, quer em termos absolutos quer em comparação com outros diagnósticos hipotéticos.

As vezes é preciso raciocinar com falta de informação e atribuir medidas de certeza às crenças em algumas hipóteses para as quais existem certas evidências de apoio. Medidas de certeza não estão relacionadas com valores verdade. Valores verdade (como em um sistema de lógica de primeira ordem) caracterizam as *fórmulas* em discussão enquanto medidas de certeza caracterizam fatos *invisíveis* (i.é exceções não cobertas nas fórmulas). Não se pode ter certeza de que ao se sair de casa com uma hora de antecedência, se chegará ao aeroporto em tempo de se embarcar no voo pretendido, mas apenas se pode afirmar que se tem uma chance de, por exemplo, 85% de não perder o voo! Não é possível prever (com exatidão) a ocorrência de incidentes que possam levar a perda do voo, tais como pneu furado, congestionamentos, acidentes, etc.

*Conhecimento incerto* é o conhecimento que não é indiscutível mas ao qual está associada alguma medida de incerteza que descreve crenças para as quais existem certas evidências de apoio [RIC 94]. As fontes de incerteza são: confiabilidade da informação, a imprecisão da linguagem de representação do conhecimento utilizada, informação parcial, e a agregação ou resumo das informações provenientes de múltiplas fontes [BON 90].

A incerteza devido a confiabilidade da informação pode estar presente em fatos ou regras. A incerteza nos fatos é devido a má definição de conceitos (do inglês, *fuzziness*), o que dificulta determinar a sua pertinência a um conjunto cujas fronteiras não são bem definidas; ou a aleatoriedade de medições decorrentes da não acuracidade ou baixa confiabilidade dos instrumentos utilizados. A incerteza nas regras é consequência de implicação fraca quando não é possível estabelecer uma forte correlação entre premissas e conclusão. O grau de implicação pode ser representado por um valor escalar (fator de certeza, probabilidade, etc.), sendo a implicação lógica transformada em uma implica-

ção plausível. Uma outra forma de representar o grau de implicação é através de quantificadores fuzzy tais como “a maioria”, etc., para expressar a cardinalidade relativa dos subconjuntos de elementos do domínio que usualmente satisfazem a propriedade implícita na regra de inferência. A incerteza pode ser composta através da agregação de fatos incertos na premissa; propagação de medidas de certeza para a conclusão; e consolidação das medidas associadas a uma conclusão que pode ser derivada por diversas regras.

A incerteza associada a imprecisão da linguagem de representação decorre do significado da regra não poder ser interpretado exatamente se não for utilizada uma linguagem formal. Face a fatos imprecisos e a regras com implicação fraca, se pode fazer inferências plausíveis com base na aplicação de uma generalização da regra de *modus ponens*.

A incerteza associada a informação parcial é geralmente modelada por formalismos de representação simbólica (i.é não numérica). Segundo Bhatnagar e Kanal [BHA 86], o conhecimento é *parcial* quando algumas respostas a questões relevantes não são conhecidas. O conhecimento é *aproximado* (baixa confiabilidade da informação ou a inerente imprecisão da linguagem de representação) quando as respostas são conhecidas mas não são precisas ou exatas. Exemplos:

- “Ayrton Senna morreu em um acidente automobilístico, ocasionado por uma falha mecânica, durante o Grand Prix de Imola, em 1994.” (parcial: não explicita qual foi a falha que ocorreu)
- “A morte do Senna foi provocada por uma falha no sistema de direção do seu Fórmula 1.” (aproximado: não explicita o tipo de falha)
- “Ao bater no muro de proteção, Senna sofreu uma desaceleração de cerca de 314 km/h para zero, em frações de milésimos de segundos.” (linguagem imprecisa)
- “Algumas fotos de amadores, feitas no dia da corrida, aparentemente mostraram um defeito na suspensão. Alguns pilotos e ex-pilotos acham que foi a barra de direção”. (informações conflitantes de fontes diversas)

A incerteza associada a agregação de informação de múltiplas fontes decorre dos seguintes problemas potenciais: a) quando os fatos possuem granularidade diversas, a medida de certeza a eles associados pode se transformar em uma medida de intervalo para o fato resultante (i.é o *fechamento* da representação não se mantêm); b) se os fatos são conflitantes, pode ser gerada uma contradição, que deverá ser detectada; c) a regra de combinação de evidências pode criar uma superestimação da medida de certeza se um procedimento de *normalização* for utilizado [BON 90].

Sandra Sandri [SAN 96] apresenta o conceito de *informação imperfeita*, normalmente conhecido como *incerteza*. Ela considera o termo incerteza muito restritivo, se aplicando melhor a um tipo específico de imperfeição. *Imprecisão, conflito, ignorância parcial*, etc., são outros exemplos de imperfeições. Assim, suponha que se deseja descobrir o horário de uma certa sessão de cinema. Algumas informações que se pode obter são:

- “A sessão começa às 20:15 h.” (informação perfeita)
- “A sessão começa entre 20:00 e 21:00 h.” (informação imprecisa)
- “A sessão começa próximo de 20:30 h.” (informação vaga)
- “Eu imagino que a sessão começa às 20:15 h.” (informação incerta)
- “É provável que a sessão comece às 20:00 h.” (informação probabilística)
- “É possível que a sessão comece às 20:00 h.” (informação possível)
- “Maria disse que a sessão começa às 20:00 h, mas João disse que começa às 21:00.” (informação inconsistente)

- “Eu não sei a que horas a sessão inicia, mas elas normalmente iniciam às 20:00 h.”  
(informação incompleta)

Para poder utilizar conhecimento em computador é necessário adquiri-lo e representá-lo. A aquisição do conhecimento consiste na obtenção deste a partir de fontes tais como especialistas humanos e/ou documentação tradicional (livros, etc.). Essa atividade é objeto da ação de um engenheiro do conhecimento e não será abordada neste trabalho. Representar conhecimento implica encontrar estruturas adequadas para expressar o tipo de conhecimento do domínio da aplicação, que facilitem resolver o problema em questão.

### 1.3 Representação do Conhecimento

Alguns autores definem *representação do conhecimento* como a formalização e estruturação de conhecimento em um computador, de forma que ele possa ser manipulado pelo sistema gerenciador da base de conhecimento. Barr [BAR 86, p. 143] a define como uma combinação de estruturas de dados e procedimentos de interpretação que, se usados de maneira correta em um programa, irão produzir um comportamento inteligente (observável através do conhecimento que demonstra).

#### 1.3.1 Fundamentos

Como vimos na discussão das idéias de Kirsh [KIR 91], os lógicos aceitam que a inteligência requer conhecimento declarativo e alguma forma de mecanismo tipo raciocínio, sendo o núcleo da IA o estudo das conceitualizações do mundo, pressupostas e usadas pelos sistemas inteligentes durante a *cognição* (processo computacional de raciocínio). Uma *habilidade cognitiva* é um mecanismo de controle (baseado em informação) que regula a performance em algum domínio. A unidade de conhecimento básica da cognição são as conceitualizações que suportam as habilidades cognitivas. Essas conceitualizações são idealizadas pelo projetista do sistema e representam um modelo o suficientemente preciso do mundo real para que o sistema possa atingir os seus objetivos. Um *conceito* é um componente (módulo) do conhecimento. Admitindo-se que cognição é inferência, aceita-se como evidente que as habilidades inteligentes são compostas de duas partes: a base de conhecimento declarativo e a máquina de inferência [KIR 91, p.6].

A fundamentação teórica para essas idéias foi apresentada por McCarthy e Hayes [McC 69], Newell e Simon durante conferência proferida na cerimônia de outorga do prêmio Turing 1975 da ACM - Association for Computing Machinery [NEW 76] e por Brian Smith [SMI 82 Apud BIT 96] [Smith, 1982 Apud LAK 94]. McCarthy e Hayes estabeleceram as bases epistemológicas<sup>1</sup> da IA, principalmente da área de representação do conhecimento [BIT 96, p.24]. Newell e Simon introduziram a Hipótese do Sistema de Símbolos Físicos e Brian Smith a Hipótese da Representação do Conhecimento.

McCarthy e Hayes, no artigo *Some Philosophical Problems from the Standpoint of Artificial Intelligence*, postulam que “um programa de computador capaz de agir inteligentemente no mundo deve possuir uma representação geral do mundo em termos da qual são interpretadas suas entradas” [BIT 96, p.24] [McC 69].

<sup>1</sup> Epistemologia é o estudo crítico dos princípios, hipóteses e resultados das ciências já constituídas e que visa a determinar os fundamentos lógicos, o valor e o alcance objetivo delas. Também é conhecida como a teoria do conhecimento e metodologia [FER 75], o estudo da natureza do conhecimento.

A representação de conhecimento pode ser analisada em termos da sua adequação aos fatos do mundo real e aos problemas que se propõe a solucionar. Essa adequação, segundo McCarthy e Hayes, é do tipo *metafísica*, *epistemológica* ou *heurística*.

Uma representação de conhecimento possui *adequação metafísica*<sup>2</sup> se um mundo construído de acordo com ela não apresentar contradições com os fatos e aspectos da realidade sendo representada. Possui *adequação epistemológica* se pode ser utilizada na prática para representar os fatos disponíveis sobre os aspectos de interesse da realidade.

Existem alguns exemplos de representações metafisicamente adequadas para representar conhecimentos científicos ou práticos a respeito do mundo físico mas que *não* são epistemologicamente adequadas para representar conhecimentos necessários para resolver problemas típicos. Como o contrário normalmente não ocorre, se dá mais ênfase ao estudo da adequação epistemológica em IA, em detrimento da metafísica. Segundo McCarthy [McC 77], a epistemologia na IA estuda os tipos de fatos disponíveis sobre o mundo, como podem ser representados em computador e que regras permitem a derivação de conclusões legítimas a partir deles.

Uma representação de conhecimento possui *adequação heurística* se os processos de raciocínio necessários para solução de problemas de interesse podem ser nela expressos. Esse tipo de adequação pode ser visto como uma medida da utilidade ou eficiência computacional da representação.

Para Newell e Simon [NEW 76, p.115] a habilidade para armazenar e manipular símbolos é um requerimento para a ação inteligente. Esses autores postularam que "o sistema de símbolos físicos tem os meios necessários e suficientes para a ação inteligente em geral" [NEW 76, p.116]. Essa hipótese, denominada *Hipótese do Sistema de Símbolos Físicos*, não pode ser comprovada ou contestada em termos lógicos, sendo sujeita à validação empírica, através da experimentação. Sua importância reside no fato de "formar a base de crença de que é possível criar-se programas de computador (sistemas simbólicos) que executam tarefas inteligentes, que no momento são realizadas por pessoas" [RIC 94, p.8]. Um *sistema de símbolos físicos* é composto de símbolos, expressões (instâncias de símbolos) e processos sobre expressões que permitem a modificação, criação, reprodução e destruição de expressões. Pode ser visto como uma máquina que produz, no tempo, uma coleção progressiva de estruturas de símbolos.

McCarthy, Hayes, Newell e Simon enfocaram símbolos e suas representações. Brian Smith enfoca a *semântica* ou seja o relacionamento entre o mundo real (com operadores para manipulação de seus elementos) e a representação (e operadores). O significado das expressões utilizadas na representação de conhecimento pode ser [BIT 96, p31]:

- a) atribuído intuitivamente, a partir dos símbolos utilizados (*semântica informal*);
- b) definido pelo comportamento dos programas que as manipulam (*semântica procedimental*);
- c) estabelecido através de descrições precisas, em linguagem natural, da sua correspondência com o mundo a representar (*semântica descritiva*);

---

<sup>2</sup> Metafísica é a parte da Filosofia que procura determinar as regras fundamentais do pensamento - aquelas de que devem decorrer o conjunto de princípios de qualquer outra ciência, e a certeza e evidência que neles reconhecemos - e que nos dá a chave do conhecimento do real, tal como este verdadeiramente é (em oposição a aparências) [FER 75].

- d) associados a expressões de uma linguagem com semântica estabelecida, como por exemplo a Lógica (*semântica de equivalência*).

A semântica informal é muito imprecisa e a procedimental tem utilidade limitada pois a semântica de programas já é complexa *per si*. Elas foram utilizadas apenas nos primeiros sistemas de IA. A partir do início dos anos 80 tornou-se um consenso que todo sistema de representação de conhecimento deve possuir descrição semântica por equivalência.”

A semântica de equivalência é obtida aplicando-se os seguintes passos:

- a) considerar o mundo real como formado por um conjunto de objetos arbitrários e identificáveis, com operadores usuais da teoria dos conjuntos;
- b) definir a representação de conhecimento através de linguagem (lógica) formal, com operadores definidos sintaticamente;
- c) definir a semântica da representação através de uma função que associa um significado matemático (da teoria dos conjuntos) à cada expressão da linguagem de representação.

Tal abordagem permite que os resultados da lógica, baseados na *teoria dos modelos*, sejam utilizados para analisar a relação entre a representação de conhecimento e o mundo abstrato. A teoria dos modelos é a parte da lógica que estuda os valores verdade. Um modelo de uma expressão lógica bem formada é uma interpretação que a satisfaz. Para uma descrição mais completa sobre o enfoque da teoria de modelos veja [BIT 96]

Em 1982 Brian Smith formulou a *Hipótese de Representação de Conhecimento* [BIT 96, p.32]: “qualquer processo inteligente realizado por uma máquina, será formado por ingredientes estruturais que: a) serão naturalmente percebidos pelos observadores externos como uma descrição *proposicional* do conhecimento exibido pelo processo, e b) independentemente de tal atribuição semântica externa, terão um papel formal, *causal* e essencial na geração do comportamento que manifesta tal conhecimento”.

Pela primeira parte dessa hipótese, o conhecimento é representado declarativamente em alguma linguagem tipo lógica e, pela segunda parte, essa representação desempenha um papel causador do comportamento do sistema. Assim todo processo inteligente tem um ingrediente (em geral denominado *interpretador*) capaz de manipular as representações (declarativas) de conhecimento de forma a gerar um comportamento inteligente.

Os primeiros sistemas de IA visavam a busca de métodos gerais de solução de problemas. Eles utilizavam uma *representação declarativa* de conhecimento baseada na lógica, a qual era defendida por pioneiros como McCarthy, Hayes e Newell, etc. Esse enfoque vê programas e seus dados como sentenças lógicas sobre o mundo e utiliza um processo de inferência para obter conseqüências explícitas dessas sentenças (veja 1.3.2).

Com a experiência adquirida no desenvolvimento dos primeiros sistemas de IA, constatou-se que grande parte do conhecimento necessário para solucionar problemas é específico ao domínio da aplicação em questão e que é bastante difícil construir programas gerais eficientes. Em decorrência surgiu a *representação procedimental*, defendida por Marvin Minsky e outros (veja 1.3.2). Nela as declarações são vistas como representações procedimentais de conhecimento e incorporam as informações de controle necessárias. Essa representação deve ser ampliada com um interpretador que segue as instruções (infor-



mações de controle) contidas no conhecimento. Esse enfoque visa a abordagem de problemas específicos em mundos simples (micromundos).

Segundo Bittencourt [BIT 96, p.33], “a hipótese da representação de conhecimento é um compromisso entre essas duas posições: manter, em relação à representação do mundo, a modularidade, generalidade e flexibilidade do enfoque declarativo, enquanto que as preocupações de eficiência do enfoque procedural determinam a construção do componente interpretador.”.

A discussão sobre qual das duas abordagens é a melhor acabou se esvaziando com a adoção da representação declarativa de conhecimento, conjugada com um interpretador, característica dos sistemas especialistas.

De forma análoga a aplicação da metodologia de separação de conhecimento declarativo e mecanismo interpretador, induzida pela hipótese da representação de conhecimento, Smith conclui, que “então também deve ser possível construir um processo que raciocine sobre si mesmo, a partir de um processo interpretador similar e representações formais sobre a estrutura de suas representações e sobre sua operação”. Tal suposição constitui a *Hipótese da Reflexão*.

Obviamente que um sistema de IA com capacidade de raciocinar sobre as suas próprias capacidades e limitações (*capacidade de introspeção*) terá melhores condições para solucionar problemas complexos ou explicar as soluções a que chegou. Para tanto é necessário representar conhecimento sobre conhecimento (representação interna), o qual é denominado *metaconhecimento*. Portanto a representação de conhecimento deve suportar representação das suas próprias estruturas internas ou apresentar diversos níveis de representação (cada um capaz de representar conhecimento sobre o nível inferior), o que a torna bem mais complexa devido ao aparecimento de problemas de recursividade e autoreferência.

Lakemeyer e Nebel [LAK 94] realçam o estabelecimento do consenso, nos anos 90, de que a pesquisa na área de representação de conhecimento assume uma representação *explícita e declarativa*. Explícita porque o conhecimento representado é armazenado em uma base de conhecimentos que consiste de um conjunto de entidades formais que o descrevem de uma forma direta e não ambígua. Declarativa porque o significado formal da representação pode ser especificado sem referência a como o conhecimento é aplicado de uma forma procedimental, implicando o uso de alguma metodologia lógica.

Essa premissa permite distingui a área de outras que lidam com outras formas de representação de conhecimento tais como linguagens de programação (conhecimento codificado através de procedimentos na forma de código de programas) ou redes neurais (conhecimento representado implicitamente em padrões de ativação de uma rede neural).

### 1.3.2 Sistemas de Representação de Conhecimento.

Um bom sistema deve possuir *adequação representacional* (capacidade de representar os tipos de conhecimento necessários no domínio), *adequação inferencial* (capacidade de manipular as estruturas de forma a derivar novas, a partir de conhecimentos antigos), *eficácia inferencial* (capacidade de agregar informações adicionais à estrutura de conhecimento existente, para direcionar os mecanismos de inferência para direções mais pro-

missoras) e *eficácia aquisitiva* (capacidade de acomodar facilmente novas informações na base de conhecimentos).

Não foi desenvolvido, até o momento, um único sistema que otimize todas essas quatro capacidades para todos os tipos de conhecimento [RIC 93].

Segundo Mattos [MAT 91] são várias as formas de representação de conhecimento existentes, algumas mais adequadas para esta ou aquela tarefa. Não existe métrica formal para medir a adequação de um esquema de representação para um determinado problema [BAR 86]. Basicamente existem três abordagens para se representar o conhecimento: *declarativa (ou lógica)*, *procedimental* e a *estrutural*.

No sistema de *representação declarativo* o conhecimento é descrito através de fatos, postulados como verdadeiros, sobre o domínio e por um conjunto de procedimentos para manipulá-los. A base de conhecimentos é vista como uma teoria ou um conjunto de teoremas numa teoria. Esse sistema possui semântica bem definida e disponibilidade de uma teoria de demonstração, sendo adequado para representar porções de conhecimento *passivas* do domínio, e em situações onde o raciocínio (monotônico) tenha importância.

Observe que o conhecimento é especificado sem fornecer o uso que será feito dele. Essa representação incorpora *aspectos estáticos* do conhecimento, i.é fatos sobre objetos, eventos, seus relacionamentos e estados do mundo. Os *declarativistas* enfatizavam a flexibilidade, economia e completude da representação declarativa, a veracidade das inferências lógicas e a modificabilidade (facilidade para alteração, mantida a consistência) das bases de conhecimentos declarativos.

Esquema de representação lógica e redes semânticas são exemplos desse sistema.

No *sistema de representação procedimental* o conhecimento é descrito através de procedimentos para a sua utilização, sendo adequado para representação de conhecimento *comportamental* da aplicação.

*Procedimentalistas* argumentavam que os sistemas inteligentes necessitavam saber como usar seu conhecimento, i.é como encontrar fatos relevantes, fazer inferências, etc, sendo esses aspectos do comportamento inteligente melhor capturados em procedimentos. Eles enfatizavam a objetividade das inferências feitas (pois usavam heurísticas específicas para o domínio da aplicação de forma a evitar linhas de raciocínio irrelevantes ou não naturais) e a facilidade de codificação e entendimento do processo de raciocínio da aplicação.

Regras de produção são um exemplo de esquema de representação de conhecimento procedimental.

No *sistema de representação estrutural* o conhecimento é descrito com objetos e relações sobre as entidades a serem modeladas. São utilizados conceitos, hierarquias de estruturas, descrições de classes de elementos e instâncias individuais ou componentes de objetos. Uma base de conhecimentos é uma coleção de objetos e suas relações. Esse sistema enfatiza os aspectos estruturais (organizacionais) do domínio, sendo adequado para situações com objetos protótipos, objetos tipados e raciocínio por analogia e estatístico ou simplesmente para propósitos de organização da base de conhecimentos.

*Frames* e *scripts* são exemplos de esquema de representação de conhecimento estrutural.

A maioria das aplicações reais e dos esquemas de representação de conhecimento empregam combinações desses sistemas.

Independente do sistema de representação de conhecimento escolhido é necessário decidir qual o *escopo* e a *granularidade* da representação, ou seja “qual porção do mundo externo pode ser representada no sistema?” e “em que nível de detalhes?” ou “quais devem ser as primitivas de representação de conhecimento a usar?”. Deve haver um pequeno número de “primitivas de baixo nível” ou deve haver um grande número de primitivas que cobrem uma faixa de granularidades?

Tais questões não são fáceis de serem respondidas e dependem totalmente da aplicação particular que se deseja implementar, como mostram Barr e Feigenbaum com o exemplo seguinte. Esquema de representação de conhecimento baseado em *lógica* pode ser de granularidade extremamente fina para um programa de raciocínio matemático mas resulta em uma simplificação grosseira para um programa de visão computacional.

A granularidade está associada com a performance desejada. Em geral, a resposta a questão “tudo que o sistema deve conhecer pode ser representado com este esquema de representação de conhecimento?” é sim, mas algumas coisas são representadas mais facilmente do que outras [BAR 86].

Para Rich e Knight [RIC 94], a vantagem de converter todas as informações em uma representação em termos de um pequeno conjunto de primitivas é que os procedimentos utilizados para derivar inferências a partir do conhecimento representado só precisam ser escritas em termos das primitivas, e não em termos das muitas maneiras nas quais o conhecimento possa ter aparecido originalmente.

As desvantagens para o uso de primitivas de baixo nível são:

- a) fatos simples de alto nível podem exigir grande quantidade de armazenamento, quando divididos em primitivas, devido ao maior detalhamento inerente às primitivas de baixo nível;
- b) o conhecimento (adquirido de especialista ou de documentação) é inicialmente apresentado ao sistema em uma forma de nível relativamente alto como a linguagem natural, logo há a necessidade de um trabalho adicional para reduzi-lo ao formato de primitivas;
- c) em muitos domínios não há segurança sobre o que as primitivas devam ser.

Assim, quanto mais baixo o nível escolhido para as primitivas, menor a inferência exigida para raciocinar com ele em alguns casos, mas maior a inferência exigida para criar a representação do conhecimento a partir da linguagem natural e maior o espaço de armazenamento necessário, já que muitas inferências serão representadas muitas vezes.

### 1.3.3 Fatores para Seleção de um Esquema de Representação de Conhecimento

Ahuja[Ahuja, 1987 Apud PAS 91]<sup>3</sup> apresenta os seguintes critérios subjetivos de seleção de esquemas:

---

<sup>3</sup> KES: A Non-production Rule Environment for Building Expert Systems. **International Workshop on Expert Systems & Their Applications**, 7. Avignon: Proceedings, p. 769-791, May 1987.

- a) se existir um formato natural de representação do conhecimento utilizado por especialista, no domínio específico da aplicação, utilize-o. A utilização desse *formato pré-codificado do conhecimento* torna mais rápida a construção de bases de conhecimento, pois não é necessária nenhuma transformação de representação para armazenar o conhecimento;
- b) selecione a representação de conhecimento que possa produzir resultados que melhor refletem o *tipo de inferências* que normalmente será feito no problema específico. Algumas aplicações apresentam decisões que são probabilísticas por natureza (como a previsão do tempo), outras apresentam basicamente decisões categóricas (i.e. relativas a classificação em categorias), enquanto outras aplicações necessitam tanto decisões probabilísticas quanto decisões categóricas;
- c) dependência das inferências, do contexto (de difícil avaliação na prática).

Em sistema com *raciocínio dedutivo* baseado em regras de produção, os antecedentes de cada regra especificam o contexto no qual a regra pode ser utilizada. Se o problema envolve a consideração de um contexto denso, antes de se tomar uma decisão, então será complexo escrever um conjunto de regras que o descreva, pois os antecedentes de cada uma delas teriam que refletir o contexto no qual ela é aplicável. É mais adequado utilizar *raciocínio abdutivo* e representação de conhecimento declarativo.

#### 1.4 Raciocínio

O processo de raciocínio associado a um esquema de representação de conhecimento permite que se chegue a conclusões adequadas a partir do conhecimento e de regras gerais disponíveis.

O *raciocínio* é entendido como o processo pelo qual, de proposições conhecidas ou assumidas, se chega a outras proposições a que se atribuem graus variados de verdade [FER 75]. Portanto refere-se a atividades tais como: tirar conclusões a partir de um conjunto de fatos, diagnosticar causas possíveis para uma condição, gerar hipóteses acerca de uma situação, analisar os fatos e os dados de um problema, demonstrar um teorema, resolver um problema e argumentar.

Os sistemas convencionais de IA lidam com conhecimento certo e utilizam o chamado *raciocínio monotônico*, baseado nas lógicas clássicas, caracterizado por:

- a) os fatos necessários à solução de um problema estão presentes no sistema ou podem ser deriváveis dos fatos presentes através do uso dos axiomas e regras de inferência da lógica de primeira ordem;
- b) os fatos (e sua representação) são consistentes; e
- c) a medida que novos fatos se tornam disponíveis, se eles forem consistentes com todos os outros fatos já definidos, nenhum dos fatos existentes será alterado.

Assim em sistemas de raciocínio monotônico não existem mecanismos de revisão de crenças; o sistema aumenta o estoque de verdade a medida que conhecimento é acrescentado e as inferências são realizadas. Essa propriedade é chamada de *monotonicidade*.

Os sistemas de *raciocínio não-monotônico* são aqueles que lidam com conhecimento que não satisfazem a qualquer uma das características acima. Os axiomas e as regras de inferência da lógica de primeira ordem são estendidos para permitir o raciocínio com informações *incompletas*, mas preservam a propriedade de que, em qualquer momento, uma proposição poder ser considerada verdadeira, falsa ou nem verdadeira ou falsa. In-

ferências dependentes da falta de algum conhecimento, são denotadas *inferências não-monotônicas*. A base de conhecimentos não cresce monotonicamente pois a adição de uma asserção pode invalidar uma inferência que dependia da ausência daquela asserção.

O raciocínio não-monotônico é chamado de *raciocínio derrotável* porque uma inferência não-monotônica pode ser invalidada (derrotada) pela adição de novas informações que violem suposições feitas durante o processo de raciocínio original. Quando um fato novo é acrescentado à base de conhecimentos ou um fato antigo é removido, podem ocorrer contradições entre fatos existentes, sendo necessário um mecanismo para lidar com tais conflitos.

São considerados raciocínios não-monotônicos o raciocínio *default*, o raciocínio minimalista, a abdução e o mecanismo de herança.

O raciocínio *default* permite tirar conclusões baseado naquilo que tem mais chance de ser verdadeiro, na ausência de informação contrária. Utiliza expressões que são válidas em *geral*, mas é capaz de reconhecer e assimilar *exceções*, quando necessário. O exemplo clássico é, ao sabermos que Tweety é um pássaro, temos evidência (não conclusiva) de que ele pode voar. Então concluímos que ele voa, pois não temos nenhuma informação contrária a esse respeito. Posteriormente, ao descobrirmos que Tweety é um pinguim, podemos rever nossa posição e concluir que ele não pode voar.

O raciocínio *minimalista* está associado a idéia de que se algo é verdadeiro e relevante, faz sentido assumir que foi inserido na base de conhecimentos ou pode ser derivado a partir dela. As únicas declarações verdadeiras são aquelas que precisam ser verdadeiras para que seja mantida a consistência da base de conhecimentos. Pode ser implementado através da Hipótese do Mundo Fechado e de circunscrição (Capítulo 8).

A *abdução* é um dos tipos de raciocínio formal a ser visto nessa seção, e consiste em, dadas duas fbf,  $(A \rightarrow B)$  e  $(B)$ , assumir  $A$ , se for consistente. Essa conclusão não é obtida aplicando-se as regras normais da lógica de predicados e pode estar errada, mas representa o melhor palpite que se pode ter sobre o que está acontecendo.

Nos sistemas de *raciocínio estatístico* a representação do conhecimento é ampliada para englobar medidas que descrevem níveis de evidência e crença. Assim, um certo tipo de medida numérica de certeza (em vez da simples avaliação ‘verdadeiro’ ou ‘falso’) é associado a cada proposição. Esses sistemas lidam com conhecimento incerto.

Foram identificados os seguintes processos de raciocínio: formal, procedimental, analógico, por abstração e generalização, e em meta-nível [BAR 86] [BIT 96].

#### a) Raciocínio Formal

Consiste na manipulação sintática de estruturas de dados visando obter novas estruturas, segundo regras de inferências pré-definidas. Em IA o raciocínio formal é aplicado em esquemas de representação de conhecimento baseados na lógica, i.é ao sistema de representação declarativo. Os processos de *dedução*, *indução* e *abdução* constituem os tipos de raciocínio formal possíveis, sendo que apenas o primeiro é formalmente correto. Os outros dois processos dependem do domínio e podem produzir resultados inválidos.

No *raciocínio dedutivo* conclui-se novos fatos através de inferências baseadas em uma teoria formal que descreve o domínio da aplicação. O sistema formal, composto pela teoria (axiomas e regras de inferência), constitui o conhecimento prévio. As conclusões a que se chega, com a aplicação das regras de inferência e os axiomas, são conseqüências lógicas, forçosamente verdadeiras. O sistema raciocina por inferência dedutiva, aumentando o seu conhecimento, por aplicação das regras de inferência ao conhecimento inicial. Os tipos de inferência utilizados são *modus ponens*, *modus tollens*, *silogismo hipotético*, *especialização*, *generalização* e *resolução*.

No *raciocínio indutivo*, face a um conjunto de exemplos ou dados particulares (*fatos experimentais*), o sistema procura inferir conceitos ou leis gerais (i.é construir conjecturas genéricas a partir de fatos simples). Esse tipo de raciocínio é estudo na área de aprendizagem de máquina.

O exemplo clássico de raciocínio indutivo é: a partir da existência de um conjunto grande de corvos, todos eles negros, é razoável supor que todos os corvos são negros.

O *raciocínio por abdução* é um mecanismo que permite raciocinar a partir dos efeitos para as possíveis causas. Assim se um determinado efeito sugere uma causa, *na falta de informação contrária*, é razoável supor que o efeito foi determinado pela causa sugerida. A confiança nessa hipótese pode estar baseada no número de outras evidências que podem comprovar o efeito.

Assim, se sabemos que ‘se local da cultura for o sangue, a morfologia do organismo for bastonete, a mancha de gram do organismo for gram-positiva e o doente está em perigo então a identidade do organismo (causador da infecção) é *Pseudomonas-aeruginosa*’ e desejando provar que o causador da infecção é a bactéria *Pseudomonas-aeruginosa*, admitimos, por hipótese, que seja ela. Portanto agora devemos provar que o local da cultura é o sangue, a morfologia do organismo é bastonete, a mancha de gram do organismo é gram-positiva e que o doente está em perigo! Esse tipo de raciocínio tem sido útil na construção de sistemas de diagnóstico médico [CHA 85].

Raciocínio dedutivo constitui um exemplo de raciocínio monotônico. Até o início dos anos 70 dedução era o principal foco de atenção quando se tratava de mecanismos de inferência. Entretanto tornou-se claro que uma grande quantidade de raciocínio de senso comum (tais como analogias, visão, herança de características, etc.) não são de natureza dedutiva. A partir de então foi dada ênfase a outros tipos de raciocínio, dentre eles a abdução (em sistemas especialistas), o não-monotônico, estatístico, etc.

#### b) Raciocínio Procedimental

Esse tipo de raciocínio usa simulação para responder questões e resolver problemas. É aplicado a esquemas de representação de conhecimento que usam representação procedimental. Portanto pode ser utilizado sempre que existir um algoritmo visando determinar uma solução para o problema envolvido no processo de inferência corrente. Normalmente esse tipo de raciocínio é melhor utilizado em micromundos.

Como visto, os *procedimentalistas* argumentam que as inferências poderiam ser mais diretas, caso fossem usadas informações de controle, as quais consistiam em heurísticas específicas para o domínio da aplicação, destinadas a evitar linhas de raciocínio não

naturais ou irrelevantes. Com a abordagem procedimental torna-se mais fácil codificar conhecimento e entender o processo de raciocínio da aplicação.

Para Bittencourt [BIT 96] as propriedades geométricas e matemáticas de elementos do domínio da aplicação e as relações espaciais entre eles podem ser obtidas via inferências através do raciocínio procedimental. Também as inferências associadas às representações analógicas de objetos físicos são mais adequadas ao raciocínio procedimental.

#### c) Raciocínio Analógico

Consiste no processo de determinar relações entre dois domínios visando adaptar e aplicar o conhecimento e os métodos aplicáveis num domínio, em tarefas semelhantes, no outro domínio. Pressupõe a existência de conhecimento prévio sobre problemas e suas respectivas soluções em domínios diversificados, o que em geral é fornecido por um “professor”. A analogia consiste no mapeamento entre elementos de um domínio fonte em um domínio alvo.

Embora pareça ser um modo bastante natural de raciocínio para as pessoas, é de difícil formalização e aplicação em sistemas de IA. Segundo Bittencourt [BIT 96, p.30], raciocínio analógico foi aplicado com relativo sucesso na demonstração automática de teoremas, sendo que “a forma restrita das expressões permitidas e a estrutura formal das provas facilitam a definição de medidas de analogias e sua utilização para guiar demonstrações (de teoremas) baseadas em outras demonstrações conhecidas”.

Também é adequado para uso em conjunto com esquemas de representação *analógicos* (também denotados *diretos*). Representações diretas são definidas como esquemas nos quais “propriedades e relações entre partes da configuração representante, reproduzem propriedades e relações das partes da configuração complexa sendo representada, de tal forma que a estrutura da representação fornece informações sobre a estrutura do que está sendo reproduzido” [BAR, p.200]. Mapas, modelos, diagramas, imagens, etc. são exemplos de representações analógicas.

#### d) Raciocínio por Abstração e Generalização

Consiste na geração de conceitos mais amplos (conceito de classe) a partir de exemplos típicos (processo de generalização) e de contra-exemplos (processo de especialização). Patrick Winston [WIN 84] apresenta um interessante exemplo, no *mundo dos blocos*, onde o programa recebe um conjunto de exemplo de arcos (cada qual composto por um bloco suportado por dois blocos separados entre si por uma certa distância) e um conjunto de contra-exemplos de arcos. De posse dessa informação, o programa generaliza o conceito de arco, introduzindo relações necessárias (p. ex. blocos de suporte devem ser separados por uma distância não nula) e relações opcionais (p. ex. bloco suportado pode ser um paralelepípedo ou uma pirâmide), de forma a explicar todas as informações (exemplos e contra-exemplos) disponíveis.

Esse tipo de raciocínio é natural para as pessoas. Por exemplo pode ser bastante natural que uma pessoa, sabendo que ‘sabiás têm asas’, ‘canários têm asas’ e ‘bem-te-vis têm asas’, acreditar que todos os pássaros têm asas. No entanto é muito difícil de ser descrito em detalhes suficientes para permitir uma implementação em programa de computador.

Esse método é utilizado principalmente na área de aprendizagem de máquina em micromundos tais como no mundo dos blocos.

#### e) Raciocínio de Meta-nível

Consiste no raciocínio com base em metaconhecimento, i.e informações sobre utilidade e disponibilidade do conhecimento sobre o domínio da aplicação. Esse tipo de raciocínio também parece ser natural em pessoas, mas apresenta problemas para implementação em programas de computador, conforme visto na discussão sobre introspecção apresentada na seção 1.3.1.

### 1.5 Conclusão

Para Mattos [MAT91], cada esquema de representação de conhecimento é caracterizado pelo conjunto de construtores para representar modelos do domínio e pelos operadores disponíveis. Esses construtores refletem a expressividade dos diferentes esquemas. Eles possuem sintaxe definida, para descrever o domínio, e alguma semântica associada. A potência desta semântica é o que diferencia os bons esquemas dos fracos. Ela define a forma de modelar o domínio da aplicação, usando o esquema em questão.

Não existe uma teoria geral de representação de conhecimento que permita uma visão global e comparação entre os diversos esquemas de representação. Portanto a adequação heurística (eficácia) de cada tipo de esquema de representação só pode ser analisada no contexto de uma aplicação [BIT 96] [BAR 86].

A partir dos anos 90 a pesquisa na área de representação de conhecimento assume uma representação *explicita* (i.e o conhecimento representado é armazenado em uma base de conhecimentos que consiste de um conjunto de entidades formais que o descrevem de uma forma direta e não ambígua) e *declarativa* (i.e o significado formal da representação pode ser especificado sem referência a como o conhecimento é aplicado). A linguagem de representação utilizada é baseada na lógica, onde as sentenças podem ser interpretadas como proposições sobre o mundo [LAK 94].

Em geral, “se o processo de raciocínio é requerido ser computacionalmente tratável, nós somos freqüentemente forçados a restringir a expressividade da linguagem de representação (p.ex. eliminando algum operador ou restringindo a estrutura das sentenças) ou ceder na acuracidade da resposta (p.ex. utilizando raciocínio parcial ou limitando o tempo para processamento de uma inferência)” [Levesque e Brachman 1987 Apud GER 94].

Pesquisas em IA, visando obter comportamento inteligente, estão muito correlacionadas com o uso de raciocínio dedutivo e do raciocínio plausível. *Raciocínio dedutivo* é o baseado em métodos dedutivos. *Raciocínio plausível* é o raciocínio que conduz a conclusões incertas porque seus métodos são falíveis ou suas premissas são incertas [SHA 90]. Em geral utiliza argumentos baseados em raciocínio não-monotônico, analogias, indução, abdução ou probabilidades, sendo indicado para o tratamento de conhecimento incerto.



## 2. Esquema de Representação Baseado na Lógica

Esse esquema de representação do conhecimento utiliza o sistema de representação de conhecimento declarativo. Pode ser utilizado qualquer tipo de lógica (de primeira ordem ou superior, multi-valorada, modal, fuzzy, etc.) mas o uso da lógica de predicados de primeira ordem é mais difundido [MAT 91].

A lógica foi desenvolvida por filósofos e matemáticos como um processo de desenvolver inferências a partir de fatos.

O esquema a ser visto é o baseado na lógica de predicados (também denominada lógica de primeira ordem). Para uma descrição de outras lógicas, veja [BIT 96] ou [TUR 84].

Essa lógica utiliza os conectivos  $\wedge$  (e),  $\vee$  (ou),  $\rightarrow$  (se então),  $\leftrightarrow$  (se e somente se),  $\neg$  (não), além dos quantificadores *universal*  $\forall$  (para todo) e *existencial*  $\exists$  (existe um).

Com o uso dos conectivos lógicos podemos formular sentenças, assumindo valores verdadeiro ou falso, que capturam parte da estrutura lógica de trechos de discurso e caracterizam fatos ou situações do domínio. Com o uso dos quantificadores podemos representar afirmações sobre indivíduos e suas propriedades ou relações, ou seja afirmações sobre relacionamentos entre objetos e generalizações destes relacionamentos para classes de objetos.

A representação de conhecimentos através da lógica de predicados difundiu-se em IA através de duas implementações [ABE 94]: a) a linguagem PROLOG, a qual implementa um subconjunto do cálculo de predicados através de *conhecimento procedimental* (programas PROLOG - veja Capítulo 3) e b) os provadores automáticos de teoremas, os quais implementam a prova da verdade ou falsidade de sentenças, confrontado-as com o banco de fatos.

Os fatos são descritos como predicados e argumentos. Os *predicados* representam os relacionamentos entre os objetos ou o nome dos atributos dos objetos. Os objetos a serem representados (ou os valores dos atributos) são fornecidos como argumentos dos predicados. Por exemplo, as sentenças clássicas “todos os homens são mortais” e “Sócrates é homem” podem ser representadas como:

1.  $\forall x (\text{Homem}(x) \rightarrow \text{Mortal}(x))$
2.  $\text{Homem}(\text{Sócrates})$

onde  $\forall$  é o quantificado universal, Homem e Mortal são predicados, x é uma variável e Sócrates é uma constante.

A vantagem do uso da representação formal é que existe na lógica um conjunto de regras de inferências que permite, a partir de fatos conhecidos como certos, derivar outros fatos que serão certos, assim como testar a verdade de qualquer nova afirmação contra os fatos que já são conhecidos como certos [BAR 86]. Por exemplo, podemos aplicar as regras de inferências *especialização* e *modus ponens* para inferir, a partir das premissas acima, que “Sócrates é mortal”, visto que é homem e todos os homens são mortais.

Uma *regra de inferência* é uma função sintática que, dado um conjunto de fórmulas lógicas, gera uma nova fórmula. As fórmulas abaixo, onde A e B são fórmulas quaisquer, são exemplos de regras de inferências comumente utilizadas:

<i>Modus Ponens:</i>	$(A \wedge (A \rightarrow B)) \rightarrow B$
<i>Modus Tollens:</i>	$(\neg B \wedge (A \rightarrow B)) \rightarrow \neg A$
<i>Silogismo Hipotético:</i>	$((A \rightarrow B) \wedge (B \rightarrow C)) \rightarrow (A \rightarrow C)$
<i>Especialização:</i>	$\forall x(A) \rightarrow A\{x/a\}$
<i>Generalização:</i>	$A\{x/a\} \rightarrow \exists x(A),$

onde  $A\{x/a\}$  significa a fórmula obtida pela substituição, na fórmula A, de todas as ocorrências da variável x por uma constante a.

Um dos motivos para a popularização do uso de representações baseadas na lógica em aplicações de IA é que a inferência de novos fatos a partir dos fatos antigos pode ser mecanizada. Diversos provadores automáticos de teoremas foram desenvolvidos para determinar a validade de uma nova asserção em uma base de conhecimento de fórmulas lógicas através de tentativas de prová-las a partir dos fatos armazenados na base.

Se existirem mais de uma regra de inferência para provar novos fatos a partir dos antigos, a escolha de qual delas usar, em um determinado passo, pode se tornar um grande obstáculo para a construção de um sistema mecânico de prova.

Uma forma de tratar o problema em questão é utilizar o *método da resolução* onde existe apenas uma regra de inferência [CAS 87]. A idéia básica é colocar todas as fórmulas na **forma clausal** (lista de predicados, negados ou não, conectados pelo conectivo *ou*). Casanova [CAS 87] e Rich [RIC 93] apresentam uma demonstração e um algoritmo para converter uma fórmula qualquer em sua notação clausal (representada através de uma disjunção finita de fórmulas atômicas). Após a aplicação do algoritmo de conversão a um conjunto de fórmulas, ele produz um conjunto de cláusulas (formas clausais) equivalentes, constituindo uma grande conjunção, pois todas elas devem ser simultaneamente verdadeiras. Esta conjunção de disjunções é denominada *forma normal conjuntiva*.

A forma típica de resolução é o método chamado de *resolução por refutação*. A proposição desejada é negada e adicionada ao conjunto de fórmulas, criando um novo conjunto. As fórmulas são convertidas para um conjunto de cláusulas na forma normal conjuntiva. Tentamos provar que essa inclusão leva a uma *contradição* (i.é as cláusulas do conjunto não serem simultaneamente verdadeiras). Para tanto seleciona-se duas cláusulas que contém a mesma fórmula atômica, uma na forma positiva e outra na forma negada. As duas cláusulas são combinadas (através da disjunção) para formar uma nova cláusula sem o par de fórmulas atômicas (que é cancelado). Se a nova cláusula for vazia, há uma contradição, caso contrário ela é incorporada ao conjunto de cláusulas e o processo de seleção continua [RIC 93]. Este algoritmo não apresenta garantia de parar quando o que estiver sendo demonstrado não for um teorema, logo é *semi-decidível*.

## 2.1 Vantagens e Desvantagens

As vantagens do esquema de representação de conhecimento lógico, segundo [BAR 86], [GEN 85] e [MAT 91], são:

- a) é uma representação formal que possui um conjunto de regras de inferência através do qual, a partir dos fatos iniciais se pode derivar outros fatos (que não estão explicitamente representados);
- b) se pode garantir a verdade dos novos fatos deduzidos formalmente, se forem verdadeiros os fatos iniciais (raciocínio monotônico);
- c) permite crescimento incremental. Teoricamente a base de conhecimentos pode ser mantida logicamente consistente e todas as conclusões podem ser garantidas corretas;
- d) o *fecho semântico* de um conjunto de declarações lógicas (i.e o conjunto de inferências e conclusões que podem ser derivadas) é completamente especificado pelas regras de inferências;
- e) capacidade de explicação: a lógica apresenta-se como uma forma natural de expressar certas noções, sendo que a representação de um problema em lógica geralmente corresponde à forma intuitiva de compreensão do domínio;
- f) o método de resolução é *completo*, visto que, a partir das regras de inferência, todas as conclusões possíveis logicamente implicadas pelo conjunto de fatos iniciais, podem ser provadas;
- g) possui semântica formal bem entendida e aceita e uma notação sintática simples.

As desvantagens do uso do esquema de representação lógico são:

- a) inadequado para:
  - inferir a partir de dados incertos, valores relativos tipo “muito quente”, graus de certeza, crenças diferentes;
  - generalizar seu conhecimento (raciocínio indutivo) a partir dos fatos iniciais;
  - raciocinar analogicamente;
- b) a lógica de predicados não fornecer pistas quanto à localização dos fatos que serão utilizados na prova de um resultado desejado, pois apresenta dificuldades para representar conhecimento *procedimental* e/ou *heurístico*. No caso do método da resolução, não sabemos previamente que cláusulas devem ser resolvidas contra outras, restando a opção de tentar todas as combinações, o que pode levar a uma *explosão combinatorial* do número de possibilidades, conforme novos fatos são armazenados na base de conhecimentos;
- c) separa representação do conhecimento e processamento, sendo que a principal dificuldade consiste em determinar como os fatos armazenados na base de conhecimentos podem ser usados, e não como devem ser armazenados;
- d) é potencialmente ineficiente: se não se prestar atenção a como as cláusulas serão utilizadas, elas podem ser escritas em uma ordem que torna a derivação de conclusões, computacionalmente complexa;
- e) falta de facilidades para estruturação do conhecimento, dificultando o uso desse esquema para grandes bases de conhecimento.

Um exemplo de sistema inteligente que utiliza representação baseada em lógica de predicados de primeira ordem é o STRIPS - Stanford Research Institute Problem Solver, projetado resolver problemas de planejamento enfrentados por um robô, ao rearranjar objetos e mover-se em um ambiente desordenado [Fikes, Hart, and Nilsson (1972) Apud BAR 86]. Outro exemplo é o FOL [Filman and Weyhrauch (1976) Apud BAR 86], um verificador de provas para provas em lógica de primeira ordem.

### 3 Regras de Produção

Esse capítulo descreve o esquema de regras de produção (ênfatizando aplicabilidade, vantagens e desvantagens) e a linguagem Prolog. Tal esquema constitui um exemplo de *sistema de representação procedimental*, baseado na lógica de primeira ordem. Prolog constitui um exemplo dessa abordagem.

#### 3.1 Regras de Produção

Especialistas tendem a expressar suas técnicas de solução de problemas em termos de conjuntos de regras *situação-ação*, onde o conhecimento é representado como uma coleção de regras do tipo *se condição então ação*, denominada **regra de produção**. A *condição* estabelece o *contexto* para aplicação da regra. A *ação* corresponde a algum procedimento que acarreta uma conclusão ou mudança no estado corrente. O conjunto de regras de produção é visto como uma *representação de conhecimento procedimental*, baseada em lógica de primeira ordem. Esse esquema constitui um dos melhores meios disponíveis para codificação da experiência de especialistas, na resolução de problemas.

Sistemas de produção foram propostos por Newell e Simon, em 1972, no livro *Human Problem Solving*, como um modelo de cognição humana, voltado para a simulação do comportamento cognitivo de especialistas humanos, talvez devido ao carácter estímulo-resposta de uma regra de produção.

Segundo Hayes-Roth [HAY 85], apresentam as seguintes propriedades:

- a) incorporam conhecimento prático (heurístico) em regras *se-então*;
- b) sua habilidade cresce em uma taxa proporcional ao crescimento da base de conhecimento (crescimento incremental);
- c) podem resolver um grande intervalo de problemas possivelmente complexos utilizando regras relativamente simples e combinando os resultados de maneira apropriada;
- d) podem explicar suas conclusões refazendo suas linhas de raciocínio e traduzindo a lógica de cada regra empregada em linguagem natural.

Cada regra aproxima um *fragmento independente do conhecimento*. O conhecimento existente pode ser refinado, com a adição de nova regra, permitindo um crescimento incremental da base de conhecimento e aumentando a performance do sistema.

Regras de produção descrevem as relações, entre os objetos do domínio, de acordo com os valores que seus atributos podem ter. Seu formato típico é:

*se  $c_1$  e  $c_2$  e ... e  $c_n$  então  $a_1$  e  $a_2$  e ... e  $a_m$ .*

Cada *condição* ( $c_i$ ) da regra representa uma informação na forma *objeto-atributo-valor* ou *objeto-atributo*. Devemos identificar os objetos e as conseqüências que as alterações dos valores dos seus atributos possam ter sobre cada *ação* ( $a_j$ ). Abel [ABE 94] apresenta o seguinte exemplo, baseado em uma regra do MYCIN, programa de diagnóstico de doenças infecciosas (Tabela 3.1):

Se local da cultura for o sangue e a morfologia do organismo for bastonete e a mancha de gram do organismo for gram-positiva e o doente estiver em perigo *então* existe evidência de que a identidade do organismo é *Pseudomonas-aeruginosa* (Regra 102).

**Tabela 3.1** - Objeto-Atributo-Valor

Objeto	Atributo	Valor
cultura	local	sangue
organismo	morfologia	bastonete
organismo	mancha de gram	gram-positiva
doente	comprometimento	verdade
organismo	identidade	<i>Pseudomonas-aeruginosa</i>

Fonte: Mara Abel, 1994. Introdução aos Sistemas Especialistas.

Um sistema típico de regras consiste de *base de conhecimentos*, *memória de trabalho* e *máquina de inferências*. Algumas linguagens de programação como Prolog já incorporam a memória de trabalho e a máquina de inferências, sendo adequadas para a codificação de regras. Para se codificar sistemas de regras em linguagens tais como a C é necessário implementar tais mecanismos. Também estão disponíveis algumas “shells”, i.e programas específicos para a implementação de regras, os quais já incorporam a memória de trabalho, a máquina de inferências e rotinas de interface com o usuário, cabendo ao desenvolvedor do sistema, apenas a codificação do conhecimento.

A base de conhecimentos é composta pelo conjunto de regras e fatos. *Regras* formalizam um conhecimento e descrevem como é o processo de inferência sobre ele. Constituem declarações sobre classes de objetos. *Fatos* são declarações sobre objetos específicos. A lógica de processamento está distribuída pelas regras, as quais podem possuir nomes apenas para efeito de documentação, não sendo permitido que referenciem umas às outras. A comunicação entre elas é feita através dos dados da memória de trabalho.

O lado esquerdo da regra (*condição*), denominado *antecedente*, é uma combinação lógica de predicados, normalmente com uso dos conectivos *e* e *não*, onde os predicados restringem os valores possíveis de atributos de objetos que possam estar na memória de trabalho. O lado direito da regra (*ação*), denominado *consequente*, geralmente implica em modificações sobre os objetos na memória de trabalho ou eventuais efeitos colaterais (chamadas a rotinas externas de entrada e saída, etc.).

A *memória de trabalho* (ou *lista de contexto*) é uma base de dados que representa, em um dado momento, o estado do problema que se quer resolver. Ela possui dados dinâmicos de curta duração (que existem enquanto uma regra está sendo interpretada), em geral na forma objeto-atributo-valor que são utilizados para conduzir a execução das regras. Esses dados podem ser simples seqüências de caracteres ou até mesmo estruturas complexas, podendo ser implementados como listas, tuplas, registros, etc.

Com a especificação do problema a resolver (*meta*), a *máquina de inferências* é ativada. Ela é a responsável pela execução das regras, determinação de quais são relevantes, dada uma configuração da *memória de trabalho* e pela escolha de quais aplicar. Esse trabalho é em ciclos. Em cada um as regras são examinadas buscando as que são apropriadas para ativação. O ciclo pode ser dividido nas fases: *seleção das regras*, *resolução*

de conflitos e ação. Se o desenvolvedor estiver utilizando Prolog, Lisp ou mesmo uma “shell”, terá pouca ou nenhuma liberdade para especificar qual técnica utilizar em cada uma dessas fases. Por outro lado, se estiver desenvolvendo a aplicação em C terá que projetar e implementar técnicas específicas para cada fase.

A *seleção das regras* (ou “matching”) consiste em encontrar todas as regras que são satisfeitas pelo conteúdo da memória de trabalho, segundo critérios estabelecidos por algoritmos específicos. Consiste na tarefa de “casar” as regras com os dados da memória de trabalho, formando um “conjunto de conflito” constituído pelas regras que satisfazem as “condições de casamento”, i.é passíveis de serem executadas. As estratégias mais utilizadas são *raciocínio para frente* (raciocínio dirigido por regras) e *raciocínio para trás* (raciocínio orientado por metas). Também podem ser utilizadas estratégias mistas.

No *raciocínio para frente* (ou dirigido por dados, antecedente, progressivo, “forward-chaining” ou “bottom-up”) [HAY85], os dados da memória de trabalho são “casados” com as *condições* das regras. O programa começa com o estado inicial e gera estados intermediários (através da execução das *ações* das regras) que podem ser alcançados a partir do estado inicial. A *meta* permanece a mesma durante todo o processo de solução do problema. Quando um desses estados alcançáveis casar com a *meta*, o problema está solucionado.

Em outras palavras, crie uma árvore (de seqüências de movimentos candidatos a solução) com o nó raiz contendo o estado inicial (dados da memória de trabalho). O nível seguinte é composto por nós contendo estados formados pelas *ações* das regras cujas *condições* casam com o nó raiz. O próximo nível de cada nó é formado pelas *ações* das regras cujas *condições* casam com o estado do nó. Continue até ter gerado um nó com um estado que case com a *meta*.

No *raciocínio para trás* (dirigido por metas, conseqüente, regressivo, “top-down”, ou “backward-chaining”) [HAY 85], o programa inicia com a configuração objetivo final (*meta*) e seleciona uma ou mais regras, cuja *ação* casa com o objetivo em alguma extensão. O lado esquerdo das regras selecionadas é utilizado para gerar novos objetivos a serem atingidos e o processo continua até que um deles seja casado com o estado inicial.

Em termos da árvore (de seqüências de movimentos candidatos a solução) inicie com raiz contendo a *meta*. O nível seguinte é composto por nós contendo estados formados pelas *condições* das regras cujas *ações* casam com o nó raiz. O próximo nível de cada nó é formado pelas *condições* das regras cujas *ações* casam com o estado do nó. Continue até ter gerado um nó com um estado que case com o estado inicial.

Considere o problema de responder a questão ‘existe x que é pessoa?’ com base na base de conhecimentos:

fatos

Marcos é homem.

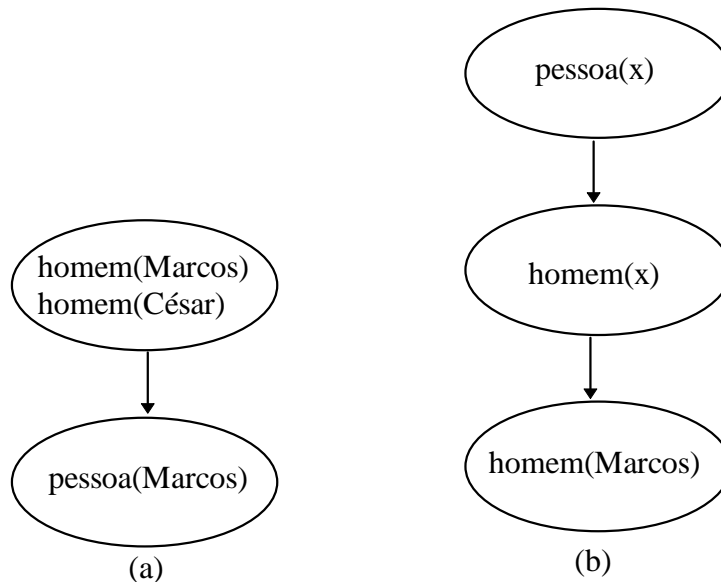
César é homem.

regra 1

se x é homem então x é pessoa.

Com a estratégia *raciocínio para frente* o nó raiz conterá o estado inicial (conjunto dos fatos) e a *meta* será encontrar uma pessoa. Como o *antecedente* da regra 1 casa com o estado inicial, o segundo nível será composto do nó ‘Marcos é pessoa’, sendo o processo de inferência encerrado porque houve o casamento com a meta, através da substituição da variável x por Marcos (Figura 3.1a). O sistema responde com ‘sim, x=Marcos’.

Com a estratégia *raciocínio para trás* o nó raiz conterá a *meta*. O próximo nível é formado pelo *antecedente* da regra 1, i.é ‘x é homem’ pois o seu *consequente* casa com a raiz. Analogamente o segundo nível é formado por ‘Marcos é homem’ e a inferência é interrompida pois se atingiu um estado inicial (Figura 3.1b). O sistema responde com ‘sim, x=Marcos’.



**Figura 3.1** - Exemplo de Raciocínios (a) para Frente e (b) para Trás

Raciocínio para trás é utilizado por provadores de teoremas e é adequado para solução de problemas *dirigidos por objetivos*. Raciocínio para a frente é mais adequado quando o problema é *dirigido por dados* informados durante a resolução do problema. Em geral essa estratégia é mais complexa do que o raciocínio para trás, mas existem problemas específicos para os quais ela é mais adequada. Por exemplo, a integração simbólica é mais facilmente resolvida resolvendo-se, passo a passo, as integrais mais simples, usando os conceitos de integração para tentar gerar uma expressão sem integrais. Também existem sistemas de regras que utilizam uma forma de raciocínio combinado. Assim um *sistema de diagnóstico médico* normalmente inicia com o informe dos dados sobre o estado do paciente (i.é os achados ou “findings”) e raciocínio para frente. Se, em um determinado ponto, o lado esquerdo de uma regra está quase satisfeito, é natural aplicar o raciocínio para trás, na tentativa de inferir o valor da condição não informada ou mesmo consultar o usuário sobre o valor, para se poder aplicar a regra em questão.

Uma seleção inteligente das regras deve escolher as que podem ser aplicadas em um determinado ponto e que tenham chance de levar a uma solução, *evitando-se* realizar uma busca exaustiva entre todas elas, comparar as condições de cada uma delas com o estado atual e extrair todas as que casam. Para tanto podem ser aplicadas as seguintes técnicas:

- a) indexar as regras e utilizar o estado atual como índice de acesso, através do uso de uma função adequada de “hashing”;

- b) tirar partido da *natureza temporal* dos dados<sup>4</sup>, *similaridade estrutural* nas regras<sup>5</sup> e persistência da *consistência das ligações* das variáveis<sup>6</sup> em sistemas com *casamento com variáveis* (i.é muitos-para-muitos), onde as condições são definidas em termos de descrições das propriedades que as situações possam ter, nos quais *muitas* regras são casadas simultaneamente a *muitos* elementos da descrição do estado atual;
- c) usar um grupo de regras para descrever como certas propriedades podem ser inferidas a partir de outras, em sistemas de casamento complexo (onde as condições de uma regra especificam a exigência de propriedades que não estão definidas explicitamente na descrição do estado atual);
- d) usar *casamento aproximado* em sistemas onde as regras podem ser aplicadas se suas condições casem aproximadamente com a situação atual, tais como em *processamento de fala* ou *reconhecimento de padrões com erros*.

O leitor interessado em maiores detalhes sobre casamento muitos-para-muitos poderá consultar [FOR 82], onde é apresentada a descrição do algoritmo RETE. O RETE grava informações do estado do problema, entre ciclos, armazenando cálculos anteriores que podem ser utilizados nos ciclos futuros, tirando partido da natureza temporal dos dados, similaridade estrutural em regras e persistência da consistência das ligações das variáveis.

A *resolução de conflitos* da fase do ciclo da máquina de inferências consiste em resolver o conflito ou seja selecionar qual a ordem em que as regras do conjunto de conflito serão aplicadas. A estratégia a ser utilizada deve possuir as seguintes características desejáveis: a) sensibilidade às modificações realizadas sobre os dados da memória de trabalho e b) estabilidade (certo grau de continuidade em relação à linha de raciocínio).

As três abordagens básicas são atribuir uma preferência com base: a) na regra que casou, b) nos objetos que casaram e c) na ação que seria executada pela regra que casou.

A *preferência baseada em regras* consiste em atribuir prioridade às regras na ordem em que aparecem. Também pode ser dada prioridade a regras especiais (mais diretas) do que a regras mais genéricas. Uma regra é mais genérica (menos específica) se o seu conjunto de condições contém *menos* condições do que uma outra ou se as suas condições são iguais às de outra regra, exceto pelo fato de que no primeiro caso são especificadas *variáveis*, enquanto no segundo caso são especificadas constantes.

A *preferência baseada em objetos* consiste em priorizar as regras encontradas, com base na importância dos objetos que foram casados.

A *preferência baseada em ação* consiste em disparar, temporariamente, todas as regras selecionadas, utilizar uma função heurística para avaliar os resultados de cada uma delas e priorizá-las segundo o seu mérito.

---

<sup>4</sup> regras geralmente não alteram de forma radical a descrição de estado, logo se ela não casou no ciclo anterior, é provável que não conseguirá ser aplicada ao ciclo atual.

<sup>5</sup> condições compartilhadas por diferentes regras são casados, no máximo, uma vez por ciclo.

<sup>6</sup> análise da consistência das condições de uma regra, antes da avaliação do casamento propriamente dito, pois embora as condições individuais de uma regra possam ser satisfeitas, pode haver conflitos nas ligações das variáveis, em relação ao estado atual, que a impeçam de ser disparada.



A *ação*, terceira fase do ciclo da máquina de inferência, consiste na execução da(s) regra(s) selecionada(s), normalmente provocando alterações na memória de trabalho que podem ativar outras regras no ciclo seguinte ou ordenar o fim do ciclo.

### 3.2 Linguagem Prolog

Prolog é uma linguagem que utiliza o paradigma de programação em lógica, no qual as definições lógicas são vistas como programas. Definições lógicas, em programas Prolog, constituem *cláusulas de Horn*. Uma *cláusula de Horn* é uma *cláusula* que tem no máximo um literal positivo. Qualquer fbf de uma lógica de primeira ordem pode ser representada na forma clausal. Uma *cláusula* é uma fbf em *forma conjuntiva normal*, mas sem nenhuma instância do conectivo *e* ( $\wedge$ ) ou seja é uma fórmula com um prefixo de quantificadores universais ( $\forall$ ) aplicado a predicados conectados por *ou* ( $\vee$ ).

Casanova [CAS 87] e Rich [RIC 93] apresentam uma demonstração e um algoritmo para converter uma fórmula qualquer em sua notação clausal (representada através de uma disjunção finita de fórmulas atômicas). Após a aplicação do algoritmo de conversão a um conjunto de fórmulas, é gerado um conjunto equivalente de cláusulas, constituindo uma grande conjunção, pois todas elas devem ser simultaneamente verdadeiras.

A Figura 3.2 apresenta uma pequena base de conhecimento representada em notação lógica padrão, forma clausal e em Prolog, adaptada de [RIC 93]. Os quantificadores não são explicitados em Prolog e o conectivo  $\wedge$  é representado por vírgula. As implicações do tipo  $p \rightarrow q$  são representadas invertidas como  $q:-p$ . As disjunções  $\vee$  são representadas como uma lista de declarações alternativas. Em algumas versões de Prolog, elas podem ser representadas por um ponto-e-vírgula. Todos os comandos Prolog são finalizados por um ponto. As *cabeças* das regras Prolog (*consequentes* das regras de produção) só podem ter um único predicado representado uma única ação. Se a fórmula contiver *variáveis*, elas são codificadas através de identificadores que começam com letras maiúsculas, enquanto as *constantes* começam com letras minúsculas ou números.

<u>Notação Lógica</u>	<u>Cláusulas de Horn</u>	<u>Notação Prolog</u>
$\forall x(\text{estimação}(x) \wedge \text{pequeno}(x) \rightarrow \text{bichoapt}(x))$	$\neg \text{estimação}(x) \vee \neg \text{pequeno}(x) \vee \text{bichoapt}(x)$	<code>bichoapt(X):-estimação(X), pequeno(X).</code>
$\forall x(\text{gato}(x) \vee \text{cachorro}(x) \rightarrow \text{estimação}(x))$	$\neg \text{gato}(x) \vee \text{estimação}(x) \vee \neg \text{cachorro}(x) \vee \text{estimação}(x)$	<code>estimação(X):-gato(X). estimação(X):-cachorro(X).</code>
$\forall x(\text{poodle}(x) \rightarrow \text{cachorro}(x) \wedge \text{pequeno}(x))$	$\neg \text{poodle}(x) \vee \text{cachorro}(x) \vee \neg \text{poodle}(x) \vee \text{pequeno}(x)$	<code>cachorro(X):-poodle(X). pequeno(X):-poodle(X).</code>
<code>poodle(fido)</code>	<code>poodle(fido)</code>	<code>poodle(fido).</code>

**Figura 3.2** - Exemplo de Notações Lógica, Cláusulas de Horn e Prolog

De fato as cláusulas Prolog representam cláusulas de Horn transformadas. Se uma cláusula de Horn não possui nenhum literal negativo, não é alterada. Caso contrário a cláusula deve ser rescrita na forma de uma implicação com todos os literais negativos formando o *antecedente* da implicação e o único literal positivo formando o seu *consequente*. Essa implicação é representada como uma cláusula Prolog, cuja *cabeça* é o *consequente* e cujo *corpo* é formado pelo *antecedente*. As variáveis que aparecem na *cabeça* da cláusula Prolog são consideradas universalmente quantificadas e aquelas que aparecem somente no *corpo* são consideradas existencialmente quantificadas.

Com a representação uniforme decorrente das cláusulas de Horn, pode ser construída uma máquina de inferência simples e eficiente, que tira partido do fato de que a lógica de cláusulas de Horn é *decidível*. Assim há garantias de parar quando a cláusula que tiver de ser demonstrada não for dedutível a partir dos fatos e regras conhecidos. Todas as cláusulas da base de conhecimento são codificadas como cláusulas de Horn transformadas, i.é cláusulas Prolog.

Para que a base de conhecimentos de um sistema de regras seja mantida em tamanho gerenciável, por razões de eficiência e praticidade, é praxe representar apenas o conhecimento positivo (asserções afirmativas) sobre um certo domínio, pois o número de asserções negativas é muito maior do que o de asserções afirmativas. Admite-se a *hipótese do mundo fechado* que define que todas as declarações relevantes e verdadeiras estejam contidas na base de conhecimentos ou que possam ser derivadas das declarações lá contidas. Qualquer declaração que não esteja presente ou que não possa ser deduzida é admitida falsa. Prolog adota essa filosofia e, como consequência, a *negação lógica* não pode ser representada explicitamente. Ela é representada implicitamente pela ausência de uma declaração. Assim para determinar se uma declaração é verdadeira (*consistente*) ou não, Prolog usa uma aproximação. A aproximação é a *noção de negação por falha*. Para mostrar que  $P$  é *consistente*, tenta-se provar  $\neg P$ . Se falhar, assume-se que  $\neg P$  é falsa e que  $P$  é *consistente*.

A máquina de inferência da Prolog é composta do procedimento de inferência e de uma estratégia de busca. O primeiro especifica o que fazer (i.é quais são as derivações admissíveis) e o segundo especifica a sequência na qual as derivações no espaço de busca são geradas, na tentativa de refutar o objetivo. A máquina de inferência utiliza a regra de inferência *resolução* e *raciocínio para trás* com estratégia de *busca em profundidade*, analisando as regras da *esquerda para a direita* e de *cima para baixo*.

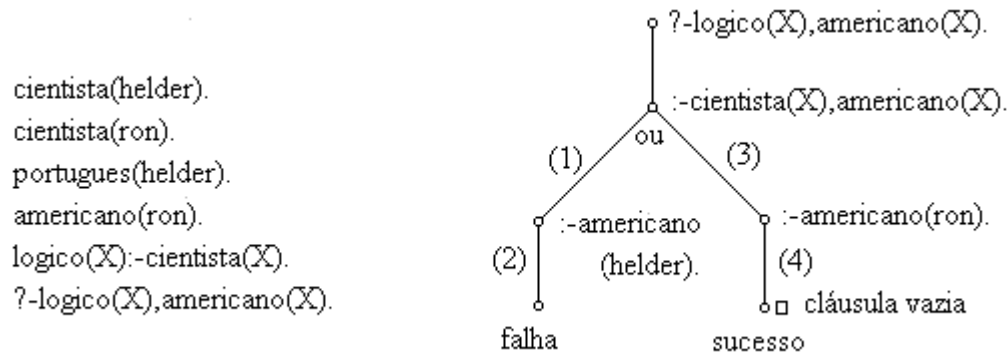
Em resumo, Prolog utiliza a) regra de inferência: *resolução*; b) seleção das regras: *raciocínio para trás* com indexação das regras pelo predicado e processo de casamento precedido pelas instanciações apropriadas das variáveis das regras (*unificação*); e c) resolução de conflitos: preferência baseada em regras (ordem das regras).

A forma típica de *resolução* é o método chamado de *resolução por refutação*. Como vimos (Capítulo 2), basicamente negamos a proposição a provar e a adicionamos ao conjunto de fórmulas, criando um novo conjunto. Então tentamos provar que essa inclusão leva a uma contradição.

Coelho e Cotta [COE 88] apresenta o seguinte fluxo de controle para execução de um programa Prolog e o exemplo da Figura 3.3:

- 1) para processar um objetivo, "case-o" com a *cabeça* de uma cláusula e processe os objetivos no *corpo* daquela cláusula;
- 2) se nenhum casamento pode ser feito, caminhe para trás (faça um "backtrack"), i.é retorne para o objetivo mais recentemente processado e procure por um casamento alternativo; caso o casamento aconteça, a cabeça da cláusula estará unificada;
- 3) processe objetivos da esquerda para a direita; e
- 4) avalie as cláusulas na ordem em que aparecem (i.é de cima para baixo).

Na Figura 3.3, o grafo é uma prova Prolog e exemplifica a resolução por refutação. O sinal ‘?-’ indica um objetivo a ser provado ou refutado. Recorde que esse método implica tentar provar a negação do objetivo; neste caso ‘ninguém é lógico e americano’. O primeiro nível dessa árvore foi gerado pelo processamento do corpo da regra ‘logico(X):-cientista(X).’, cuja cabeça casa com parte do objetivo. O segundo nível foi gerado pelo processamento do fato ‘cientista(helder)’ implicando a instanciação da variável X com a constante ‘helder’, resultando no objetivo ‘:-americano(helder)’. Esse objetivo não pode ser provado, resultando em falha. Nesse ponto o sistema retorna para o objetivo imediatamente anterior (‘:-cientista(X),americano(X)’) e procura por um casamento alternativo, encontrando ‘cientista(ron)’ e gerando a sub-árvore direita, com novo objetivo ‘americano(ron)’. O sistema pesquisa novamente a base de conhecimentos encontrando o fato ‘americano(ron)’, podendo provar o objetivo, o que é indicado pela ocorrência da cláusula vazia (representada por um pequeno quadrado) e a indicação de sucesso. Os números entre parênteses indicam a ordem de geração dos ramos da árvore.



**Figura 3.3** - Exemplo de Execução de um Programa Prolog

### 3.3 Vantagens e Desvantagens

O esquema baseado em regras de produção possui ampla expressividade e o seu uso apresenta as seguintes vantagens:

- naturalidade: sintaxe e semântica muito simplificadas, torna bastante fácil entender e aplicá-lo em sistemas de diagnóstico. É um dos melhores meios disponíveis para codificação da experiência de especialistas na resolução de problemas;
- modularidade: uma regra pode ser acrescentada, eliminada ou mudada de forma independente das outras, pois só se comunicam através da memória de trabalho, embora isso possa afetar a performance do sistema. Em algumas aplicações as regras podem ser estruturadas de forma a se obter subconjuntos de regras independentes e complementares o que facilita o processo de resolução de conflitos.
- facilidades para prototipação devido à possibilidade de crescimento incremental da base de conhecimentos;
- uniformidade: todas as informações podem ser codificadas segundo a estrutura de regras, tornando-as mais fáceis de serem entendidas por outras pessoas.

Suas principais desvantagem de uso são:

- falta de estruturação da base de conhecimentos, que faz com seja difícil introduzir modificações, bem como localizar informações desejadas, em sistemas com número expressivo de regras e em domínios complexos. O conhecimento resulta expresso como um conjunto desordenado e não-estruturado de regras, dificultan-

do bastante a representação das estruturas inerentes ao domínio da aplicação, tais como taxonomias de classes, relações temporais, relações estruturais, herança de atributos, etc.;

- b) ineficiência de execução. A alta modularidade e uniformidade das regras impõe que cada inferência executada, o seja através do ciclo seleção da regra-ação e consolidação das informações através da memória de trabalho. Isso dificulta responder a determinadas seqüências de situações, quando for o caso;
- c) opacidade. É difícil seguir o fluxo de controle durante a resolução do problema, pois os algoritmos de resolução não estão aparentes. A falta de comunicação explícita entre as regras e a não existência de mecanismos tais como chamadas de funções ou subrotinas, contribui para esse fato. Regras de produção são adequadas para a representação de conhecimento do tipo condição-ação e não para representar conhecimento procedimental;
- d) não facilita distinções semânticas entre propriedades essenciais e propriedades complementares dos objetos.

Normalmente o sistema de regra de produção é associado com o raciocínio monotônico, mas podemos ter sistemas de regras que realizam raciocínio “default” ou estatístico, como veremos na apresentação de esquemas voltados para o tratamento de conhecimento incerto (lógica não-monotônica, lógica “default” e regras com fatores de certeza associados).

Regras de produção são adequadas para representar associações empíricas, principalmente em domínio não estruturado, onde não é possível expressar conhecimento de forma exata. Normalmente representa conhecimento sobre como as pessoas fazem uma tarefa específica, tal como diagnóstico médico, exploração de minerais, etc.

Davis e King [Davis and King (1977) Apud BAR 86, p.195] apresentam a seguinte caracterização de domínios adequados para utilização de regras de produção:

- a) domínios nos quais o conhecimento é difuso, consistindo de muitos fatos (p.ex. medicina clínica), em oposição a domínios nos quais existe uma concisão, uma teoria unificada (física);
- b) domínios nos quais processos podem ser representados como um conjunto de ações independentes (p.ex. sistema médico de monitoramento de pacientes), em oposição a domínios com sub-processos dependentes (p.ex. folha de pagamento);
- c) domínios nos quais conhecimento pode ser facilmente separado da maneira na qual será usado (p.ex. uma classificação de taxionomias como as usadas em biologia), em oposição a casos nos quais representação e controle estão misturados (p.ex. uma receita ou uma prescrição).

O sistema MYCIN, que abordaremos na discussão do uso de regras com fatores de certeza, é um exemplo de sistema especialista de sucesso que utiliza regras de produção com raciocínio para trás.

## 4 Redes Semânticas e Sistemas de Frames

Para Rich e Knight [RIC 93, p.297] não existe uma distinção clara entre rede semântica e um sistema de *frames*. Quanto maior estrutura tiver o sistema, mais possibilidades terá de ser chamado de sistema de *frames*. Assim apresentaremos conceitos básicos sobre redes semânticas, considerando-as como um caso particular de sistemas de *frames*.

### 4.1 Redes Semânticas

Redes semânticas constituem uma classe de formalismos de representação de conhecimento que usam nós para representar objetos, conceitos ou situações, ligados por arcos orientados (os quais representam relacionamentos entre eles) [BAR 86]. A idéia subjacente é que o significado de um conceito vem do modo como ele é associado a outros conceitos. Nós e arcos podem ser rotulados, formando um grafo rotulado direcionado.

As redes semânticas foram propostas por Quillian (1968), no artigo *Semantic Memory* [Quillian (1968) Apud BAR 86], como um modelo psicológico explícito da memória associativa humana. Esse modelo explica alguns resultados experimentais sobre o comportamento da memória humana, dentre eles o fato de que o reconhecimento de objetos pertencentes a classes com mais elementos toma mais tempo do que o reconhecimento dos pertencentes a classes menos numerosas [BIT 96].

Os atributos ou características de um objeto ou classe podem conter várias informações para melhor representá-los, podendo ser informações estáticas (valores definidos) ou procedimentos de resolução que permitem derivar os dados representados.

Nós representam objetos, conceitos ou situações; em especial indivíduos, conjuntos, predicados, etc. Ao nó que representa uma classe de objetos, denominamos *nó genérico* e *nó individual* ao que representa uma instância (indivíduo).

Existem arcos especiais (denominados *é-um*) que transmitem a noção de *herança de atributos* de uma classe de elementos para subclasses ou quaisquer instâncias da classe ou subclasse. Assim podemos definir propriedades de objetos individuais a partir de propriedades definidas para a classe a qual pertence o objeto.

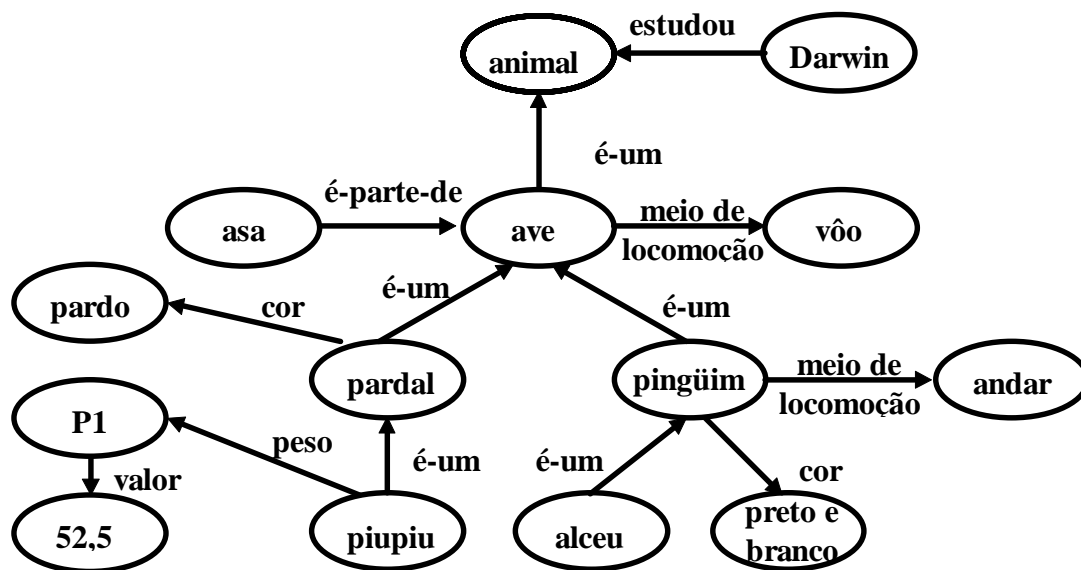
Através da herança de propriedades, fatos podem ser inferidos sem terem sido explicitamente representados, mas essas inferências possam não ser válidas em situações especiais. Para inibir esse mecanismo é necessário associar o novo atributo ao nó que *quebra* a herança. A relação que faz a ligação entre o nó e o seu atributo deve ser a mesma (i.é ter o mesmo nome).

A Figura 4.1 mostra um caso clássico de *quebra* de herança em uma rede semântica (através do atributo *meio-de-locomção*) e exemplifica as relações mais utilizadas.

O arco *é-um* é utilizado para representar taxonomia de classes, subclasses e instâncias. Alguns autores utilizam o arco *é-um* para representar relação de *generalização* ou *especialização* entre nós genéricos e os arcos *instância-de* ou *membro-de* para representar a *relação de classificação* entre um nó individual e o nó genérico correspondente.

O arco *é-parte-de* representa a *relação de agregação* entre um objeto e suas partes e define um componente de uma instância ou classe. No exemplo, temos “todas as aves têm asas”. Já o arco *valor* não define nova entidade, mas uma relação entre entidades existentes. O uso do nó P1 e do arco *valor* ao invés de ligar o nó 52,5 gramas ao arco *peso* permite representar fatos como “o peso de Piupiu aumentou” porque o valor da propriedade peso pode aumentar mas o número 52,5 não. Os outros arcos definem relações específicas para aquela rede.

Podem ser especificados valores “*default*” para atributos de forma que se um atributo não tiver um valor especificado, assume o valor “*default*”.



**Figura 4.1** - Exemplo de uma Rede Semântica

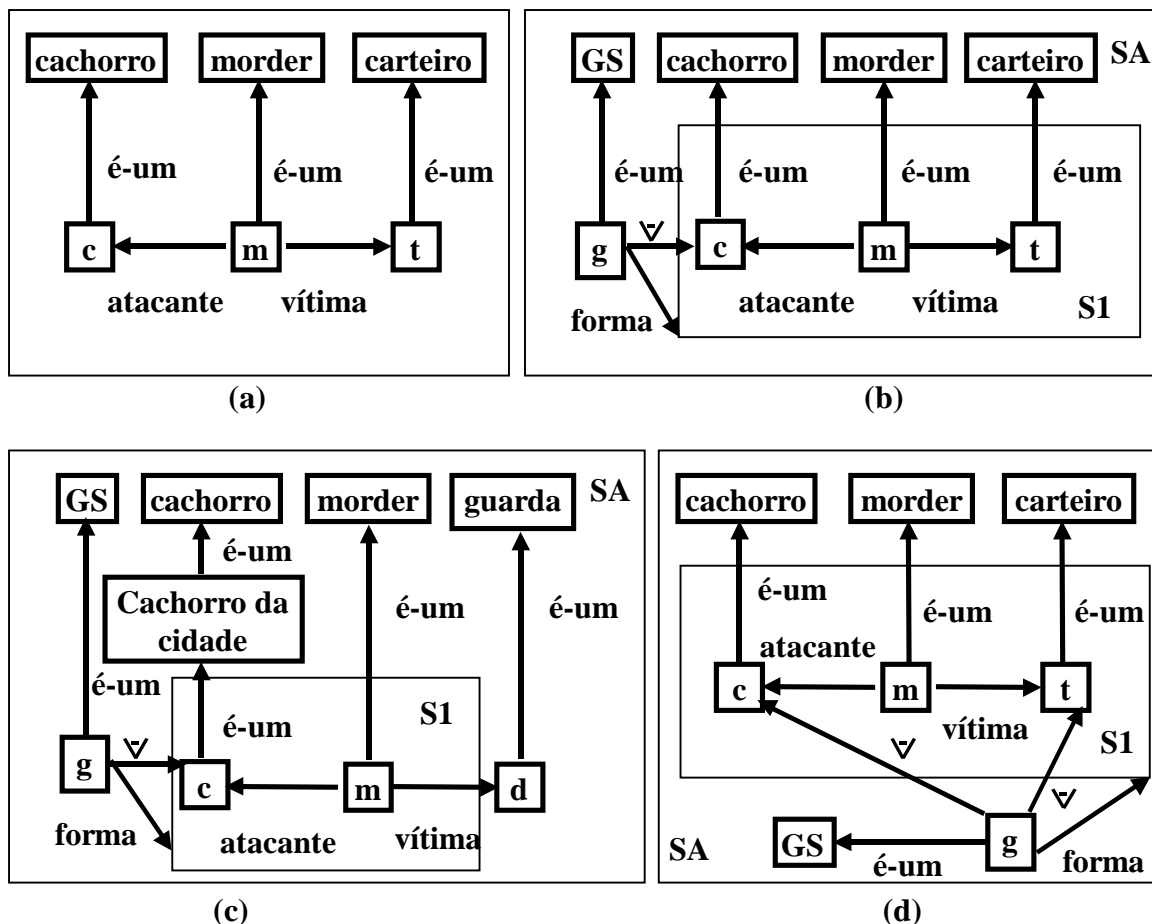
O fato que Piupiu voa não está explicitamente representando na rede semântica, mas é inferido devido ao mecanismo de herança da subclasse pardal, do meio de locomoção da classe ave. De forma geral, se um atributo não está definido para um objeto, ele pode ser herdado da classe mais próxima, acessível a partir do objeto, através de arcos *é-um*. Portanto o objeto herda valores de atributos de todas as classes das quais é membro, sendo que um valor de uma classe mais restrita tem precedência sobre um valor de uma classe mais ampla. Há uma quebra de herança para a subclasse pingüim, pois o atributo *meio-de-locomocão* foi redefinido como *andar*.

Está representado que Darwin estudou animais e pelo mecanismo de herança podemos concluir que Piupiu ou Alceu foram estudados por Darwin, o que não é necessariamente verdadeiro. Tal fato é derivado da não distinção entre um indivíduo e uma classe de indivíduos, nesse formalismo. Assim todas as asserções sobre uma classe são entendidas serem válidas para todos os membros da classe.

Existem implementações de redes semânticas que permitem, através de um recurso computacional denominado “*demon*”, especificar um procedimento de cálculo para o valor de um atributo ao invés de um valor específico ou “*default*”. O valor do atributo depende da informação armazenada na estrutura e o procedimento de cálculo é disparado ao ser feita uma referência ao atributo em questão. Esse procedimento pode utilizar, como entrada, todos os valores dos atributos do nó.

A representação de predicados binários da lógica de predicados é direta em redes semânticas, visto que um arco é basicamente uma relação binária. Predicados de aridade maior podem ser representados através da criação de um objeto que represente toda a declaração do predicado e da introdução de arcos para descrever o relacionamento de cada um dos argumentos originais com este novo objeto.

É possível representar expressões que usam os quantificadores universal e existencial da lógica de predicados, particionando a rede semântica em um conjunto de espaços, cada um correspondendo ao escopo de uma ou mais variáveis, conforme proposto por Hendrix[Apud RIC 93]<sup>7</sup>. A Figura 4.2, extraída de [RIC 93], exemplifica as seguintes sentenças: a) o cachorro mordeu o carteiro, b) todo cachorro já mordeu um carteiro, c) todo cachorro da cidade já mordeu o guarda, e d) todo cachorro já mordeu todo carteiro. O nó  $g$  é uma instância da classe especial GS das declarações universais (i.é com quantificadores universais). Todo elemento de GS tem dois atributos ou mais: forma (expressa a relação sendo definida) e uma ou mais conexões (arcos), uma para cada uma das variáveis universalmente quantificadas. *Forma* determina um contexto para os quantificadores; variável no escopo de forma, não diretamente ligada por arco ao nó  $g$ , é considerada existencialmente quantificada. Não existem variáveis quantificadas em (a);  $c$  é universalmente enquanto  $m$  e  $t$  são existencialmente quantificadas em (b);  $c$  é universalmente e  $m$  é existencialmente quantificada em (c);  $c$  e  $t$  são universalmente e  $m$  é existencialmente quantificada em (d).



**Figura 4.2 - Rede Semântica Particionada**

<sup>7</sup> Expanding the Utility of Semantics Networks through Partitioning. **Proceedings of the IJCAI, 1977.**

O raciocínio baseado apenas em percorrer as ligações de uma rede semântica pode envolver inferências que não são necessariamente válidas, como vimos através do exemplo “Piupiu ou Alceu foram estudados por Darwin”. O significado de uma rede semântica é estabelecido também pelos procedimentos que a interpretam e não somente pela rede em si, mas não há uma convenção sobre a semântica das suas estruturas. Assim as inferências obtidas pela manipulação da rede não são garantidamente válidas, assim como elas o são no esquema baseado em lógica [BAR 86, p.157].

Outra forma comum de raciocínio em redes semânticas é a inferência por “*matching*”, análoga ao processo de unificação, no esquema de regras de produção, que é baseado na construção de uma subrede que possui nós com valores definidos e nós com valores desejados, mas desconhecidos, representados por variáveis. O procedimento de “*matching*” procura na rede por uma subrede que “*case*” com a subrede construída e, quando ela é encontrada, as variáveis são instanciadas com os valores correspondentes da subrede encontrada.

Redes semânticas constituem um exemplo de esquema de representação de conhecimento que utiliza o sistema *de representação declarativo*. Esse formalismo tem sido utilizado diversas áreas tais como modelo psicológico da memória humana [Quillian, 1968; Anderson and Bower, 1973. Programa HAM; e Norman and Rumelhart, 1975. Apud BAR 86], processamento de linguagem natural [Simmons and Slocum, 1972; Simmons, 1973. Apud BAR 86], tutores [Carbonell and Collins, 1974. Programa SCHOLAR. Apud BAR 86], processamento de voz [Woods et al., 1976. Apud BAR 86], etc.

Em 1975 Minsky, no artigo “A framework for representing knowledge” [Minsky, 1975. Apud BAR 86] introduziu nós com estruturação interna, criando o sistema de *frames* e Woods, no artigo “What’s in a link: Foundations for semantic networks” [Woods, 1975. Apud BAR 86] atentou para a necessidade de uma semântica formal para redes semânticas. Em resposta a Woods diversos autores passaram a associar redes semânticas ao formalismo lógico.

## 4.2 Sistemas de *Frames*

Segundo Rich [RIC 93] e Barr [BAR 86], um sistema de *frames* (também denotado sistema de enquadramento) é uma coleção de locais onde armazenar conhecimento, em geral chamados de “*slots*” e valores a eles associados (e possivelmente restrições a estes valores) que descrevem alguma entidade do mundo. *Frames* foram propostos por Minsky em 1975 e definidos como uma estrutura de dados para a representação de *situações estereotipadas*. A idéia é baseada no entendimento de que a codificação de conceitos realizada pelo cérebro humano está mais relacionada com *propriedades marcantes*, associadas a objetos que são *típicos* de sua classe, do que com definições exatas e exaustivas das propriedades que uma determinada entidade deve possuir para ser considerada como pertencente a uma certa categoria. Eles podem descrever um objeto prototípico, representante de uma classe de objetos, conceito ou situação [FRI 85].

As pessoas possuem a habilidade de reagirem a uma nova situação, aplicando expectativas baseadas em experiências passadas. A menos que existam evidências em contrário, elas esperam que as suas expectativas sejam verdadeiras para a nova situação e mantêm



grandes coleções mentais de estruturas de conhecimento que incluem estas expectativas como “*default*” para as características correspondentes. *Frames* proporcionam uma estrutura na qual novos dados são interpretados em termos de conceitos adquiridos através de experiência prévia. A organização desse conhecimento facilita *raciocínio guiado por expectativa*, i.é procura por coisas que são esperadas, com base no contexto que se imagina estar [BAR 86].

De forma análoga às redes semânticas, sistemas de *frames* constituem uma classe de esquemas de representação de conhecimento, não existindo uma notação padrão.

Podemos criar sistemas de *frames* a partir de coleções de *frames* conectados entre si, atribuindo um outro *frame* ao valor de um “*slot*”. “*Slots*” podem armazenar valores, lista de valores, restrição sobre valores válidos, tipo de dado (inteiro, real, “*string*”, lógico), indicação de valor não especificado, unidades de medida, ponteiros para *frames*, valores “*default*”, “*demon*” (procedimentos através dos quais um valor é obtido ou validado), nome de um outro *frame* e tipos de relacionamento entre os diversos *frames*.

*Frames* é exemplo do sistema de *representação estrutural*, mas a inclusão de procedimentos (conhecimento dinâmico) nos “*slots*” permite o uso de *representação procedimental* do conhecimento. Os “*demons*” podem ser do tipo *se-necessário* (ou *se-buscado*) e *se-adicionado* (ou *se-definido*). Procedimentos *se-necessário* descrevem o processo utilizado para obter um valor para um determinado “*slot*”, quando o valor se faz necessário. Procedimentos *se-adicionado* são ativados pela atribuição ou adição de valores a um “*slot*”, sendo normalmente utilizados para verificação de restrições de integridade e consistência.

Minsky também propôs que um *frame* tivesse a habilidade de determinar se ele é aplicável ou não em uma dada situação. Se o *frame* considerado mais provável para o entendimento de uma situação corrente (p.ex. um diálogo, cena ou problema), ao tentar utilizar os dados disponíveis, descobrir que não corresponde à situação em questão, deverá transferir o controle para outro *frame*, mais adequado [BAR 86, p.159].

Tal comportamento pode ser obtido com o uso de procedimentos do tipo *se-adicionado* ou propriedades *trigger* e tem sido utilizado principalmente em sistemas para diagnóstico médico, baseados em *frames*, pois normalmente quando um diagnóstico é descartado, um outro similar e mais provável, é indicado.

A atribuição de valor estático a um “*slot*” permite o uso de uma *representação declarativa* do conhecimento: a asserção de que um fato (atributo e seu valor) é verdadeiro.

Os relacionamentos podem ser do tipo:

- é-um (isa)* - relação de *especialização/generalização*, relaciona superclasses com suas subclasses. É análoga a relação de subconjunto da teoria dos conjuntos. Permite definir uma hierarquia de *frames* formando uma taxonomia de classes.
- instância* - corresponde a relação *elemento-de* da teoria dos conjuntos.
- mutualmente-disjunta-de* - relaciona uma classe a uma ou mais classes que garantidamente não têm elementos em comum.
- é-coberto-por* - relaciona uma classe a um conjunto de subclasses, cuja união é igual a ela. Se o conjunto de subclasses é formado por subclasses mutuamente disjuntas, é denominado uma *partição* da classe coberta por ele.

Alguns autores utilizam as relações *um-tipo-de* (*a kind of* ou *ako*) ou *especialização-de* (*specialization-of*) para expressar especialização/generalização e *é-um* (*is a* ou *isa*) para relacionar objetos (instâncias) às suas classes.

Em geral os sistemas de *frames* implementam inversas para as relações *é-um* e *instância*.

Todos os *frames* em um sistema de *frames* devem estar relacionados entre si e ligados direta ou indiretamente a um *frame* inicial chamado *raiz*. A partir do *raiz*, os *frames* são criados e relacionados através do “*slot*” *é-um*.

Na base de conhecimentos, os *frames* são classificados como construtores ou instâncias.

Os *construtores* definem a estrutura que as informações devem ter para serem incluídas na base de conhecimentos, de forma análoga ao esquema conceitual de um banco de dados. Desta forma não correspondem a objetos do mundo real, mas a um estereótipo de como estes objetos são e como se relacionam com os demais. Esses *frames* admitem o uso de “*facets*” para o “*slot*” i.é. definir restrições simultâneas para os valores que um “*slot*” possa ter. As “*facets*” podem definir o tipo de dado, o intervalo de valores válidos, a lista de valores possíveis, o valor que deve ser assumido como “*default*”, “*demons*”, etc.

Os *frames instâncias* correspondem aos objetos do mundo real, representados na base de conhecimentos.

Não é possível a representação de qualquer conceito, através de *frames*, que não possa ser representado por meio da lógica de primeira ordem, mas a integração de toda a informação (sobre uma entidade do domínio) em um *frame* e os mecanismos de suporte disponíveis o tornam um formalismo poderoso [Ni80 Apud MAT 91].

Os relacionamentos *é-um* e *instância* transmitem a noção de *herança de propriedades* de uma classe de elementos para subclasses ou quaisquer instâncias da classe ou subclasse. Assim podemos definir propriedades de objetos individuais a partir de propriedades definidas para a classe a qual pertence o objeto. O uso de “*default*” e de valores de atributos herdados, permite raciocínio eficiente porque desobriga a necessidade de realizar inferências para redescobrir fatos velhos em novas situações, de maneira similar a que as pessoas utilizam o conhecimento de experiências passadas.

Rich e Knight [RIC 93] apresentam o algoritmo a seguir (Tabela 4.1), para herança de propriedades *monovaloradas* em um sistema de *frames*, o qual é admitido implementado como um grafo acíclico direcionado. O algoritmo consiste em, a partir do *frame* para o qual um valor de “*slot*” é necessário, efetuar uma busca (em profundidade ou amplitude), subindo pela hierarquia através das ligações *instância* ou *é-um*. Se um caminho produzir um valor, ele pode ser encerrado, e também todos os outros caminhos cujas *distâncias inferencial* excedam a distância inferencial do percurso bem-sucedido. Se encontrar duas respostas concorrentes, situadas a mesma distância inferencial, deve ser reportado uma contradição.

O conceito de *distância inferencial*, introduzido por Touretzky [TOU 84], em 1984, pode ser definido como: *a Classe<sub>1</sub> está mais perto da Classe<sub>2</sub> do que da Classe<sub>3</sub> se e*

apenas se a *Classe<sub>1</sub>* tem caminho de inferência através da *Classe<sub>2</sub>* para a *Classe<sub>3</sub>*, ou seja *Classe<sub>2</sub>* está entre a *Classe<sub>1</sub>* e a *Classe<sub>3</sub>*. Esse conceito define apenas uma organização parcial; algumas classes não são comparáveis entre si.

O algoritmo de herança de propriedades sempre pára porque *frames* constituem grafos *acíclicos* direcionados. Os exemplos a seguir, Figura 4.3, retirados de [RIC 93], ilustram a aplicação do algoritmo de herança de propriedades através de *é-um* e *instância*.

**Tabela 4.1** - Algoritmo Herança de Propriedades

---

Para recuperar um valor V para um “slot” S de um <i>frame</i> F, faça o seguinte:	
1.	Atribua vazio a CANDIDATOS.
2.	Execute uma busca em amplitude ou em profundidade a partir de F, subindo pela hierarquia <i>é-um</i> , seguindo todas as ligações <i>instância</i> e <i>é-um</i> . Em cada etapa veja se um valor de S ou uma de suas generalizações está armazenado.
a)	Se for encontrado um valor, acrescente-o a CANDIDATOS e encerre aquele ramo da busca.
b)	Se nenhum valor for encontrado, mas houver ligações <i>instância</i> ou <i>é-um</i> acima, siga-as.
c)	Caso contrário, encerre o ramo.
3.	Para cada elemento C de CANDIDATOS
a)	Veja se há qualquer outro elemento de CANDIDATOS que tenha sido derivado de uma classe mais perto de F do que a classe de onde C veio.
b)	Em caso afirmativo, remova C de CANDIDATOS
4.	Verifique a cardinalidade de CANDIDATOS
a)	Se for 0, então reporte que nenhum valor foi encontrado.
b)	Se for 1, então retorne o único elemento de CANDIDATOS como V.
c)	Se for maior do que 1, reporte contradição.

---

Fonte: Rich e Knight, 1993. Inteligência Artificial, p.315.

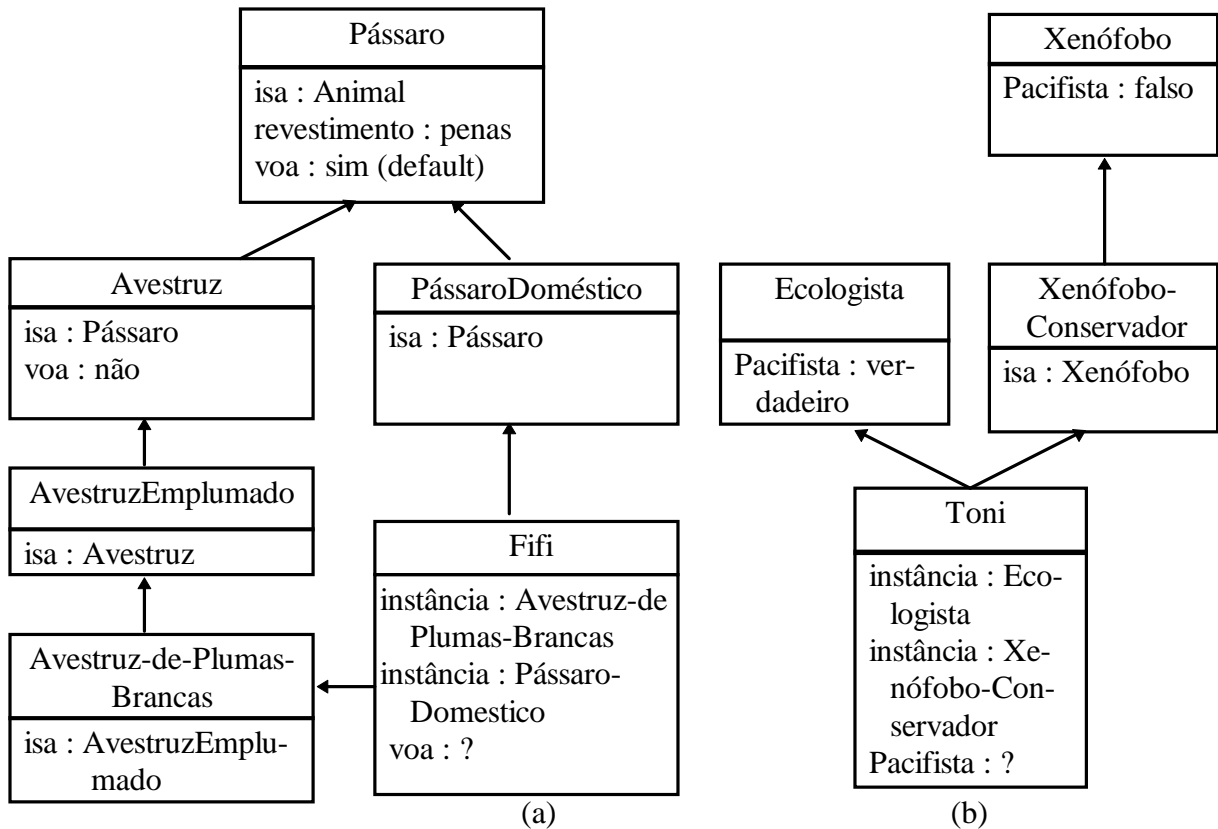
Para responder a pergunta “Fifi voa?” deve-se aplicar o algoritmo de herança de propriedades aos *frames* da Figura 4.3(a). Obtém-se a resposta *não* associada ao *frame* Avestruz e a resposta *sim* associada ao *frame* Pássaro. Como Avestruz tem menor distância inferencial até Fifi do que Pássaro, conclui-se que Fifi *não* voa.

Analogamente, para verificar se Toni é pacifista, aplicando-se o algoritmo de herança de propriedades à Figura 4.3(b), obtêm-se duas respostas: *não* associada ao *frame* Xenóforo e *sim* associada ao *frame* Ecologista. Como nenhuma delas está mais perto de Toni do que a outra, identifica-se corretamente uma contradição.

*Frames* e redes semânticas permite raciocínio (parcial) quando a informação disponível é incompleta e inferir rapidamente, através dos mecanismos de herança, fatos que não são observados diretamente na base de conhecimentos.

Para guiar o processo de raciocínio pode-se classificar os atributos de um objeto em propriedades *essenciais*, *complementares*, *negativas* e “*trigger*”. *Essenciais* são aquelas que necessariamente devem existir para que o objeto seja identificado como pertencente a uma determinada classe. Propriedades *complementares* representam características que podem auxiliar no reconhecimento do objeto como pertencente a classe em questão.

Propriedades *negativas* representam características que, quando existem, indicam que o objeto não pertencem a classe em questão. Propriedade “*trigger*” representa uma característica marcante que, quando presente, permite a imediata identificação do objeto [REÁ 92].



**Figura 4.3** - Herança de Propriedades Monovaloradas

Como visto com redes semânticas, o raciocínio baseado apenas em percorrer a hierarquia dos *frames* também pode desenvolver inferências que não são necessariamente válidas. Por isso o significado de um sistema de *frames* é estabelecido pelos procedimentos que o interpretam e não somente pelos *frames* em si.

A inferência por “*matching*” também é utilizada. Baseia-se na construção de *frames* que possuem “*slots*” com valores definidos e “*slots*” com valores desejados, mas desconhecidos, representados por variáveis. O procedimento de “*matching*” procura por *frames* que “casem” com os *frames* construídos e, quando os encontra, instancia as variáveis com os valores correspondentes dos *frames* encontrados. Esse tipo de inferência é análoga ao processo de unificação em regras de produção.

A popularização do uso de *frames* em IA levou ao desenvolvimento de diversas linguagens orientadas a *frames* que possuem meios eficientes para representação de conhecimentos declarativos, tais como KRL (Knowledge Representation Language), KL-ONE (Knowledge Representation - ONE), FRL-0 (Frame Representation Language - 0), etc. Também é possível implementar *frames* utilizando Lisp ou Prolog, obviamente às custas de maior esforço por parte do desenvolvedor da aplicação.

#### 4.1 Vantagens e Desvantagens

Em geral, as características abaixo se aplicam a redes semânticas e a sistemas de *frames*. O esquema de representação de conhecimento por redes semânticas permite representar relações estruturais através dos arcos *é-um* e *parte-de*. *Frames* representam relações estruturais através dos diversos tipos de relacionamentos, tais como *é-um*, *instância*, etc. Em ambos, toda a informação sobre um objeto está distribuída ao seu redor e é diretamente acessível a partir dele.

O uso de “*defaults*” e da herança de propriedades permitem raciocínio eficiente e minimizam redundâncias no armazenamento de dados, bastando representá-los na classe e não nos seus membros. É possível deduzir, de uma forma fácil e direta, fatos que não estão explicitamente representados.

*Frames* são úteis para projeto de grandes sistemas baseados em regras de produção. Representando-se regras através de *frames*, elas podem ser agrupadas em classes (baseadas por exemplo na ordem de ativação) de forma a facilitar a organização e indexação de conjuntos modulares de regras, de acordo com a utilização pretendida [RIC 93]. *Frames* pode facilitar a elicitação do conhecimento, por utilizar uma forma de representação de conhecimento similar a utilizada por muitos especialistas, para a representação do conhecimento em certos domínios estruturados tais como biologia ou paleontologia.

“*Slots*” podem ser modelados como objetos complexos e implementados como *frames*.

Como desvantagens, podemos citar:

- a) é difícil associar *comportamento* dependente do domínio aos *frames* (o que é feito através de referência em “*slots*” a procedimentos escritos em linguagens de programação utilizadas em IA). Já as redes semânticas só utilizam conhecimento declarativo;
- b) falta de uma semântica formal e de uma terminologia padrão;
- c) não existem regras rígidas e rápidas sobre a que tipos de objetos e relacionamentos o uso desses esquemas se presta melhor para representar conhecimento;
- d) pode ser difícil avaliar a correção das inferências realizadas, com base no conhecimento armazenado na rede semântica ou nos *frames*;
- e) uma rede semântica e, de forma análoga, um sistema de *frames*, torna-se muito complexa quando tentamos agregar maior capacidade de representação da estruturação dos nós, bem como das ligações, para representar propriedade de posse (alguém possui determinado objeto), aspectos temporais (período da posse do objeto), quantificação de variáveis, etc.

Os esquemas de representação do conhecimento redes semânticas e *frames* são genéricos. Possuem ligações e procedimentos de inferência especializados e rápidos, mas não existem regras estritas sobre que tipos de objetos e relacionamentos são bons, em geral, para representação do conhecimento; cabendo a quem está usando esses esquemas tomar essas decisões.

*Dependência conceitual*, *script* e *CYC* (Capítulo 5) são esquemas de representação do conhecimento, similares às redes semânticas e *frames*, mas que incorporam noções específicas sobre a que tipos de objetos e relações são mais adequados. Eles podem ser considerados com maior conteúdo semântico do que todos os anteriormente analisados.

## 5 Dependência Conceitual, Scripts e CYC

*Dependência conceitual* constitui uma teoria, com um número reduzido de primitivas, sobre como representar e raciocinar sobre *eventos* normalmente contidos em frases de linguagem natural. *Script* é um mecanismo que utiliza *frames* para representar uma *seqüência estereotipada de eventos*, em um determinado contexto. As *cenas* de um *script* são representadas utilizando as primitivas da *dependência conceitual*. CYC é uma teoria específica sobre como descrever o conhecimento comum do mundo, voltada para o desenvolvimento de bases de conhecimento de grande porte (i.é com milhões de objetos), permitindo a representação de *eventos, objetos, atitudes, tempo*, etc.

### 5.1 Dependência Conceitual

A dependência conceitual (DC), criada por Roger Schank entre 1969-75, representa o conhecimento sobre eventos contidos em frases, de modo a *facilitar inferências* e que a representação seja *independente da linguagem natural* na qual são expressas. Para tanto, não utiliza primitivas que correspondam às palavras da frase, mas sim um conjunto de primitivas conceituais que podem ser combinadas para formar os significados das palavras em qualquer linguagem natural. As representações de *ações* são criadas a partir de *primitivas* da Tabela 5.1 [Schank, 1975a; Schank and Abelson, 1977 Apud BAR 86].

**Tabela 5.1** - Ações Primitivos

AÇÃO- ACT	SIGNIFICADO
	<i>Atos Físicos</i>
GRASP	Domínio físico de um objeto por um ator (p. ex., agarrar)
EXPEL	Expulsão algo do corpo de um animal ou ator (p. ex., chorar)
INGEST	Ingestão de algo por um animal ou ator (p. ex., comer)
MOVE	Movimentação de parte do corpo (p. ex., chutar)
PROPEL	Aplicação de força a um objeto físico (p. ex., empurrar)
	<i>Atos caracterizados pela mudança de estado resultante</i>
ATRANS	Mudança em relação abstrata, relativo a um objeto (p. ex., dar, possuir)
PTRANS	Mudança de local de um objeto (p. ex., ir, correr, colocar)
	<i>Atos usados como instrumentos para outros atos</i>
ATTEND	Direcionamento de um órgão de sentido para um estímulo (p.ex. escutar)
SPEAK	Produção de sons (p. ex., falar)
	<i>Atos mentais</i>
MBUILD	Construção de novas informações a partir de antigas (p. ex., decidir)
MTRANS	Transferência de uma informação mental (p. ex., contar, ver, ouvir)

Fonte: Barr, 1986. The Handbook of AI, p.212

Além das primitivas ACT, Schank também considera o *tempo*, o *local* e as categorias: PP, *Picture Producer* (atores, objetos físicos, forças naturais ou as divisões da memória humana: processador do consciente, memória intermediária e de longo termo); AA, *Action Aiders* (modificadores - atributos - de ações); PA, *Picture Aiders* (modificadores - atributos - de PP); as quais podem ser combinadas em representações denominadas *conceitualizações*. As conceitualizações básicas são: a) um ator (PP) fazendo uma primitiva ACT e b) um objeto (PP) com uma descrição de seu estado (PA). Os elementos primiti-

vos de conceitualizações não são palavras, mas conceitos que refletem um nível de pensamento subjacente a linguagem, em vez da linguagem em si [BAR 86]. As dependências entre conceitualizações (Figura 5.1) correspondem às relações semânticas entre esses conceitos subjacentes, independentemente da linguagem original.

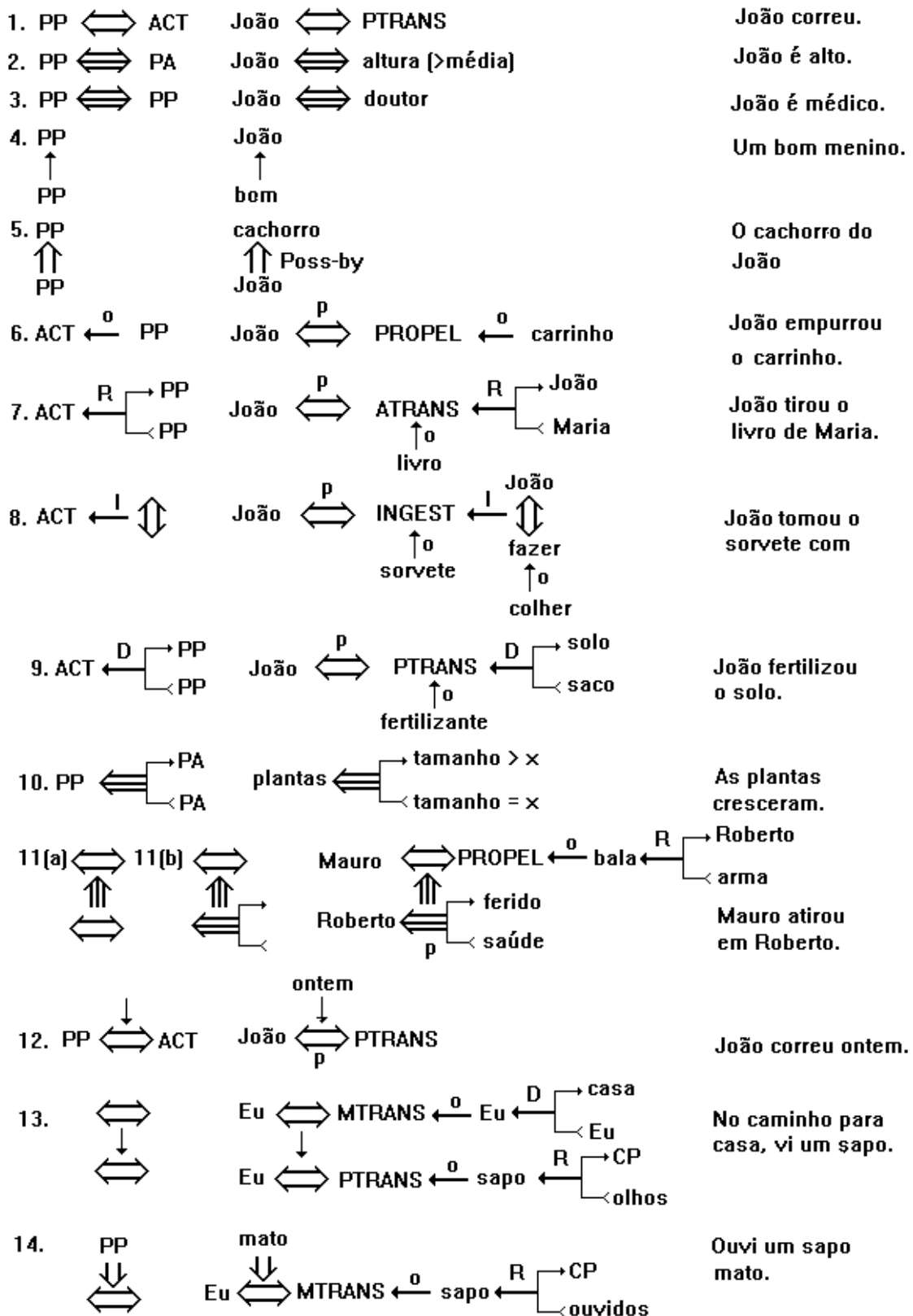


Figura 5.1 - Dependências entre Conceitualizações

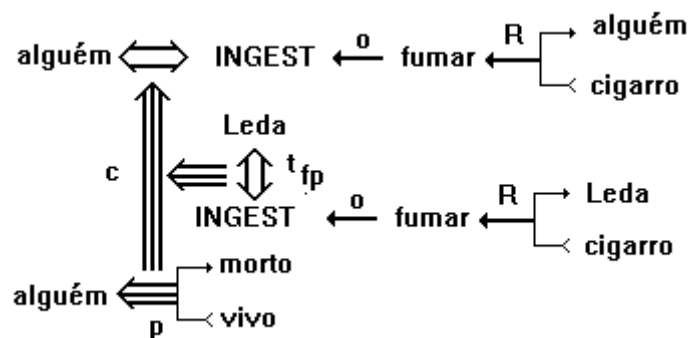
Observe que o uso do conjunto de primitivas generaliza as representações de dependência conceituais. Assim a representação da Figura 5.1-1 tanto pode está associada ao evento “João correu” como ao “João saltou”, pois *correr* e *saltar* são transferência de objeto físico de local, ambas representadas por PTRANS.

As conceitualizações que representam eventos (ações) podem ser modificadas de várias maneiras para fornecer as informações da linguagem natural relativas aos tempos, modos e aspectos verbais. A Tabela 5.2 apresenta o conjunto de *tempos conceituais* proposto por Schank [RIC 93].

Tabela 5.2 - Tempos Conceituais	
Modificador	Significado
p	Passado
f	Futuro
t	Transição
t <sub>s</sub>	Iniciar transição
t <sub>f</sub>	Transição encerrada
k	Continuidade
?	Interrogação
/	Negação
nil (nenhum)	Presente
delta	Infinito
c	Condicional

Fonte: Rich e Knight, 1993. Inteligência Artificial, p.323.

A Figura 5.2 ilustra a representação, em dependência conceitual, da frase “Já que fumar pode matar quem fuma, Leda parou de fumar.” [RIC 93].



**Figura 5.2** - Exemplo de Representação usando DC

A ligação de causalidade vertical indica que “fumar pode matar alguém”, pois a presença do modificador **c**, transforma a relação de causa em uma possibilidade. O elo de causalidade horizontal indica que essa primeira causa foi o motivo que fez Leda parar de fumar. A qualificação **t<sub>fp</sub>** indica que a dependência entre Leda e **INGEST** parou e que esta parada aconteceu no passado.

A Tabela 5.3 apresenta a interpretação das *dependências* entre conceitualizações, apresentadas na Figura 5.1.



**Tabela 5.3** - Interpretação das Dependências entre Conceitualizações

Regra	Relação entre
1	<i>Ator</i> (PP) e a <i>ação</i> (ACT) que ele causa. Como nem o <i>ator</i> nem a <i>ação</i> pode ser considerado primário, a dependência é bidirecional.
2	<i>Objeto</i> (PP) e um <i>atributo</i> (PA) que o descreve. Em DC, muitas descrições de estado são representadas como escalas numéricas.
3	Dois <i>objetos</i> (PP), um dos quais pertence ao conjunto definido pelo outro.
4	<i>Objeto</i> (PP) e um <i>atributo</i> (PP) que já tinha sido seu predicado. A seta aponta para o PP que está sendo descrito.
5	Dois <i>objetos</i> (PP), um dos quais fornece uma informação sobre o outro, sendo que as mais comuns são: <i>posse</i> (Poss-by), <i>localização</i> (Loc) e <i>contenção física</i> (Cont). A seta aponta para o <i>conceito</i> sendo definido.
6	<i>Ação</i> (ACT) o seu (o) <i>objeto</i> (PP). A seta aponta para a <i>ação</i> , já que o contexto determina o significado da relação com o <i>objeto</i> .
7	<i>Ação</i> (ACT) e a sua <i>fonte</i> e o (R)ecipiente.
8	<i>Ação</i> (ACT) e o (I)strumento com que é executada. O <i>instrumento</i> precisa ser uma <i>conceitualização completa</i> (i.é conter uma ACT) e não ser apenas um simples objeto físico.
9	<i>Ação</i> (ACT) e a sua <i>fonte</i> e o (D)estino físicos.
10	<i>Objeto</i> (PP) e o seu estado <i>inicial</i> e <i>final</i> (mudança de estado).
11	Uma <i>conceitualização</i> e outra que a <i>motiva</i> . As formas apresentadas descrevem a <i>causa de uma ação</i> e a <i>causa de uma mudança de estado</i> . A seta aponta para a <i>causa</i> .
12	<i>Evento</i> e o <i>momento</i> em que ocorreu.
13	Uma <i>conceitualização</i> e outra que é o <i>momento</i> da primeira. O exemplo ilustra a forma de tratar <i>processamento das informações humanas</i> : <i>ver</i> é representado como uma transferência de informação entre os <i>olhos</i> e o <i>processador do consciente</i> (CP). CP é uma abstração utilizada em DC.
14	Uma <i>conceitualização</i> e o <i>lugar</i> em que ocorre.

Fonte: Rich e Knight, 1993. Inteligência Artificial, p.321.

### 5.1.1 Vantagens e Desvantagens no uso de dependência conceitual

O uso do esquema dependência conceitual apresenta as seguintes vantagens:

- é adequado para representação de eventos que normalmente ocorrem em frases da linguagem natural;
- apresenta um conjunto de primitivas que tornam a representação independente da linguagem natural de origem;
- facilita o raciocínio porque menos regras de inferência são necessárias. Só é necessário representada uma regra de inferência para cada primitiva de *ação* (veja Tabela 5.1) e não uma para cada palavra que descreve aquela *ação*;
- muitas inferências já estão presentes na própria representação, em decorrência do uso das estruturas de dependência entre conceitualizações.

Mas, de acordo com Schank e Owens [Apud RIC 93, p.326], “A DC é uma teoria para a representação de *ações relativamente simples*. Para expressar, por exemplo: ‘João apostou cinquenta mil reais com Samuel que o São Paulo seria campeão brasileiro’, seriam necessárias, aproximadamente, duas páginas de formalismo em dependência conceitual. Isto não parece razoável.”

Este fato é devido a utilização de primitivas de nível muito baixo, com todas as desvantagens que isso acarreta (veja seção 1.3.2). Outra desvantagem do uso de dependência conceitual, como modelo genérico para representação do conhecimento, é que ela é apenas uma teoria da representação de *eventos*. Um esquema genérico precisa ter a capacidade de representar outras coisas, além de eventos. Assim foram propostas alguns conjuntos de primitivas, similares ao conjunto de *ações* da DC, para representar outras coisa, tais como *objetos físicos* [Lehnert, 1978 Apud RIC 93], atualmente representados em DC como unidades atômicas, e *ações sociais* [Schank e Carbonell, 1979 Apud RIC 93], não modelados explicitamente em DC. Representações em DC podem ser vistas como instâncias de redes semânticas, com mecanismo de inferência mais especializados.

A teoria de dependência conceitual intenciona ser uma teoria intuitiva do processamento de linguagem humana e não apenas um suporte para programas que processem linguagem natural. Assim uma representação em DC é sempre não ambígua e única, mesmo que as frases originais, com o mesmo conteúdo conceitual, possuam ambigüidades.

Schank desenvolveu o MARGIE (Meaning Analysis, Response Generation, and Inference on English) em 1975, com o objetivo de prover um modelo intuitivo do processo de entendimento da linguagem natural. Nele, DC foi utilizada como uma *interlingua*, um esquema de representação independente da linguagem, para codificação de sentenças. Todo o processamento posterior ao processamento da sentença e geração da sua representação em DC, era feito com base apenas na representação em DC. A existência de uma representação canônica para todas as sentenças com o mesmo significado facilitava algumas tarefas, tais como responder questões e parafrasear [BAR 86].

Outros exemplos de sistemas que utilizam DC são o SAM (Script Applier Mechanism), desenvolvido por Roger Schank e Robert Abelson em 1977, e o PAM (Plan Applier Mechanism), desenvolvido por Wilensky (aluno de doutorado de Schank e Abelson) em 1978. Ambos aceitam histórias como entrada e geram uma representação interna da mesma em DC, podendo parafrasear as histórias e fazerem inferências inteligentes a partir das representações em DC. SAM entende as histórias, enquadrando-as em *scripts* pré-definidos e faz um resumo das mesmas. Baseia-se em expectativas construídas com base nos scripts. PAM entende as histórias determinando quais são as metas a serem nelas atingidas e tentando relacionar as ações implicadas na realização dessas metas, com métodos que o sistema sabe que irão resultar na realização delas [BAR 86].

## 5.2 Scripts

Scripts foram desenvolvidos por Schank e Abelson para uso no SAM (seção 5.1). Segundo Bartlett [Bartlett 1932 Apud BAR 86, p.216] “existe evidência psicológica abundante de que as pessoas usam conhecimento assimilado de experiências prévias para interpretar novas situações em suas atividades cognitivas diárias”. Ao irmos a um restaurante, onde nunca estivemos, temos um conjunto de expectativas sobre o que encontraremos, baseados na nossa experiência em outros restaurantes. Essas expectativas envolvem *objetos* típicos (mesas, garçons, etc.) e a *seqüência de eventos* que ocorreram (obter uma mesa, fazer o pedido, etc.). *Frames* e scripts (uma especialização de *frames*) são métodos que permitem organizar a representação de conhecimento de uma forma que focaliza a atenção e facilita recuperar e fazer inferências com o conhecimento. Scripts são adequados para representação do conhecimento sobre *seqüências comuns de*

*eventos*, em um certo contexto. *Scripts* consistem em um conjunto de estruturas similares aos *frames* (seção 4.2), com “*slots*” e valores “*default*”, mas que possuem um papel mais especializado. São úteis porque, no mundo real, os eventos ocorrem segundo padrões, que aparecem em decorrência do relacionamento causal entre eventos. Assim, agentes executam uma ação para poderem depois executar uma outra ação. Os eventos descritos no script forma uma *cadeia causal*, cujo início é constituído pelas *condições de entrada* e o final pelos *resultados*. Dentro da cadeia, os eventos são conectados tanto ao eventos anteriores que os ativaram, quanto aos eventos posteriores, por eles ativados.

Assim um script característico [RIC 93] é composto de:

- *condições de entrada*, que em geral precisam ser satisfeitas antes da ocorrência dos eventos descritos no script;
- *resultado*, condições que em geral serão verdadeiras após a ocorrência dos eventos descritos no script;
- *acessórios*, i.é *frames* que representam objetos envolvidos nos eventos descritos no *script* (de forma explícita ou não);
- *papéis*, i.é “*slots*” que representam pessoas envolvidas (explicitamente) nos eventos descritos no script, ou cuja presença pode ser inferida. Se indivíduos específicos forem mencionados, poderão ser inseridos nos *frames* apropriados;
- *trilha*, i.é uma variação específica do padrão mais genérico, representado pelo script em questão;
- *cenas*, i.é as seqüências de eventos que ocorrem no script. Os eventos podem ser representados no formalismo de dependência conceitual.

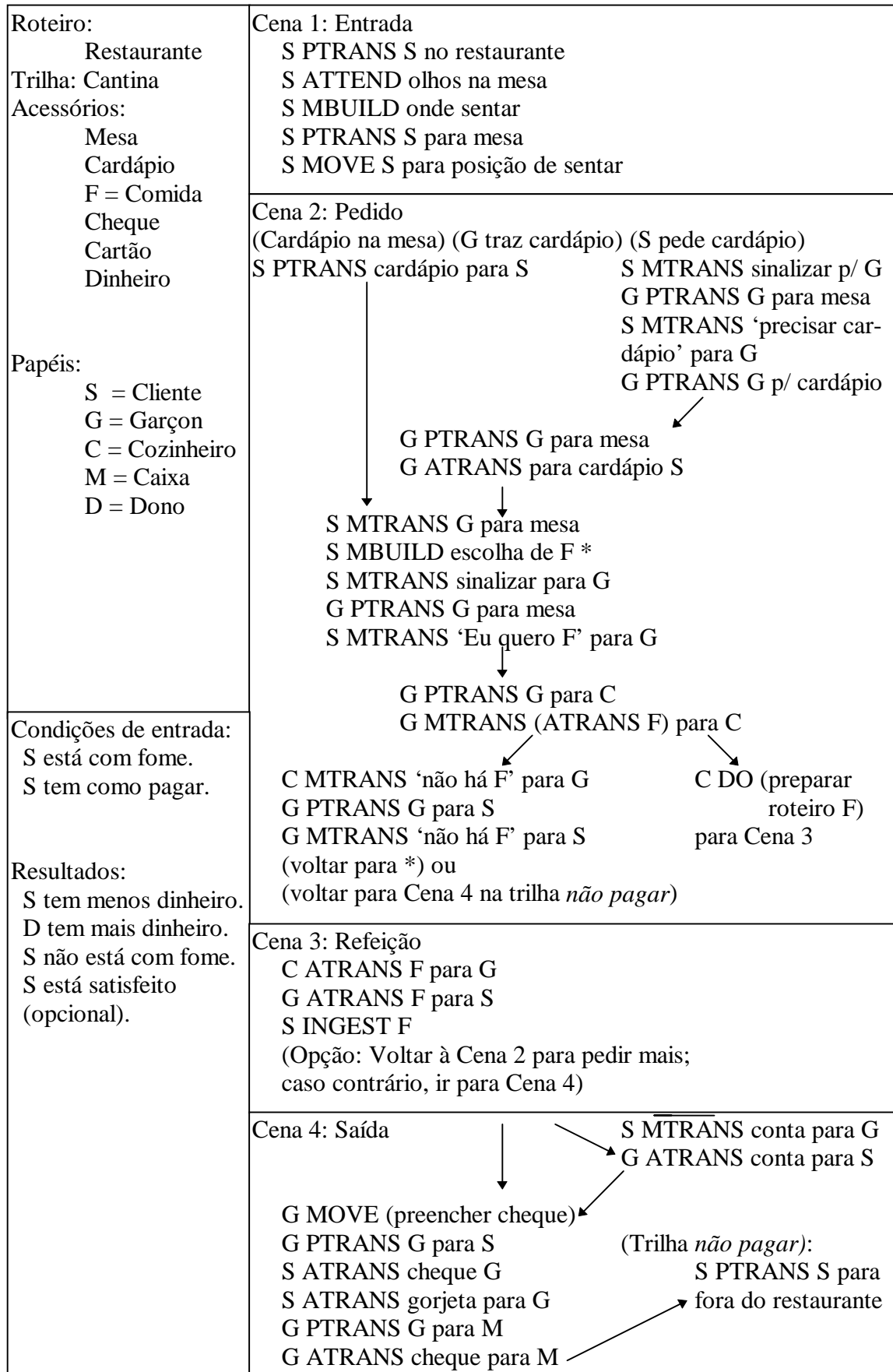
Um script pode ser ativado de duas maneiras, dependendo da sua importância no contexto no qual se raciocina:

- apenas armazenar um ponteiro para o script, de forma que se possa ter acesso a ele mais tarde, se necessário, quando se tratar de scripts transitórios. *Script transitório* é aquele mencionado de passagem e que pode voltar a ser citado no futuro, mas que não é central à situação atual.
- ativar o script inteiro, se tratando de script não-transitório e tentar preencher os *frames* com objetos e pessoas envolvidas na situação atual. Para reduzir o número de vezes em que um script é ativado desnecessariamente, pode-se ativá-lo somente se a situação contiver, no mínimo, dois cabeçalhos. *Cabeçalhos* são as *condições de entrada*, *acessórios*, *papéis*, *locais preferidos* e *eventos* de um script.

A Figura 5.3 [RIC 93] ilustra um script possível, seguido quando se deseja ir a um restaurante. Deve haver outras *trilhas* para permitir pagamento da conta através de dinheiro ou cartão de crédito, ao invés de apenas cheque.

Após a ativação do script, ele pode ser utilizado para:

- prever eventos que não foram observados explicitamente; porque pode-se inferir que a seqüência de eventos previstos pelo script correspondem aos eventos da situação que possibilitou a sua ativação e que eles irão ocorrer normalmente;
- possibilitar a criação de interpretação coerente para um grupo de observações, visto que o script pode ser visto como uma grande *cadeia causal*, fornecendo informações sobre como os eventos estão relacionados entre si;
- enfatizar *eventos incomuns*, os quais se desviam da seqüência esperada de eventos previstos no script.



**Figura 5.3** - Exemplo de Script

Scripts, assim como *frames*, são exemplos do sistema de *representação estrutural*, mas a inclusão de procedimentos (conhecimento dinâmico) nos “*slots*” permite o uso de *representação procedimental* do conhecimento, através dos “*demons*” (veja seção 4.2). Por outro lado, a atribuição de valor estático a um “*slot*” permite o uso de uma *representação declarativa* do conhecimento: a asserção de que um fato (atributo e seu valor) é verdadeiro.

Além de direcionar a obtenção de outras informações, o preenchimento dos “*slots*” do script fornece uma confirmação ou não de que o script sendo seguido é apropriado para o entendimento da cena ou evento sendo processado. Se o script for considerado inadequado, procedimentos em “*demons*” podem ser disparados para transferir o controle para outro script. Por exemplo, o programa SAM (Script Applier Mechanism) de Schank e Abelson, utilizado para processar histórias com base em scripts, entende um história escrita quando cada “*slot*” do script apropriado, é preenchido com um evento (explícito ou implícito) da história [BAR 86].

### 5.2.1 Vantagens e Desvantagens com o uso de Scripts

O uso de script como esquema genérico de representação do conhecimento não é aconselhável, pois são estruturas menos genéricas e flexíveis do que redes semânticas ou *frames*, embora semelhantes. São bastantes eficazes para a representação de sequência de eventos e, se um script em particular é sabidamente apropriado para uma determinada situação, a sua utilização é muito útil para prever a ocorrência de eventos não explicitamente mencionados, ajudar na indicação de como os eventos se relacionam entre si e auxiliar a identificação de eventos incomuns, i.é aqueles que quebram a sequência prevista.

Em seus experimentos de entendimento de histórias com o uso do programa SAM, Schank e Abelson [Apud BAR 86, p.308] observaram problemas quando:

- a) *os eventos da história não eram completos*, como “Eu fui a um restaurante. Comi um hambúrguer. Então comprei alguns doces.”. Com a última sentença, não fica claro para o sistema se o script do restaurante terminou e começou um novo script relativo a uma doceria ou se os doces foram comprados no próprio restaurante.
- b) *um fato posterior da história aborta o acompanhamento de um script*: “Eu fui ao posto de gasolina encher o tanque do meu carro, mas o dono me disse que não tinha mais gasolina.”.
- c) *em situações com referências implícitas*, perfeitamente compreendidas pelas pessoas, mas sem informação suficiente para que o sistema invoque um script: “João necessitou dinheiro. Ele pegou uma arma e foi para uma loja de bebidas.”

Eles concluíram que “Deve existir conhecimento relevante disponível para ligar sentenças pois senão elas não teriam ligação. O problema é que existe uma grande quantidade de histórias onde a ligação não pode ser estabelecida por técnicas de encadeamento causal ou por referência a um script. Entretanto elas são obviamente conectadas. Essa conexão deriva da referência implícita a planos.” [BAR 86, p.309]. Para esses pesquisadores um *plano* é caracterizado pelo ator, suas metas e os métodos por ele escolhidos para atingi-las. Assim, em uma história baseada em planos, o sistema deve discernir quais são

as metas do ator principal e as ações que resultaram na consecução delas. O PAM - Plan Applier Mechanism, implementou essa abordagem para o processamento de histórias.

### 5.3 CYC

CYC é uma teoria proposta por Lenat e Guha [Lenat e Guha, 1990 Apud RIC 93] sobre como captar e representar conhecimento humano de senso comum em grandes bases de conhecimento. Essa teoria é especialmente voltada para questões de escala, decorrentes de se ter uma base de conhecimentos com milhões de objetos. CYC e DC representam teorias específicas sobre como descrever o mundo, mas CYC é mais abrangente do que DC, pois além de representar eventos também permite a representação de objetos, atitudes, tempo, etc. Ambas podem ser úteis para a construção de sistemas inteligentes com capacidade de interpretação de linguagem natural.

A criação de uma grande base de conhecimentos é tarefa muito complexa. Idealmente os conhecimentos poderiam ser adquiridos automaticamente através do uso de técnicas de *aprendizagem automática* por máquina ou *compreensão da linguagem natural*. Mas as técnicas atuais de aprendizagem automática permitem apenas extensões modestas do conhecimento de um programa e, para que um programa aprenda bastante, ele já precisa saber bastante [RIC 93]. Os métodos de interpretação de linguagem natural também precisam de conhecimento intenso. Embora existam diversas versões de enciclopédias e dicionários *on-line*, um programa atual de computador, se alimentado com esses dados, não consegue assimilá-los. Assim, para um programa assimilar o conhecimento contido em uma enciclopédia, ele precisa já saber muita coisa sobre o mundo. Para testar a sua teoria, Lenat e Guha estão construindo uma grande base de senso comum, que após ser construída, poderá ser combinada com bases de conhecimentos especializados para produzir sistemas inteligentes mais robustos do que a maioria dos disponíveis hoje. Os atuais sistemas baseados em conhecimentos não conseguem enfrentar situações novas e o seu conhecimento torna-se obsoleto rapidamente. Espera-se que os programas criados em cima de profundos conhecimentos de senso comum, sejam mais adequados. A equipe deste projeto optou por codificar manualmente cerca de 10 milhões de fatos, que consideram compor o conhecimento de senso comum. Após essa etapa, imagina que será possível utilizar métodos mais automatizados. Todo esse conhecimento está sendo codificado em uma linguagem de representação chamada CYCL.

Segundo [RIC 93], *CYCL* é uma linguagem de representação, baseada em *frames*, que:

- generaliza o conceito de herança: as propriedades possam ser herdadas ao longo de qualquer ligação e não apenas nas relações *é-um* e *instância*, podendo ainda, serem aplicadas a uma *cadeia* de relações;
- incorpora mais formas de relacionar classes como: herança múltipla, mutuamente-disjunto-de, *é-coberto-por*, transfere-através e representação de atributos como *frames*;
- contém uma linguagem de restrições que permite expressar fórmulas da lógica de primeira ordem, indicadas pelo “*slot*” restrições. Ele aponta para um *frame* no qual as fórmulas são codificadas no “*slot*” valor-do-Slot-Inclui, sendo a direção de propagação indicada pelo atributo direção-de-Propagação;
- realiza raciocínio “*default*”, com momento da execução das regras de restrições determinado pelo atributo direção-de-Propagação. Se for “*backward*” a regra será do tipo “*se-necessário*” e se for “*forward*”, a regra será do tipo “*se-adicionado*”;

- implementa os níveis de representação epistemológico (NE) - fatos expressos em linguagem de restrições lógicas - e heurístico (NH) - mesmos fatos armazenados em modelos de inferência eficientes;

- executa verificação *de consistência e resolução de conflitos*.

Vamos analisar cada uma dessas características. As declarações abaixo, extraídas de [RIC 93], ilustram o conceito de herança estendida da CYCL:

1. Todos os pássaros têm duas pernas

*Herança padrão* através da relação instância: o atributo número-de-pernas de qualquer *frame* de um pássaro específico, herda valor dois do *frame* da classe pássaro, através da relação *instância*, que o liga à essa classe.

2. Todos os amigos de Maria falam espanhol

*Herança estendida* através da relação amigo: o atributo línguas-faladas de qualquer *frame* de um amigo específico de Maria, herda valor espanhol do *frame* Maria (que possui a restrição de que todos os seus amigos falem espanhol), através da relação *amigo*, que o liga à Maria.

3. Todos os pais dos amigos de Maria são ricos

*Herança estendida* através de composição de relações: o atributo situação de qualquer *frame* de um pai de amigo específico de Maria, herda o valor rico através de uma composição das relações *amigo e pai-de*.

Para o raciocínio “*default*”, se a *direção-de-Propagação* for “*backward*”, as regras codificadas no *valor-do-Slot-Inclui*, serão invocadas sempre que alguém inquirir sobre o valor do atributo que é restrito. Essas regras são do tipo *se-necessário* e serão utilizadas para determinar o(s) valor(es) a ser(em) colocado(s) no *slot* restrito.

Se a *direção-de-Propagação* for “*forward*”, as regras do *valor-do-Slot-Inclui* serão disparadas sempre que ocorrer uma adição à base de conhecimento. Se a adição satisfizer as restrições impostas pelas regras, ela será automaticamente propagada para o “*slot*” restrito. As regras são do tipo *se-adicionado*.

Os *frames* a seguir, extraídos de [RIC 93], ilustram o uso da *linguagem de restrições*. Eles representam a declaração “Maria gosta de pessoas que só programam em LISP”:

Maria	% <i>frame</i> Maria
gosta-de:	???
restrições:	(restrições-Lisp)
restrições-Lisp	% <i>frame</i> de restrições, associado ao “ <i>slot</i> ” gosta-de
slot-Restringido:	(gosta-de)
valor-do-Slot-Inclui:	
	(o-conjunto-de X (pessoa todas-Instâncias)
	(and (programa-Em X linguagem-Lisp)
	(not (existe Y (linguagens todas-Instâncias)
	(and (not (equal Y linguagem-Lisp))
	(programa-Em X Y) ) ) ) )
direção-de-Propagação:	forward
 Beto	 % <i>frame</i> Beto
programa-Em:	(linguagem-Lisp)

Jane	% <i>frame</i> Jane
programa-Em:	(linguagem-Lisp linguagem-C)

O “*slot*” valor-do-Slot-Inclui, do *frame* restrições-Lisp, assegura que o “*slot*” gosta-de do *frame* Maria será preenchido com aqueles indivíduos que programem apenas em linguagem-Lisp. No exemplo a regra infere que Maria gosta de Beto e não de Jane.

O CYC possui um Sistema de Manutenção da Verdade (veja Capítulo 7) para garantir a consistência dos “*slots*” com restrições. Assim se Beto começar a programar em C, deixará de aparecer no “*slot*” gosta-de do *frame* Maria.

As regras “*forward*” exigem tempo e espaço substanciais para propagar seus valores. As regras “*backward*” retardam o raciocínio até que os valores nelas contidos, sejam requisitados. CYC mantém um processo rodando em segundo plano (“*background*”) para realizar as propagações “*forward*”. Assim pode-se continuar a inserir conhecimentos enquanto seus efeitos são propagados durante os intervalos em que o teclado fica ocioso.

O *nível epistemológico* (NE) da CYCL contém fatos expressos na linguagem de restrições lógicas, enquanto o *nível heurístico* (NH) contém os mesmos fatos representados através de modelos de inferência mais eficientes. Existe um programa de tradução para conversão de uma declaração representada em um nível no outro. Embora a linguagem de restrições permita expressar fatos como fórmulas lógicas de primeira ordem arbitrárias, o raciocínio lógico genérico é muito complexo em termos computacionais. Já a inferência baseada em *frames* é bastante eficiente do ponto de vista computacional. A existência dos dois níveis de representação tira partido dessas vantagens.

A *verificação de consistência* da CYCL detecta quando um valor ilegal é colocado em um “*slot*”. Cada mecanismo de inferência utilizado tem o seu próprio recurso de manutenção da verdade. Já a *resolução de conflitos* permite detectar casos em que procedimentos de inferência múltipla atribuem valores incompatíveis a um mesmo “*slot*”.

Muitos dos mecanismos de inferência usados por CYC são armazenados explicitamente na base de conhecimentos, como modelos no nível epistemológico. Eles podem ser modificados, como qualquer outro *frame*, e o usuário pode editá-los e modificá-los. A cada modelo NE de mecanismo de inferência está associado um código LISP, gerado automaticamente pelo sistema.

Do ponto de vista de implementação, CYC é um sistema multiusuário com interface textual e gráfica com a base de conhecimentos. Uma consulta pode ser associada com um nível de esforço. No nível mais baixo, CYC simplesmente verifica se o fato está armazenado na base de conhecimentos. Nos níveis mais altos, são invocados procedimentos de inferência “*backward*” e considera cadeias de inferência [RIC 93].

As modificações que o usuário faz na base de conhecimentos são transmitidas a um servidor central, onde são verificadas e depois propagadas aos outros usuários.



Do ponto de vista da ontologia<sup>8</sup> o conceito de mais alto nível em CYC é denotado *Coisa*. Todo que é representado é uma instância de *Coisa*. Abaixo desse conceito aparecem *Coleção* e *Objeto-Isolado*. Eles não têm instâncias em comum.

*Coleção* é a maior superclasse dos conjuntos. Apenas ela pode ter superconjuntos e subconjuntos; qualquer coisa que seja sua instância é também um subconjunto de *Coisa*.

*Objeto-Isolado* é composto de objetos isolados. Apenas as suas instâncias podem ter partes. Dentre as suas instâncias destacam-se *Objeto-Tangível* e *Substância*. *Objeto-Tangível* são coisas com massa, mas sem nenhum aspecto intangível. *Substância* é algo que mantém suas propriedades quando cortado em pedaços menores. CYC armazena, para cada instância de *Substância*, a unidade de tamanho ou grânulo.

As instâncias de *Objeto-Intangível* são coisas sem massa. Também existe o conceito *Objeto-Composto*, cujas instâncias possuem dois “slots” chaves: extensão-Física e extensão-Intangível. Uma pessoa é um *Objeto-Composto* cuja Extensão-Física é o seu corpo e cuja Extensão-Intangível é a sua mente.

*Slot* é uma subclasse de *Objeto-Intangível*. *Slot-de-Contabilidade* registra informações de contabilidade (*account*), tais como data de criação do *frame*, usuário que o criou, etc. *Slot-de-Definição* refere-se as propriedades do objeto representado pelo *frame*. *Slots-Quantitativos* assumem apenas uma faixa escalar de valores, p. ex. altura.

*Agente* é uma subclasse de *Objeto-Composto* e representa a coleção dos seres inteligentes. Podem ser coletivos ou individuais. Todos eles possuem a propriedade crença. As crenças de um agente podem estar certas ou incorretas. O CYC distingue os fatos de sua própria base de conhecimentos e dos “fatos” dos agentes. Em geral, um agente imputa as suas crenças a outros agentes, a fim de facilitar a comunicação entre eles.

As coisas possuem propriedades Intrínsecas e Extrínsecas. Uma propriedade é *Intrínseca* se, quando um objeto possui essa propriedade (p.ex. cor), todas as suas partes também a possuem; caso contrário é *Extrínseca*. As propriedades Intrínsecas normalmente são herdadas das Substâncias e as Extrínsecas, normalmente de *Objeto-Isolado*. Se uma coisa tem extensão temporal, tipo andar, estudar, ela é um *Evento*. Os Eventos podem ter propriedades temporais como duração e Começa-Antes-de. Existem dois tipos de medidas temporais: intervalos e conjuntos de intervalos. Os conjuntos de intervalos são criados através da aplicação de operações como união e interseção a intervalos básicos, permitindo expressar fatos como “João vai ao cinema à 3 da tarde todo o domingo”.

### 5.3.1 Vantagens e Desvantagens com o uso de CYC

CYC não é um esquema de representação do conhecimento de uso geral, devido à complexidade do seu ambiente computacional e a baixa portabilidade decorrente. Atualmente situa-se mais no contexto de um grande projeto de pesquisa, no qual são feitas experimentações visando adquirir maior domínio sobre técnicas de representação de conhecimento e construção de grandes bases de conhecimentos. A filosofia por trás do CYC é representar coisas como espaço, tempo, movimento e estrutura em um nível

<sup>8</sup> Estudo filosófico do que existe. Em IA preocupa-se com as categorias que podem ser quantificadas e como elas se relacionam entre si.

muito alto, mas de uma forma acurada. Uma vez que essa representação esteja disponível, sistemas específicos mais eficientes poderão ser construídos, sem muito ônus. Esses sistemas, por compartilharem as mesmas primitivas, poderão comunicar-se entre si facilmente.

CYC apresenta uma ontologia bastante completa e útil para modelar conhecimento humano de senso comum.

## 6 Esquemas de Representação de Propósito Específico

Neste capítulo são apresentados a técnica de pesquisa em *espaço de estados* e o esquema de *representação direta*. A representação em espaço de estados foi o primeiro formalismo de representação largamente utilizado em IA, tendo sido desenvolvido para programas de resolução de problemas e para jogos. O esquema de representação direta (ou analógica) é uma classe composta por mapas, modelos, diagramas, pauta de música, etc. que pode representar conhecimento sobre certos aspectos do mundo de uma forma bastante natural [BAR 86].

### 6.1 Espaço de Estados

A técnica de busca em espaço de estados não é um esquema de representação de conhecimento *per se*, pois o que ela permite representar é a estrutura do problema em termos das alternativas disponíveis (i.é legais) em cada estado possível do problema. Em um jogo, significa os movimentos alternativos disponíveis após cada jogada [BAR 86].

Essa técnica é caracterizada pela existência de um conjunto de estados iniciais, ponto de partida através do qual, usando um conjunto de regras para ir de um estado para outro, tentamos terminar em um estado do conjunto de estados finais desejados (i.é aqueles que seriam aceitáveis como solução para o problema). A idéia básica é poder gerar todos os estados possíveis, aplicando um pequeno conjunto de regras, chamado *operadores de transição* (ou geradores de movimentos legais) a um estado qualquer. Cada regra deve conter informações sobre o que precisa ser verdadeiro antes da ação (decorrente da sua aplicação) ocorrer, e como e o que será alterado por esta ação.

Um espaço de estados pode ser tratado como um grafo direcionado cujos nós são os estados e cujos arcos são os operadores de transformação de um estado em outro. A solução para uma busca no espaço de estados é um caminho de um nó inicial para um nó que representa um estado desejado [BAR 86]. Note que a busca em espaço de estados corresponde de fato a uma técnica de solução de problemas, pois segundo [RIC 93, p.37]:

- a) permite definir um problema como sendo a necessidade de converter uma dada situação na situação desejada, usando um conjunto de operações permitidas;
- b) permite definir o processo de solução de um determinado problema como uma combinação de técnicas conhecidas (cada uma representada por uma regra que define uma única etapa no espaço) e busca, sendo que a técnica geral é explorar o espaço para tentar descobrir o percurso do estado atual para o estado desejado.

Em um jogo, uma forma de obter o movimento vencedor é determinar todos os movimentos alternativos (a partir de uma configuração qualquer), então tentar todos os movimentos possíveis, decorrentes da jogada de resposta do oponente, e assim por diante até que todas as configurações possíveis do jogo tenham sido testadas e verificada qual é a ótima. Quando houver muitas combinações possíveis de movimentos, pode ocorrer uma explosão combinatorial do número de alternativas possíveis e a capacidade computacional do sistema pode ser insuficiente para fornecer uma resposta em tempo hábil.

A solução é limitar o número de alternativas pesquisadas, em cada estágio do processo de análise em diversos níveis de profundidade da melhor alternativa.

Essa abordagem permite resultados ótimos em jogos como torre de Hanói mas não produz bons resultados em xadrez. Para jogos (ou problemas) complexos como o xadrez é necessário, além de limitar o número de níveis de geração das alternativas possíveis, acrescentar ao sistema algum conhecimento sobre o domínio e capacidade dele raciocinar com base nesse conhecimento, além de utilizar um método de busca mais eficiente do que a busca exaustiva.

Existem diversos algoritmos de busca em espaço de estado, além da busca exaustiva (cega), tal como busca com custo uniforme, busca em profundidade, busca em abrangência, busca bidirecional e busca heurística. As buscas heurísticas utilizam informação adicional sobre as propriedades do domínio do problema específico, além daquelas que são necessárias para construir os estados e as definições dos operadores. Essa informação é utilizada para ajudar a reduzir o esforço computacional durante a busca de um estado desejado e pode ser aplicada [BAR 86]:

- a) para decidir qual nó (do grafo representado o espaço de estados) expandir, ao invés de fazer a expansão em uma ordem estritamente de abrangência ou profundidade;
- b) durante a expansão de um nó, para decidir qual(is) sucessor(es) gerar, ao invés de gerar indistintamente todos os sucessores possíveis, em um certo instante;
- c) decidir que certos nós podem ser descartados (cortados) da pesquisa naquele grafo.

O leitor interessado em métodos de busca pode consultar [BAR 86] ou [RIC 93].

A busca em espaço de estados só deve ser utilizada quando não há conhecimento de um método mais direto de solução do problema.

## 6.2 Representação Direta ou Analógica

Representação analógica é uma classe de esquemas de representação de conhecimento nas quais “propriedades e relações entre as partes da configuração representante reproduzem propriedades e relações entre as partes da configuração complexa sendo representada, de tal forma que a estrutura da representação fornece informação sobre a estrutura do que está sendo reproduzida” [Sloman, 1971 Apud BAR 86, p.200].

Portanto é necessário existir uma correspondência entre as relações nas estruturas de dados e as relações na situação sendo por elas representada. Essa similaridade estrutural é denotada *homomorfismo*. Observe que é a combinação das estruturas de dados e dos procedimentos utilizados para interpretá-las (*função de interpretação semântica*) que devem ser referidos como analógico (direto) e, mesmo assim, com respeito a certas propriedades. Por exemplo, um mapa e uma de interpretação adequada é *direto* com respeito a localização e distância dos objetos nele representados, mas normalmente não o é com relação a altitude.

Alguns exemplos dos primeiros sistemas inteligentes que utilizavam representação direta são: o Provedor de Teoremas Geométricos desenvolvido por Gelernter (em 1963); o

Planejador de Espaço Geral, desenvolvido por Eastman (em 1970); e o WHISPER, desenvolvido por Funt (em 1966).

O Provedor de Teoremas Geométricos é aplicado a prova de teoremas simples, no campo da geometria Euclidiana apresentada no curso de 2º grau. Ele utilizava um diagrama bidirecional para representar os teoremas. Esse diagrama era utilizado para guiar a prova: assim qualquer hipótese que não fosse verdadeira no diagrama era rejeitada e considerada falsa. O programa utilizava técnica de redução de problema e o sistema trabalhava do teorema a ser provado para trás (“*backward*”), realizando uma busca em um espaço proposicional (i.e. composto de um conjunto de declarações formais, com notação lógica), usando a representação analógica (o diagrama) como um guia para eliminar algumas sub-provas. Esse procedimento conseguia eliminar cerca de 99,5% das sub-provas possíveis em cada nível da busca.

O sistema de Eastman alocava objetos em um espaço dado, levando em conta as restrições a respeitar. Essas restrições, do tipo *deve-ser-adjacente-a*, *deve-ser-visível-de*, etc., eram estruturadas em um grafo de restrições relativo ao espaço disponível. A estrutura de representação do domínio era um vetor bidimensional, de forma que as propriedades de tamanho, forma e posição dos objetos fossem representadas em uma forma *direta*. Além disso o espaço dado era particionado em um conjunto de retângulos, o que permitia fácil verificação do espaço vazio e da superposição de objetos.

Esse sistema utilizava busca em profundidade, localizando posições viáveis para objetos sucessivos e retornando um nível quando alguma restrição fosse violada. Começando com o grafo mais restritivo, a busca era relativamente eficiente [BAR 86]. Observe que Eastman trabalhou em uma forma contrária a abordada por Gelernter, pois seu sistema executava busca em um espaço analógico (o vetor bidimensional) usando uma forma proposicional (o grafo de restrições) como um guia heurístico.

WHISPER era um sistema para raciocinar exclusivamente com representação analógica e operava em um mundo de blocos bidimensional resolvendo problemas devido a interação entre blocos. Nesse caso eram utilizadas duas formas de analogias: a) entre os estados estáticos da representação bidimensional do mundo (posição dos blocos, tamanho, orientação, etc.) e o mundo em si; e b) comportamento dinâmico dos blocos representados e o comportamento dos objetos no mundo real (considerados corpos rígidos interagindo entre si sob o efeito da gravidade). Para as analogias comportamentais, utilizava um “raciocinador” de alto nível, correspondente a procedimentos dependentes do domínio, contendo conhecimento físico qualitativo sobre choque entre corpos.

### 6.2.1 Vantagens e Desvantagens da Representação analógica

Devido ao homomorfismo, representações analógicas permitem *observação* de certos fatos, enquanto sistemas declarativos requerem que os mesmos sejam *deduzidos*. Em situações especiais (como ilustrado pelo sistema de Gelernter), o uso de representação analógica pode facilitar o procedimento de busca. Como desvantagens podemos citar: alguns aspectos da analogia podem não se manter na situação real e nós podemos não saber quais são eles (i.e. quais são os limites da representação); pode se tornar difícil lidar com representações analógicas quando temos certos tipos de informação incompleta. Por exemplo, suponha que a localização de uma cidade é conhecida de forma

indiretamente, como eqüidistante entre as cidades X e Y. A distância com relação as outras cidades deverá ser representada através de equações e o poder da analogia é perdido [BAR 86].

## Parte II Representação de Conhecimento Incerto

Nesta parte são discutidas as principais abordagens simbólicas (lógicas não-monotônica, sistemas de manutenção da verdade e teoria do endosso de Cohen) e numéricas (teoria de probabilidades, regras com fatores de certeza associados, redes bayesianas, teoria da evidência de Dempster-Shafer e as técnicas baseadas na teoria dos conjuntos fuzzy: lógica fuzzy, teoria das possibilidades de Zadeh e normas triangulares) para representar conhecimento incerto e raciocinar com base nele.

Freqüentemente as pessoas resolvem problemas e tomam decisões em ambientes onde a informação é parcial ou aproximada. Os pesquisadores de IA têm tentado emular essa capacidade em sistemas inteligentes. Com informação parcial ou aproximada, o problema somente pode ser resolvido aproximadamente, i.é com incerteza. Um solução sem incerteza só é possível com informação completa e exata. As abordagens utilizadas para lidar com incerteza consideram contexto simbólico ou contexto numérico [BHA 86].

A abordagem simbólica é voltada para o tratamento da incerteza em informação incompleta, sendo geralmente inadequada para o caso de informação imprecisa pois não possui mecanismos para quantificar níveis de confiança. Também é mais adequada do que a abordagem numérica para permitir a reconstituição dos caminhos do raciocínio a partir das fontes de informação até as conclusões finais. Essa abordagem utiliza enfoque *formal* (possui uma teoria lógica que determina o mecanismo utilizado na avaliação das inferências) ou enfoque *heurístico* (possui um conjunto de regras dependente de contexto para manipular estruturas de endosso) [BON 90].

Lógicas não-monotônica e sistemas de manutenção da verdade são exemplos do enfoque formal. Nessas técnicas a informação incompleta é representada como uma disjunção de diversas possibilidades, as quais são consideradas caso a caso pelo procedimento de provas. O sistema não pode coexistir com informação inconsistente, devendo a inconsistência causada pela adição de nova informação ser resolvida imediatamente: se evidências apontam para mais de um valor possível para uma variável, algum método deve ser utilizado para selecionar um valor apenas para ela. Se não existir nenhuma informação sobre o valor de uma variável, um valor deve ser adotado. Esse formalismo representa um modelo de um mundo com incerteza somente se utilizar predicados com interpretação incerta. O valor atribuído a uma variável é considerado ser uma *crença*, a qual pode ser revisada se nova informação estiver disponível.

Teoria do endosso de Cohen constitui o representante do enfoque simbólico heurístico. O *endosso* representa conhecimento sobre situações de incerteza, incluindo, mas sem se limitar a razões para acreditar e desacreditar em proposições incertas [SUL 85]. Neste contexto raciocinar consiste em discriminar esses fatores.

A abordagem numérica geralmente representa a incerteza como uma quantidade precisa (único valor ou um intervalo) em uma dada escala e requer que o usuário ou o especialista faça uma atribuição numérica precisa e consistente para a incerteza dos dados atômicos e das suas relações [BON 87]. Um terceiro enfoque é o de representar incerteza através de um valor fuzzy. Teoria de probabilidades (estatística bayesiana), fatores de

certeza e redes bayesianas são exemplos do enfoque monovalorado; teoria da evidência de Dempster-Shafer exemplifica o enfoque de intervalos; e lógica fuzzy, a teoria das possibilidades de Zadeh e normas triangulares exemplificam formalismos baseados na teoria dos conjuntos fuzzy.

A abordagem numérica permite definir um cálculo que especifica o mecanismo a ser utilizado para combinar e propagar a incerteza durante o processo de raciocínio. O uso de operadores de agregação de incerteza permite obter uma classificação dos fatos resultantes com incerteza resumida, de forma a permitir o seu uso na tomada de decisão. Entretanto essa abordagem não provê meios para explicar claramente as razões que conduziram a uma dada conclusão [BON 90].

Pearl [PEA 88] apresenta as seguintes escolas para lidar com incerteza: lógicos<sup>9</sup>, neo-calculistas<sup>10</sup> e neo-probabilistas<sup>11</sup> e uma com abordagem heurística informal (teoria do endosso de Cohen) na qual não é dada uma notação explícita para incertezas, sendo elas embutidas em procedimentos específico, dependentes do domínio e em estruturas de dados. Essa classificação considera mais aspectos sintáticos do que semânticos. Pearl também considera a taxonomia de extensionais vs. intensionais (sintáticos vs. semânticos).

A abordagem *extensional* trata incerteza como um valor verdade generalizado associado às fórmulas (derivadas da lógica clássica) e calcula a incerteza de qualquer fórmula como uma função de incerteza das suas sub-fórmulas. Essa abordagem é computacionalmente eficiente, mas sem semântica clara, podendo produzir resultados inesperados e não intuitivos. De forma geral a escola neo-calculista é extensional, sendo o cálculo dos fatores de certeza do MYCIN um representante típico. Os conectivos de uma fórmula MYCIN servem para selecionar a função de combinação de pesos adequada para as sub-fórmulas.

A abordagem mais difundida hoje em dia é a *intensional* (também conhecida como declarativa ou baseada em modelo) na qual incerteza é associada a “estados do assunto” ou subconjuntos de “mundos possíveis”, com significado semântico claro. A sintaxe consiste de declarações sobre estados do assunto que refletem o conhecimento corrente sobre o mundo. De forma geral a escola neo-probabilística é intensional, sendo a teoria de probabilidades um representante típico. Na teoria de probabilidades as medidas de certeza são associadas a conjuntos de mundos e os conectivos combinam conjuntos de mundos através das operações da teoria dos conjuntos. Assim, a probabilidade  $P(A \wedge B)$  é dada pelo peso atribuído a interseção de dois conjuntos de mundos, aquele no qual A é verdadeiro e aquele no qual B é verdadeiro, não podendo ser determinada a partir das probabilidades individuais  $P(A)$  e  $P(B)$ .

As regras têm papéis diferentes nestes sistemas. A regra fornece uma “licença” para certas atividades simbólicas no sistema extensional. Assim a regra  $A \rightarrow B(m)$  pode significar “se você observa A, então tem a licença para atualizar a certeza de B por uma certa quantidade que é função da certeza m associada a regra”. A regra é interpretada como

<sup>9</sup> lidam com incertezas utilizando técnicas não numéricas, principalmente lógicas não-monotônicas.

<sup>10</sup> usam representações numéricas. Consideram o cálculo de probabilidades inadequado e desenvolveram novos cálculos como fatores de certeza, teoria da evidência de Dempster-Shafer e lógica fuzzy.

<sup>11</sup> utilizam a teoria de probabilidades (Bayes), mas tentam reforça-la com facilidades computacionais necessárias para executar tarefas típicas de programas de IA.



um sumário da experiência passada, descrevendo a maneira como um agente normalmente reage a situações do problema ou a evidências. Já no sistema intensional, as regras denotam restrições (que podem ser relaxadas) sobre o mundo. No formalismo Bayesiano a regra  $A \rightarrow B$  ( $m$ ) é interpretada como a probabilidade condicional  $P(B|A)=m$ , estabelecendo que entre todos os mundos satisfazendo  $A$ , aqueles que também satisfazem  $B$ , constituem uma fração de valor  $m$ .

A abordagem extensional apresenta os seguintes problemas principais: a) manipulação inadequada de inferências bidirecionais (ocorrência de  $A$  e de  $B$ , onde  $A$  é evidência para  $B$  e vice versa); b) dificuldades em explicar as conclusões obtidas; e c) tratamento impróprio de fontes de evidências correlacionadas. A intensional não tem problemas com manipulação de inferências bidirecionais ou evidências correlacionadas, o que decorre da existência de um modelo global coerente. As declarações sintáticas não estão associadas a procedimentos (como no caso da extensional). Como torna-se necessário construir mecanismos especiais que convertam essas declarações em rotinas que possam ser utilizadas para responder as questões (fazendo as inferências que forem necessárias), essa abordagem é computacionalmente mais ineficiente, embora seja semanticamente clara.

A lógica fuzzy permite inferências sobre dados fuzzy, i.é aqueles aos quais está associado um grau de pertinência gradual (de 0 a 1) em relação a um conjunto (com fronteiras mal definidas). Pertinência zero significa que o elemento não pertence ao conjunto fuzzy e um significa que pertence. Um valor intermediário indica uma pertinência incerteza.

Na teoria de possibilidades a informação sobre o verdadeiro valor de uma variável é codificado sobre a forma de uma distribuição de possibilidades, a qual pode ser vista como a função de pertinência a um conjunto fuzzy da variável em questão. A possibilidade de  $X$  assumir valor  $x$  reflete até que ponto é possível que o valor de  $X$  seja  $x$ .

Zadeh [ZAD 78] e Bhatnagar e Kanal [BHA 86] consideram as técnicas numéricas baseadas na teoria dos conjuntos fuzzy adequadas para representar conceitos vagos inerentes a alguns termos lingüísticos, tais como: a) imprecisão lingüística de predicados como alto, médio, baixo, maior, menor, próximo, longe, jovem, velho, etc., b) imprecisão do tipo ‘a caixa pesa entre 10 e 16 quilos’ e c) quantificadores imprecisos (fuzzy) do tipo ‘muitos’, ‘alguns’, ‘poucos’, ‘a maioria’, etc. Já Dubois e Prade [DUB 88] consideram a lógica fuzzy uma lógica de predicados vagos e a lógica de possibilidades uma lógica de ignorância parcial, que permite representar estados de conhecimento variando de informação completa a ignorância total. Lógica de possibilidades é uma lógica de incerteza que está voltada para o raciocínio incerto fuzzy (i.é raciocínio preciso em uma situação descrita incompletamente em uma base de conhecimentos fuzzy). Lógica fuzzy está voltada para o raciocínio fuzzy com informação precisa (i.é raciocínio com predicados vagos em uma base de conhecimentos precisa).

Bonissone [BON 86, 87 e 90] considera que a abordagem simbólica é inadequada para representar e resumir medidas de incerteza e que a abordagem numérica impõe restrições sobre a estrutura da informação (p.ex. independência condicional das evidências ou exclusão mutua de hipóteses) e exige que o usuário ou o especialista forneça uma estimativa precisa e consistente da incerteza dos dados atômicos e de suas relações, o que é irreal. Assim propõe caracterizar incerteza com probabilidades lingüísticas e semântica fuzzy e o uso de normas triangulares para combinar e resumir medidas de incerteza.

## 7 Técnicas Baseadas no Raciocínio Não-Monotônico

Neste capítulo são descritos os formalismos de lógica não-monotônica, lógica *default*, raciocínio minimalista e sistemas de manutenção da verdade, para tratamento de informação incompleta (com a abordagem simbólica para representação de incerteza). O conhecimento é representado com base em lógica ou regras de produção.

Raciocínio não-monotônico lida com inferências não-monotônicas (i.e. dependam da falta de algum conhecimento) através da associação de crenças às declarações. Acredita-se em uma declaração ou não acredita-se, não sendo permitido outros graus de crença. Uma declaração é considerada verdadeira, falsa ou nem verdadeira nem falsa. Utiliza arcabouço lógico das regras de produção ou lógica de predicados e estende axiomas e/ou regras de inferência para permitir raciocínio com *informações incompletas*.

Raciocina com asserções (*declarações default*) válidas em geral, como ‘tipicamente aves voam’, mas que admitem exceções para casos particulares, como pinguins, avestruzes, etc. Na *ausência* de informação contrária, assume-se uma posição geral que poderá ser revista mais tarde, ao se dispor de novas informações que a invalidam.

Após a representação de uma situação, são atribuídas crenças as asserções. Havendo inconsistência, é necessário determinar as crenças inconsistentes, analisar as suas evidências de suporte e rever a que possui menor sustentação. Nessa análise as crenças inconsistentes são separadas e várias outras crenças recebem status subordinado de evidência subjacente. Essa forma permite enfocar a atenção nas crenças proeminentes, entre as quais uma será eliminada e classificar as outras crenças, embaixo destas, como meras auxiliares no processo de eliminação de uma das proeminentes. Esse processo traz à baila várias crenças ocultas e a necessidade de mais investigação para buscar mais evidências para elas. A investigação pára quando se está satisfeito sobre qual a melhor maneira de restaurar a consistência (i.e. determinar qual das crenças examinadas será descartada).

Essa abordagem suscita as seguintes questões-chave, segundo [RIC 93]:

- a) como a base de conhecimentos pode ser ampliada para permitir inferências com base na falta de conhecimento (não-monotônicas) tanto quanto na presença dele?
- b) como a base de conhecimentos pode ser atualizada quando um novo fato é acrescentado ao sistema (ou quando um fato antigo é removido)?
- c) como é que o conhecimento pode ser usado para ajudar a resolver conflitos quando várias inferências não-monotônicas inconsistentes podem ocorrer?

Para os autores citados, ainda não surgiu um formalismo único com todas as propriedades desejadas. São considerados raciocínios não-monotônicos o raciocínio *default*, o raciocínio minimalista, a abdução e a herança. Apresentaremos os seguintes formalismos que utilizam raciocínio não-monotônico: lógica não-monotônica, lógica *default*, raciocínio minimalista, retrocesso dirigido por dependência e sistema de manutenção da verdade baseado em justificativas.

Raciocínio não-monotônico é utilizado em IA na implementação de herança de propriedades em redes semânticas e *frames*; na afirmação da não negação de fatos em uma base

de conhecimentos, baseado na hipótese do mundo fechado; durante o uso do raciocínio abdutivo em sistemas de diagnóstico; em sistemas com revisão de crenças, etc.

## 7.1 Lógica Não-Monotônica

Com a introdução do operador modal  $M$  (lido como “talvez” ou “é consistente com tudo que é conhecido”) estende a lógica de predicados, visando tratar declarações *default* através de fórmulas. Foi proposta por McDermott e Doyle, em 1980 [McD 80].

Aplica-se o operador  $M$  a uma asserção quando se acredita que a asserção é consistente com todas as outras crenças existentes. O problema passa a ser definir ‘é consistente’, mas na lógica de predicados e na lógica não-monotônica (LNM) a consistência é indecidível. O procedimento de resolução não dá garantia de parar quando o que tiver de ser demonstrado (consistente com os outros fatos conhecidos) não for um teorema. É preciso usar uma aproximação. A aproximação normalmente usada é a noção de negação por falha. Assim, para mostrar que  $B$  é consistente, tenta-se provar  $\neg B$ . Se falhar, assume-se que  $\neg B$  é falsa e que  $B$  é consistente. Por questões práticas (esforço computacional), pode ser necessário definir consistência em bases heurísticas, como por exemplo associar a falha em provar inconsistência com o desenvolvimento de um nível fixo de esforço.

A LNM define o conjunto de teoremas que podem ser derivados de um conjunto  $A$  de fbf não-monotônicas (fecho semântico) como sendo a *interseção* dos seus pontos fixos (conjuntos de teoremas que resultam das várias maneiras de combinação das fbf de  $A$ ) [REI 87] [RIC 93].  $T$  é um *ponto fixo* de  $A$  se  $T = \Gamma(A \cup \{Mw \mid \neg w \notin T\})$ , onde  $\Gamma$  denota fechamento (em relação a derivabilidade na lógica de primeira ordem). Essa expressão intuitivamente nos autoriza a considerar  $w$  consistente, caso  $\neg w$  não seja derivável.

Por exemplo, retomando a discussão sobre se Toni é ou não pacifista, iniciada no seção 4.2, admita o seguinte conjunto de asserções:

$$\forall x (Xenófobo(x) \wedge M\neg Pacifista(x) \rightarrow \neg Pacifista(x)) \quad (1)$$

(leia: se é xenófobo e o fato de não ser pacifista é consistente com tudo mais em que se acredita, então conclua que é não pacifista)

$$\forall x (Ecologista(x) \wedge MPacifista(x) \rightarrow Pacifista(x)) \quad (2)$$

$$Xenófobo(Toni) \quad (3)$$

$$Ecologista(Toni) \quad (4)$$

Aplicando (3) em (1), conclui-se que Toni não é pacifista. Aplicando-se (4) em (2), conclui-se que Toni é pacifista. A interseção desses dois teoremas é vazio, logo não se pode derivar nenhuma conclusão sobre o pacifismo de Toni.

O significado do operador  $M$  *depende do contexto* (teoria formada pelas asserções da base). Por exemplo,  $M\neg Pacifista(Toni)$  significa que ‘Toni não ser pacifista’ é consistente com a teoria formada pelas asserções 1,2 e 3. Se posteriormente descobrimos que Toni é ecologista e o incluirmos na base de conhecimentos, então  $M\neg Pacifista(Toni)$  passa a ser consistente apenas com as asserções 1,2 e 4 (Toni não pode ser xenófobo!).

Os principais problemas da LNM são os seguintes, segundo [BIT 96] e [REI 87]:

- a) não validade das inferências: inferir  $\neg A$  se  $A$  não é consistente e inferir que  $A$  é consistente se  $A \wedge B$  é consistente;
- b) o fato do conjunto  $\{MA, \neg A\}$  ser consistente, o que é contrário a intuição;
- c) os teoremas (fecho semântico) da base não são enumeráveis recursivamente.

Algumas conseqüências indesejáveis da LNM foram contornadas por Moore em sua lógica autoespistêmica [BIT 96]

## 7.2 Lógica Default

É uma extensão da lógica de predicados, baseada no conceito de *consistência* com o que é conhecido (as asserções da base de conhecimentos), proposta por Raymond Reiter [REI 80] que trata declarações *default* através da inclusão da nova classe de regras denominadas regras *default* ou “quasi-regras” de inferência com o formato:

$$\frac{A : B_1, \dots, B_n}{C}$$

$$\frac{A : B}{B}$$

$$\frac{A : C \wedge B}{C}$$

onde a primeira pode ser lida como: se  $A$  (*pré-requisito*) ocorre e se assumir  $B_i$  (*justificativas*) é *consistente* com o que é conhecido, então conclua  $C$  (*conseqüente da exceção*).  $B_i$  é consistente se não existir uma prova para  $\neg B_i$ . As duas últimas formas de exceções são chamadas, respectivamente, *normais* e *semi-normais*.

Essas regras *default* são usadas como base para a obtenção de um conjunto de *extensões* plausíveis à base de conhecimentos. Cada *extensão* corresponde a uma *expansão máxima e consistente* da base de conhecimentos, definida como uma nova teoria lógica que contém a base de conhecimentos, tal que: a) nenhuma das regra *default* pode consistentemente ser aplicada para obter uma conclusão que não esteja na extensão (expansão máxima), b) a extensão é minimal sujeita a condição anterior (i.é apenas expressões geradas pela aplicação das regras *default* à base de conhecimentos são permitidas).

Do ponto de vista da representação de conhecimento, começamos com uma base lógica de primeira ordem, a qual representa todas as coisas que sabemos ser verdadeiras sobre o domínio. Por normalmente ser esse conhecimento incompleto é necessário estender essa base, introduzindo regras *default* para representar *conclusões plausíveis* que podem ser revistas mais tarde, na presença de informações contrárias. Uma *teoria de exceções* é definida como um par  $(D, W)$ , onde  $D$  é um conjunto de regras *default* e  $W$  é um conjunto de sentenças da lógica de primeira ordem. O conjunto de sentenças lógicas  $E$  é uma *extensão* da teoria de exceções  $(D, W)$  sse  $E$  é um ponto fixo do operador  $\Gamma$ , ou seja  $\Gamma(E) = E$ . Para um conjunto  $S$  qualquer de fórmulas da lógica de primeira ordem,  $\Gamma(S)$  é definido como o *menor* conjunto de fórmulas da lógica de primeira ordem, satisfazendo as três propriedades seguintes:

1.  $W \subseteq \Gamma(S)$ ,
2.  $\Gamma(S)$  é fechado sobre o operador de derivabilidade da lógica de primeira ordem (i.é se  $H$  é derivável a partir de  $\Gamma(S)$ , então  $H \in \Gamma(S)$ ; inclui as implicações lógicas),
3. Se  $(A : B_1, \dots, B_n)/C$  for uma regra *default* de  $D$ ,  $A \in \Gamma(S)$  e  $\neg B_i \notin \Gamma(S)$ , então  $C \in \Gamma(S)$ .

Intuitivamente uma extensão corresponde ao conjunto de crenças justificadas pelo que é conhecido sobre o mundo. Se for necessário decidir entre *extensões*, a lógica *default* não

provê nenhum mecanismo especial, sendo necessário recorrer a algum outro mecanismo específico. Para Reiter [REI 87], cada extensão é um conjunto de crença possível para uma agente. Também é possível considerar, a exemplo de McDermott e Doyle, que as crenças de um agente são formadas pela *interseção* de todas as extensões possíveis.

Retomando a discussão sobre se Toni é ou não pacifista, as únicas verdades sobre o domínio que conhecemos são:  $W = \{Xenófobo(Toni), Ecologista(Toni)\}$ . Seja as seguintes regras *default*  $D = \{ (Xenófobo(x) : \neg Pacifista(x)) / \neg Pacifista(x), (Ecologista(x) : Pacifista(x)) / Pacifista(x) \}$ . Então a teoria *default* disponível é dada por  $(D, W)$ .

A primeira regra *default* pode ser utilizada visto que seu *pré-requisito* ( $Xenófobo(Toni)$ ) é verdadeiro e que sua *justificativa* ( $\neg Pacifista(Toni)$ ) é consistente com tudo que se conhece (i.é  $W$ ). Podemos concluir  $\neg Pacifista(Toni)$ . Essa conclusão impede a ativação da segunda regra *default*, pois é inconsistente com a *justificativa*  $Pacifista(Toni)$ . Assim obtemos uma extensão da teoria  $(D, W)$  acima, dada por  $\{W \cup \neg Pacifista(Toni)\}$ .

Analogamente, podemos obter outra *extensão* desta teoria de exceções, dada por  $\{W \cup Pacifista(Toni)\}$ , a partir do fato de que Toni é ecologista e aplicando a segunda regra *default*.

Como a lógica *default* não provê nenhum mecanismo especial de discriminação entre extensões de uma teoria, nada podemos concluir sobre o estado de beligerância de Toni.

Os principais problemas da LD são, segundo [BIT 96], [REI 87] e [PEA 91]:

- a) as extensões de uma teoria *default* não são enumeráveis recursivamente.
- b) não há mecanismo previsto para discriminação entre múltiplas extensões.

Reiter e Criscuolo [Reiter and Criscuolo, 1981 Apud PEA 91] propuseram o uso de regras *default* semi-normais da forma  $(A : B \wedge C) / C$ , onde  $B$  é utilizado para bloquear a ativação da regra (p.ex.  $(Ave(x) : \neg Pingüim(x) \wedge Voa(x)) / Voa(x)$ , significando aves voam, a menos que sejam pingüins) mas que apresenta o inconveniente de exigir a enumeração de todas as exceções dentro de regras *default*.

O conceito de distância inferencial de Touretzky [TOU 84], visto na discussão sobre herança múltipla em redes semânticas e *frames* (seção 4.2), pode ser utilizado como uma abordagem natural para priorizar regras *default* de subclasses em relação as regras análogas associadas às superclasses.

- c) as expressões não-monotônicas são regras de inferência, o que impede a sua manipulação como expressões (deduzir novas regras *default* a partir das antigas) e as vezes produz resultados inesperados.

Assim, a partir das regras “Normalmente canários são amarelos.” e “Coisas amarelas nunca são verdes.” não se pode concluir que “Normalmente canários nunca são verdes.”. Na LNM se pode obter a seguinte expressão  $\forall x (Canário(x) \wedge \neg Amarelo(x) \rightarrow \neg Verde(x))$  embora haja alguma discussão sobre se ela pode legitimamente representar o sentido pretendido acima.

Dadas as duas regras *default*:  $(A : B) / B$  e  $(\neg A : B) / B$  e nenhuma afirmação sobre o pré-requisito  $A$ , nada poderá ser concluído sobre  $B$ , já que nenhuma das regras se aplica.

- d) não permite o raciocínio por caso. Suponha que Tweety, representado por  $b$ , é um pássaro de gênero desconhecido e que saibamos que pássaros( $b$ ) machos( $m$ ) tipicamente voam( $v$ ) e que pássaros fêmeas ( $\neg m$ ) tipicamente voam, o que dá origem a teoria *default*  $W = \{b\}$ ,  $D = \{b \wedge m : v, b \wedge \neg m : v\}$ . Como nenhum dos pré-requisitos dessas regras são satisfeitos, nenhuma delas pode ser ativada; logo a conclusão natural de que Tweety voa ( $b \rightarrow v$ ) não pode ser inferida.
- e) não há consenso sobre a semântica da LD.

Pearl [PEA 91] propôs uma semântica probabilística para o raciocínio *default*, com o uso de probabilidades condicionais, baseada na interpretação probabilística de uma regra *default* “tipicamente aves voam” como  $\forall x (P[Voa(x) \mid Ave(x)] = ALTA)$ , mas essa abordagem recebeu diversas críticas de pesquisadores da escola dos lógicos.

### 7.3 Aplicações do Raciocínio Não-Monotônico

**Abdução.** Forma de raciocínio muito utilizada em sistemas de diagnóstico médico, consiste em, dadas duas fbf,  $A \rightarrow B$  e  $B$ , assumir  $A$ , se for consistente com o que se conhece até o momento. Concluir  $A$  pode estar errado, mas representa o melhor palpite que se pode ter sobre o que está acontecendo. É particularmente útil no caso de ser associada alguma medida de certeza às expressões resultantes da sua aplicação (Capítulo ?????). Essas expressões quantificam o risco do raciocínio abdutivo estar errado. Este risco será maior quanto maior for o número de outros antecedentes (além de  $A$ ) que possam produzir  $B$ . O raciocínio abdutivo pode ser descrito tanto na LNM quanto na LD.

**Herança.** A lógica *default* pode ser utilizada para representar herança de atributos de uma subclasse ou elemento da classe, em uma hierarquia do tipo é-um ou instância (Capítulo 4). Por exemplo, a herança do meio-de-locomção voar, da superclasse ave por uma subclasse qualquer, pode ser representada através da regra *default*:

$$\frac{Ave(x) : \neg Pingüim(x) \wedge Meio-de-locomção(x, voar)}{Meio-de-locomção(x, voar)}$$

A quebra de herança, para a subclasse Pingüim (anda ao invés de voar), pode ser representada pela regra de inferência:  $Pingüim(x) \rightarrow Meio-de-locomção(x, andar)$ . Como a ordem em que as regras e declarações ocorrem não tem importância para a LD (pois uma regra só será disparada se seu pré-requisito ocorrer e sua justificativa for consistente); simplesmente admite-se que um valor explícito bloqueará a herança de um valor *default*.

Como a regra *default* acima é do tipo *semi-normal*, apresenta o inconveniente de exigir a enumeração de todas as exceções, o que pode se tornar muito confuso. Uma outra abordagem é considerar todo o conjunto de exceções como uma anormalidade (i.e não é típico da classe) e associar as diversas subclasses a informação de que constitui uma das anormais (através do predicado  $Ab$ ). Então podemos ter uma única regra *default*, a qual dispara se o indivíduo for normal. O exemplo abaixo ajuda a esclarecer essa abordagem:

$$\frac{Ave(x) : \neg Ab(x) \wedge Meio-de-locomção(x, voar)}{Meio-de-locomção(x, voar)}$$

$$Pingüim(x) \rightarrow Ab(x)$$

$$Avestruz(x) \rightarrow Ab(x)$$

$$Ema(x) \rightarrow Ab(x)$$

**Raciocínio Minimalista.** Esse tipo de raciocínio está associado a idéia de que “há menos declarações verdadeiras do que falsas. Se algo é *verdadeiro e relevante*, faz sentido assumir que foi inserido na base de conhecimentos ou pode ser derivado a partir dela. Portanto assumamos que as únicas declarações verdadeiras são aquelas que precisam ser verdadeiras para que seja mantida a consistência da base de conhecimentos.” [RIC 93].

O conceito de negação por falha do Prolog é baseado nessa idéia (a negação de uma declaração é representada implicitamente pela sua ausência na base de conhecimentos), a qual juntamente com o uso de cláusulas de Horn permitiu a automatização de um procedimento *decidível* (a própria máquina de inferência do Prolog!). Veremos quais as implicações do ponto de vista lógico ao utilizarmos raciocínio minimalista, sem as restrições decorrentes de cláusulas de Horn.

Como exemplos de raciocínio minimalista serão apresentados a Hipótese do Mundo Fechado e a Circunscrição.

**Hipótese do Mundo Fechado.** A HMF, proposta por Reiter [Reiter 1978b Apud REI 87], aparece mais na teoria de banco de dados e estabelece que toda a informação positiva relevante foi explicitamente representada. Se um fato positivo não está explicitamente presente no banco de dados, assumimos a ocorrência da sua negação. No caso de banco de dados dedutivos, um fato não estar explicitamente presente não é suficiente para conjecturarmos sobre a sua negação, pois ele poderá ser dedutível a partir dos fatos conhecidos. Assim a HMF toma a seguinte forma: todas as declarações relevantes e positivas estão contidas na base de conhecimento ou podem ser derivadas a partir das declarações lá contidas. Declaração positiva que não esteja presente, pode ser assumida falsa.

Reiter define *fechamento* (do inglês, *closure*) de uma teoria de primeira ordem (DB) como:  $CLOSURE(DB) = DB \cup \{\neg P(t) \mid DB \not\models P(t)\}$ , onde  $P$  é um predicado  $n$ -ário de DB e  $t$  é uma  $n$ -tupla de termos atômicos formados usando os símbolos de DB. A informação negativa implícita, decorrente da adoção da HMF, é o conjunto de símbolos atômicos negativos cujos símbolos atômicos positivos não podem ser inferidos da base (i.e. DB não satisfaz a  $P(t)$ ). As pesquisas são avaliadas com respeito a  $CLOSURE(DB)$  e não com respeito a base de conhecimentos propriamente dita.

A HMF é poderosa como base para raciocínio em bases de conhecimentos que são supostas *completas* com relação ao domínio que descrevem, mas podem causar muitos problemas se as bases de conhecimentos estiverem incompletas.

Raciocínio minimalista baseado na HMF pode produzir uma resposta não adequada por dois motivos:

- algumas partes do mundo não são “fecháveis”, pois apresentam fatos relevantes para o raciocínio que ainda não foram descobertos e portanto não podem estar presentes na base de conhecimentos;
- o processo de raciocínio baseado na HMF é puramente sintático e seus resultados dependem da forma das declarações presentes na base de conhecimento. Seja a base de conhecimentos  $DB = \{\text{Solteiro}(\text{João}), \text{Solteiro}(\text{Maria})\}$ . Conforme mostra [RIC 93], ao indagarmos se Jane é solteira, a HMF produzirá a resposta  $\neg \text{Solteiro}(\text{Jane})$ , logo admite-se que Jane é *casada*. Ao invés de termos usado o predicado Solteiro, tivéssemos

usado Casado,  $DB = \{\neg \text{Casado}(\text{João}), \neg \text{Casado}(\text{Maria})\}$ , a indagação “Casado(Jane)?” produziria o resultado  $\neg \text{Casado}(\text{Jane})$ , logo admite-se que Jane é *solteira*!

A HMF apresenta as seguintes limitações essenciais:

- opera sobre predicados individuais, sem considerar as interações entre os predicados definidos na base de conhecimentos, o que pode implicar na obtenção de fechamentos de base de conhecimentos inconsistentes;
- assume que todos os predicados têm todas as suas instâncias listadas na base de conhecimentos. Alguns predicados podem ser considerados completamente definidos (pois a parte do mundo que eles descrevem está fechada) mas outros não (por que a parte do mundo que eles descrevem está aberta). Por exemplo, é razoável supor que o predicado *tem-camisa-verde(X)* deva ser considerado aberto, já que não se pode estar sempre seguro sobre o guarda-roupa das pessoas.

A não consideração das interações entre os predicados definidos na base de conhecimentos pode conduzir a inconsistências se a base não for composta de cláusulas de Horn. No caso de cláusulas de Horn, Reiter [Reiter 1978b Apud REI 87] mostrou que o fechamento preserva a consistência da base de conhecimentos.

Veremos o seguinte exemplo de obtenção de fechamento inconsistente. Seja a base de conhecimentos  $DB = \{P \vee Q\}$ . Já que  $P$  e  $Q$  não precisam ser necessariamente verdadeiros,  $CLOSURE(DB) = DB \cup \{\neg P, \neg Q\} = \{P \vee Q, \neg P, \neg Q\}$ , o que é inconsistente!

**Circunscrição.** Face as limitações da HMF, McCarthy propôs basear o raciocínio não-monotônico na noção de verdade em todos os modelos mínimos de uma teoria de primeira ordem. A idéia é acrescentar novos axiomas à base de conhecimentos (teoria) existente, para forçar uma interpretação mínima de uma parte dela selecionada. Cada axioma descreve uma maneira de circunscrever (delimitar, minimizar) o conjunto de valores para o qual um determinado axioma da teoria original é verdadeiro. O enfoque de modelos mínimos de primeira ordem, proposto por McCarthy [McC 80], foi generalizado através da adoção de uma nova teoria de circunscrição com um enfoque sintático [McC 86], a ser visto, também seguido por Lifschitz [Lifschitz 1985b Apud REI 87].

Seja  $L$  uma linguagem de primeira ordem e os seguintes símbolos de  $L$ :  $\mathbf{P}$  uma tupla de predicados livres,  $\mathbf{Z}$  uma tupla de predicados diferentes de  $\mathbf{P}$ , e  $A(\mathbf{P};\mathbf{Z})$  uma sentença que faz menção aos predicados de  $\mathbf{P}$  e  $\mathbf{Z}$ , podendo se referir a outros. A *circunscrição* de  $\mathbf{P}$  relativa a  $A(\mathbf{P};\mathbf{Z})$  e *variável*  $\mathbf{Z}$  é definida através da sentença de segunda ordem [REI 87]:

$$A(\mathbf{P};\mathbf{Z}) \ \& \ \forall(\mathbf{P}';\mathbf{Z}') \ (\neg(A(\mathbf{P}';\mathbf{Z}') \ \& \ \mathbf{P}' < \mathbf{P})). \quad (1)$$

A parte da direita desta conjunção é denotada *axioma de circunscrição* de  $A(\mathbf{P};\mathbf{Z})$  e estabelece que as extensões do predicado  $\mathbf{P}$  em  $A(\mathbf{P};\mathbf{Z})$  não podem ser menores, mesmo quando  $\mathbf{Z}$  varia. Assim a expressão acima representa a sentença  $A$ , restrita pela minimização de  $\mathbf{P}$  com  $\mathbf{Z}$  variando. As implicações não-monotônicas de interesse são as implicações (conseqüências lógicas) de (1). A relação de ordem entre os predicados  $Q$  e  $R$  (ou tupla de predicados), de mesma aridade é definida por:

$$Q \leq R \equiv \forall(x) (Q(x) \rightarrow R(x)),$$

$$Q < R \equiv Q \leq R \ \& \ \neg(R \leq Q),$$

$$(Q_1, \dots, Q_n) \leq (R_1, \dots, R_n) \equiv Q_1 \leq R_1 \ \& \ \dots \ \& \ Q_n \leq R_n \ \& \ \neg(R_1 \leq Q_1 \ \& \ \dots \ \& \ R_n \leq Q_n).$$



A expressão (1) é muito genérica. Para facilitar a identificação de quais predicados devemos associar a **P** e a **Z**, McCarthy [McC 86] sugeriu que fatos do senso comum possam ser representado de forma uniforme, em sentenças do tipo *A*, com o uso dos predicados unários  $Ab_i$  (do inglês, *Abnormal*), para capturar padrão “tipicamente, tal é o caso”. Eles devem ser minimizados durante a circunscrição de *A*. O exemplo abaixo, extraído de [REI 87] e [McC 86], ilustra esse caso:

1.  $\forall x (Coisa(x) \& \neg Ab_1(x) \rightarrow \neg Voa(x))$  (coisas normais não voam)
2.  $\forall x (Ave(x) \rightarrow Coisa(x) \& Ab_1(x))$  (aves são coisas anormais com relação a não voar)
3.  $\forall x (Ave(x) \& \neg Ab_2(x) \rightarrow Voa(x))$  (aves normais voam)
4.  $\forall x (Ema(x) \rightarrow Ave(x) \& \neg Voa(x))$  (emas são aves que não voam)

Então  $A(Ab; Voa)$  denota a conjunção das sentenças acima. Devemos minimizar  $Ab_1$  e  $Ab_2$ , com  $Voa$  variando, usando a equação (1), i.é dentre as entidades que satisfazem  $Ab_i$  e a  $Voa$ , permitir o mínimo de indivíduos possível, a saber apenas aqueles forçados pela base de conhecimentos a satisfazerem  $Ab_i$ . Como  $A(Ab; Voa)$  é verdade pois é parte do que assumimos em nossa base de conhecimentos, podemos simplificar (1) e considerar somente o axioma de circunscrição, para esse valor específico de *A*:

$$\forall (Ab'_1, Ab'_2; Voa') (\neg(A(Ab'_1, Ab'_2; Voa') \& Ab'_1 \leq Ab_1 \& Ab'_2 \leq Ab_2 \& \neg (Ab_1 \leq Ab'_1 \& Ab_2 \leq Ab'_2))).$$

Expandido a expressão acima, obtemos:

$$\begin{aligned} \forall (Ab'_1, Ab'_2; Voa') (\neg( \\ & (\forall x (Coisa(x) \& \neg Ab'_1(x) \rightarrow \neg Voa'(x)) \& (\forall x (Ave(x) \rightarrow Coisa(x) \& Ab'_1(x))) \\ & \& \\ & (\forall x (Ave(x) \& \neg Ab'_2(x) \rightarrow Voa'(x))) \& (\forall x (Ema(x) \rightarrow Ave(x) \& \neg Voa'(x))) \\ & \& \\ & Ab'_1 \leq Ab_1 \& Ab'_2 \leq Ab_2 \& \neg (Ab_1 \leq Ab'_1 \& Ab_2 \leq Ab'_2))) \end{aligned}$$

Como  $Ab'_1$ ,  $Ab'_2$  e  $Voa'$  são variáveis predicados quantificadas universalmente, podemos instanciá-las com quaisquer predicados de *A* que desejarmos, desde que as expressões resultem em verdade. O axioma  $A(Ab; Voa)$  determina  $Voa$ , i.é quem realmente voa, ou seja aves que não sejam emas. Portanto uma escolha adequada é dada por:

$$Voa'(x) \equiv Ave(x) \& \neg Ema(x), \quad Ab'_1(x) \equiv Ave(x) \text{ e } \quad Ab'_2(x) \equiv Ema(x).$$

Substituindo nas variáveis predicados e considerando  $A(Ab; Voa)$ , podemos inferir pela lógica de primeira ordem as expressões (não implicadas pela base não circunscrita):

$$\begin{aligned} \forall(x) (Ab_1(x) \equiv Ave(x)), & \quad \forall(x) (Ab_2(x) \equiv Ema(x)), \\ \forall(x) (Coisa(x) \& \neg Ave(x) \rightarrow \neg Voa(x)) \text{ e } & \quad \forall(x) (Ave(x) \& \neg Ema(x) \rightarrow Voa(x)). \end{aligned}$$

Reiter [REI 87] considera a teoria de circunscrição “o formalismo mais rico dentre todos os propostos para raciocínio não-monotônico e o mais ameno para a análise matemática. Como resultado, suas propriedades formais têm sido estudadas extensivamente”.

Os principais problemas apresentados pela circunscrição, na visão de Reiter, são:

- em geral não é óbvio que predicados escolher para instanciar as variáveis predicados. Lifschitz [Lifschitz 1985b Apud REI 87] apresenta alguns casos especiais.
- por utilizar lógica segunda ordem, as teorias de circunscrição possuem fórmulas válidas em geral não enumeráveis recursivamente.

- pode ocorrer que uma teoria satisfatível tenha uma circunscrição insatisfatível. Se a teoria original possui todas as suas sentenças na forma normal prenex, isso não ocorrerá, conforme Etherington [Etherington et al. 1985 Apud REI 87].

**Outras Abordagens.** Alguns pesquisadores abordam o raciocínio não-monotônico através de lógicas de crenças (ou de conhecimento). Esse enfoque é denotado epistêmico. Nesse contexto declarações *defaults* como “tipicamente, aves voam” são consideradas como “se  $x$  é uma ave e se você não acredita (conhece) que  $x$  não pode voar, então  $x$  pode voar”. A não-monotonicidade é obtida pela consideração de lógicas que utilizam um princípio similar à HMF: essas lógicas autorizam  $\neg B\alpha$  (denotando que um agente não acredita em  $\alpha$ ) quando  $\alpha$  não faz parte de um conjunto de crenças do agente [REI 87]. Assim, se o conjunto de crenças de um agente contém  $Ave(Tweety)$  e a regra *default*  $\forall x (Ave(x) \& \neg B\neg Voa(x) \rightarrow Voa(x))$ , mas não contém a crença  $\neg Voa(Tweety)$ , então pelo princípio acima  $\neg B\neg Voa(Tweety)$  será acrescentado ao seu conjunto de crenças. Aplicando *modus ponens*, tal conjunto de crença também conterá  $Voa(Tweety)$ .

O leitor interessado em lógicas autoepistêmicas poderá consultar [Moore 1984, 1985] ou [Konolige 1987] Apud REI 87; aqueles interessados na linguagem de representação para base de conhecimento que considera conhecimento próprio (conhecimento da falta de um conhecimento, embora o seu valor seja desconhecido) podem consultar [Levesque 1982, 1984 Apud REI 87], dentre outros.

Outro enfoque não-monotônico é o baseado em lógica condicional, em sentenças *default* do tipo “se  $x$  fosse um pássaro, então  $x$  poderia voar” [Delgrande 1986 Apud REI 87].

### 7.3.1 Sistemas de Manutenção da Verdade

Segundo Rich [RIC 93] os formalismos vistos para raciocínio não-monotônico não estão suficientemente desenvolvidos de forma a permitir o seu uso na resolução de problemas. Em geral, eles não são eficientes do ponto de vista computacional e nenhum deles é decidível, embora alguns sejam semi-decidível em suas formas proposicionais. Todos eles não conseguem lidar com os seguintes problemas que aparecem em sistemas reais:

- como derivar as conclusões não-monotônicas relevantes para solucionar o problema, sem perder tempo na derivação de conclusões lógicas desnecessárias;
- como atualizar incrementalmente o conhecimento durante o processo de solução. A abordagem de a cada alteração na base de conhecimentos, partir da base original e reaplicar etapas do raciocínio não-monotônico usado anteriormente, pode exigir um esforço computacional proibitivo;
- formalismos de raciocínio não-monotônico normalmente geram várias extensões em um dado momento, mesmo que muitas delas possam ser eliminadas com a disponibilidade de novos conhecimentos. Como escolher entre as diversas extensões?

Para resolver esses problemas, as técnicas de implementação normalmente dividem o processo de raciocínio não-monotônico em duas partes: um resolvidor de problemas e um sistema de manutenção da verdade. O *resolvidor de problemas* lida com as questões lógicas e pode utilizar alguma heurística em seu mecanismo para tirar as conclusões que se fizerem necessárias. O *sistema de manutenção da verdade* visa fazer as anotações ne-

cessárias, no decorrer do processo de solução, visando uma solução para o problema de como atualizar incrementalmente o conhecimento.

**Resolvedor de problemas.** Podem ser classificados de acordo com a abordagem utilizada para o controle da busca (no espaço de regras da base de conhecimentos) em: busca em profundidade ou busca em amplitude. A *busca em profundidade* consiste em seguir o percurso mais *provável*, até surgir novas informações e a busca terminar ou obrigar a desistir daquele caminho e encontrar outro. A *busca em amplitude* consiste em considerar que *todas* as possibilidades são igualmente prováveis e eliminar algumas delas quando novas informações estiverem disponíveis. Pode acontecer que apenas uma ou um pequeno número delas seja consistente com todas as informações disponíveis.

Outra classificação, baseada na direção em que o raciocínio não-monotônico é realizado, é: raciocínio para a frente ou para trás. *Raciocínio para frente* parte das evidências observadas e obtém conclusões. Esses sistemas estende as regras normais com cláusulas *default*. O controle (inclusive sobre qual interpretação *default* escolher) é tratado do mesmo modo que as outras decisões de controle do sistema (p.ex. baseado no uso de meta-regras, na ordem em que aparecem as regras, etc.).

O *raciocínio para trás* parte das hipóteses e busca as evidências que a suportam ou refutam, i.e. visa determinar se uma expressão P é verdadeira ou encontrar um grupo de instanciações para as suas variáveis que a torne verdadeira. Os sistemas desse tipo podem:

- permitir cláusulas *default* (do tipo EXCETO) nas regras e resolver os conflitos entre *defaults* usando a mesma estratégia de controle usada para outros tipos de raciocínio (em geral, a ordem em que as regras aparecem); e/ou
- dar suporte a um *sistema de debates* no qual há tentativa de construir argumentos tanto a favor de P quanto contra P e posteriormente aplicar conhecimentos adicionais para decidir qual lado tem argumentos mais fortes.

Um exemplo de uso de cláusulas *default* do tipo EXCETO, apresentado em [RIC 93, p.226] é o caso do “Assassinato ABC”, proposto por Quine e Ullian, em 1978. Os pontos (1) a (4) abaixo são inconsistentes, sendo necessário escolher um para rejeição:

“Alberto, Beto e Carlos são suspeitos em um assassinato. Alberto tem um álibi, no registro de um respeitável hotel em Manaus. Beto também tem um álibi, já que seu cunhado testemunhou que Beto estava visitando-o em Rio do Sul na época do crime. Carlos pleiteia um álibi, pois alega que estava em um torneio de vela em Guaratuba, mas só temos a sua palavra. Como Carlos está documentando o seu álibi (ele teve a sorte de ser pego pelas câmaras de televisão bem ao lado dos veleiros no torneio de vela) acreditamos que:

- 1) Alberto não cometeu o crime, pois há um registro no hotel e o hotel é antigo e goza de boa reputação;
- 2) Beto não cometeu o crime, pois seu cunhado testemunhou a seu favor, mas ele pode estar mentindo;
- 3) Alberto ou Beto ou Carlos o cometeu, pois não há sinais de roubo e apenas Alberto, Beto ou Carlos podiam ganhar com o crime em caso de ausência de roubo, mas pode haver um quarto beneficiário;
- 4) Carlos não cometeu o crime, tendo em vista a evidência da televisão.”

A Tabela 7.1 representa essa situação, com regras de produção e cláusulas EXCETO. Na notação da LD a codificação seria como:  $(\text{Beneficiário}(x) : \neg \text{Álibi}(x)) / \text{Suspeito}(x)$ . Se o resolvidor de problemas estiver usando controle tipo Prolog (i.e regras casadas de cima para baixo e da esquerda para a direita), ao fazermos a pergunta *Suspeito(x)*, tentará Alberto, Beto e Carlos; mas como todos eles têm um álibi, retornará contradição.

**Tabela 7.1** - Base de Conhecimento da História do Assassinato ABC

Suspeito(x)	$\leftarrow \text{Beneficiário}(x) \text{ EXCETO } \text{Álibi}(x)$
Álibi(x)	$\leftarrow \text{Outro-lugar}(x)$
Outro-lugar(x)	$\leftarrow \text{Registro-de-hotel}(x,y) \wedge \text{Distante}(y) \text{ EXCETO } \text{Registro-forjado}(x)$
Álibi(x)	$\leftarrow \text{Defende}(x,y) \text{ EXCETO } \text{Mente}(y)$
Outro-lugar(x)	$\leftarrow \text{Foto-de}(x,y) \wedge \text{Distante}(y)$
Contradição( )	$\leftarrow \text{VERDADEIRO EXCETO } \exists x \mid \text{Suspeito}(x)$
Beneficiário(Alberto)	
Beneficiário(Beto)	
Beneficiário(Carlos)	
Registro-de-hotel(Alberto,Manaus)	
Distante(Manaus)	
Defende(Beto,Cunhado)	
Foto-de(Carlos,Guaratuba)	
Distante(Guaratuba)	

Fonte: Rich e Knight, Inteligência Artificial, p. 244

A abordagem de *sistema de debates* busca encontrar várias respostas. A pergunta *Suspeito(x)*, os três possíveis suspeitos seriam considerados. Depois haveria uma tentativa de escolher argumentos, p.ex. uma regra de escolha que dissesse ser mais provável que as pessoas mintam para defender a si mesmas ou conhecidos, do que para defender desconhecidos e uma outra regra de escolha que dissesse que é preferível acreditar em registros de hotéis ou em registros de televisão do que em pessoas, poderia levar o resolvidor a concluir que Beto é o suspeito mais provável e/ou que existem outros suspeitos.

**Retrocesso dirigido por dependência.** Em um sistema com busca em *profundidade*, pode ocorrer a seguinte situação [RIC 93]: deve-se derivar um fato F, o qual não pode ser derivado monotonicamente da base de conhecimentos, mas pode ser derivado se uma suposição A parecer plausível. Assumindo-se A, F é derivado, e depois G e H, fatos adicionais, são derivados a partir de F. No decorrer do processo também são derivados os fatos M e N, mas eles são completamente independentes de A e F. Então surge um fato que invalida A. É preciso anular a prova de F e as provas de G e H, pois dependem de F. Como M e N não dependem de F, não há necessidade lógica de invalidá-los.

Se o sistema acima utiliza um esquema tradicional de retrocesso, também denominado *retrocesso cronológico*, pois depende da ordem cronológica em que as decisões foram tomadas, todas as conclusões derivadas após A ter sido assumido até o fato que invalida A, devem ser abandonadas. Assim, é necessário voltar a N e M, desfazendo-os, e a partir deles, retornar a H, G, F e A, para desfazei-los. Para contornar esse problema, Stallman e Sussman [Apud RIC 93] propuseram, em 1977, o *retrocesso dirigido por dependência*, que é um método de retrocesso baseado em dependências lógicas das derivações e não na ordem cronológica em que foram obtidas. Assim pode-se eliminar apenas a su-

posição que estava errada (A) e as suposições que explicitamente dependiam dela (F, G e H).

Para implementar esse tipo de retrocesso é necessário realizar uma busca de árvore e:

- associar a cada nó uma ou mais justificativas, cada uma correspondendo a um processo de derivação que levou ao nó e contendo uma lista de todos os nós (fatos, regras, suposições) dos quais depende a derivação;
- prover um mecanismo que, dado um nó de contradição e sua justificativa, calcule o conjunto de suposições “não adequadas” que sustentam a justificativa;
- fornecer um mecanismo para analisar o conjunto de suposições “não adequadas” e escolher uma suposição para retirar;
- fornecer um mecanismo para propagar o resultado da retirada de uma suposição, invalidando todas as justificativas que direta ou indiretamente dela dependiam.

Os sistemas de manutenção da verdade baseados em justificativas e os baseados em lógica são exemplos de implementações de retrocesso dirigido por dependência.

**Sistema de manutenção da verdade.** Um TMS (do inglês, *Truth Maintenance System*) é um mecanismo que permite que as declarações sejam conectadas através de uma *rede de dependências lógicas* e serve como um “guarda-livros” para um resolvidor de problemas de um sistema não-monotônico com busca em profundidade e retrocesso dirigido por dependência. O resolvidor de problemas fornece ao TMS as declarações e as dependências lógicas entre elas. Esse mecanismo foi proposto por Jon Doyle [DOY 79].

Um TMS pode ser baseado em justificativas (JTMS, do inglês *Justification-based TMS* ou simplesmente TMS), baseado em lógica (LTMS, do inglês *Logic-based TMS*) ou Baseado em suposições (ATMS, do inglês *Assumption-based TMS*).

O JTMS (em conjunto com o resolvidor de problemas) constrói justificativas para cada declaração de uma rede de derivações (dependências lógicas), utilizando o retrocesso dirigido por dependência e busca em profundidade. Os nós são tratados como átomos pelo JTMS, o qual assume não existir nenhum relacionamento entre eles, exceto aqueles explicitamente definidos nas justificativas. JTMS não possui nenhum mecanismo automático para detecção de contradições. A rede de dependências é um meio puramente sintático e independente do domínio de representar crenças e mudá-las consistentemente.

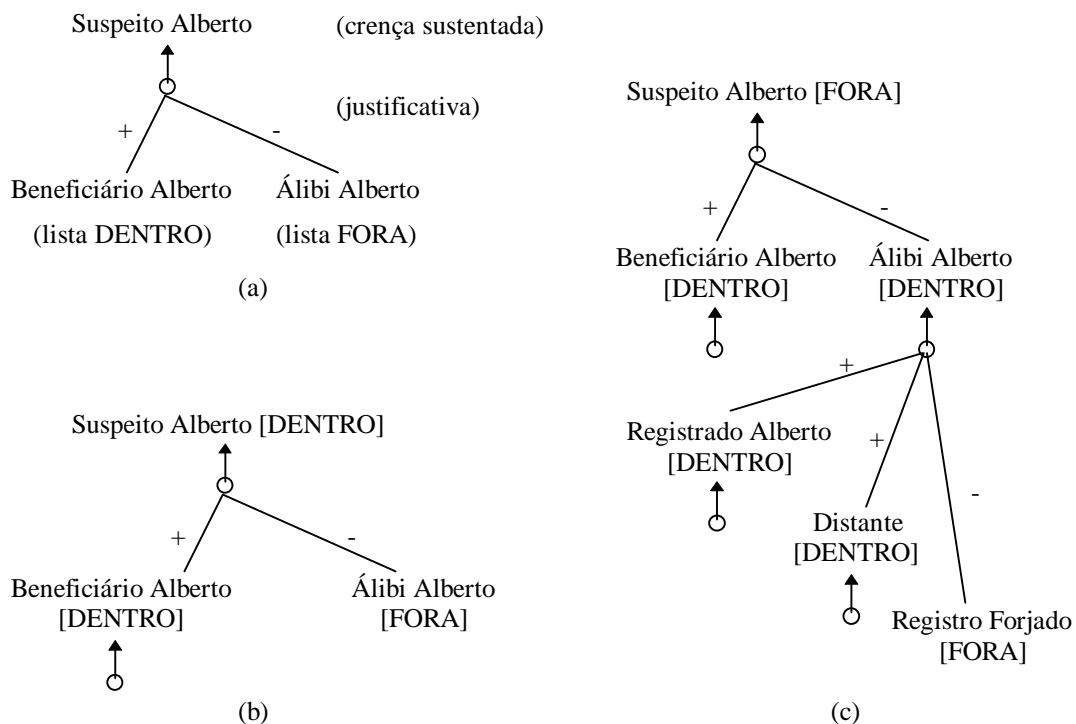
Cada nó da rede de derivações possui uma declaração e uma justificativa associada. Cada *justificativa* consiste de duas partes: uma lista DENTRO e uma lista FORA. Cada subconjunto de justificativas é ligado à declaração que sustenta. As declarações da *lista DENTRO* são asserções em que acreditamos (i.é. a favor da crença na declaração que sustentam) e as declarações da *lista FORA* são asserções em que não acreditamos (i.é. contrárias à crença). Um nó (e a declaração a ele associado) têm crédito quando contém justificativa válida. Nesse caso a declaração de *per si* é marcada como DENTRO. Se não houver um bom motivo para se acreditar na declaração, ela é marcada como FORA. Uma *justificativa é válida* se toda declaração na sua lista DENTRO e nenhuma na sua lista FORA, for tida como verdadeira.

A tarefa de marcação de nós de um TMS satisfaz a dois critérios sobre a estrutura da rede de dependência: consistência e boa fundamentação.

Pelo *critério de consistência*, todo nó marcado DENTRO é sustentado por, pelo menos, uma justificativa válida, i.e uma justificativa que possui todos os seus nós da lista DENTRO marcados DENTRO e todos os seus nós da lista FORA marcados FORA.

Pelo *critério da boa fundamentação* uma cadeia de justificativas sobre um grupo de nós não deve depender propriamente de nenhum dos nós que ela sustenta. Logo deve ser evitado um argumento circular. Em particular, se a sustentação de um nó depende apenas de uma cadeia ininterrupta de ligações da lista DENTRO que levam de volta a ela mesma, então esse nó deve ser marcado como FORA.

Vamos retomar o caso do “Assassinato ABC”. A Figura 7.1(a) ilustra o nó associado a declaração ‘Suspeito Alberto’ e a sua justificativa, mas sem nenhuma marcação de crença. A Figura 7.1(b) ilustra o mesmo nó, com marcação de crenças, supondo que ainda não temos nenhuma informação sobre a credibilidade do hotel no qual Alberto afirma ter se hospedado. *Beneficiário Alberto* foi marcado DENTRO porque recebeu uma justificativa de premissa (representada por uma seta). Uma *justificativa de premissa* é uma justificativa para um fato que precisamos apenas aceitar, sem nenhuma outra justificativa formal. Portanto possui listas DENTRO e FORA vazias. A Figura 7.1(c) representa que Alberto deixou de ser suspeito após ter-se verificado seu registro em um hotel de bom nome e distante do local do crime. Assim seu álibi passa a ter credibilidade pois não temos nenhum motivo para pensar que seu registro tenha sido forjado. Portanto *Suspeito Alberto* não é mais declaração em que acreditamos (válida) e precisa ser marcada como FORA, visto que sua lista FORA está marcada como DENTRO (é válida). Por outro lado não associamos nenhuma justificativa de premissa a declaração *Registro Forjado*, o que representa que no *momento* não temos nenhum motivo para acreditarmos que foi forjado. A marcação de crenças na Figura 7.1 seguiu o critério de *consistência*.



**Figura 7.1** - Exemplo de Rede de Dependências em JTMS

De forma similar ao caso Alberto, podemos construir uma rede de dependências para a declaração *Suspeito Beto*, com marcação FORA. Já para o caso Carlos, a princípio obteremos uma marcação DENTRO, durante a fase em que ele estivesse documentando o seu álibi. Este seria o caso de uma justificativa sem *boa fundamentação* pois o álibi dependeria de se considerar que estivesse falando a verdade e o suporte para o fato dele estar falando a verdade seria sabermos se ele esteve na competição. Com a apresentação de imagens de TV, o álibi de Carlos fica inquestionável. Assim a introdução desse novo fato na lista de justificativas de seu álibi força marcação DENTRO para a declaração *Álibi Carlos* (i.é temos bons motivos para acreditar no álibi de Carlos).

Como todos os três suspeitos apresentaram álibis considerados válidos, a marcação de cada uma das declarações de que Alberto, Beto e Carlos é suspeito deve ser FORA. Como também não temos outros suspeitos, a declaração *Outros Suspeitos* também deve ser marcada com FORA. A justificativa de *Contradição* passa a ser válida, pois a sua lista FORA só contém marcação FORA e sua lista DENTRO é vazia, e deve ser marcada com DENTRO. A tarefa do JTMS passa a ser determinar como tornar a marcação FORA para *Contradição* (i.é levantar a contradição), invalidando todas as justificativas de uma ou mais das suas declarações de sustentação.

As justificativas monotônicas não podem ser invalidadas sem que se retire da base de conhecimento as declarações explícitas, utilizadas na rede de dependências, que lhe dão sustentação. Já as justificativas não-monotônicas podem ser invalidas desde que sejam definidos alguns fatos, cuja ausência seja exigida pela justificativa. A justificativa é *não-monotônica* se sua lista FORA não for vazia ou se qualquer declaração da sua lista DENTRO possuir uma justificativa não-monotônica. Essas justificativas são denominadas *suposições*. Uma suposição pode ser invalidada se algum elemento de sua lista FORA de justificativas receber status DENTRO. Embora possa haver muitas suposições a rede de dependências permite identificar as que são relevantes para a contradição que se deseja levantar. Com o uso de algoritmos de retrocesso dirigido por dependências lógicas, as ligações da rede de dependência podem ser utilizadas para construir uma árvore E/OU de suposições que possam ser eliminadas e formas de eliminá-las através da justificativa de outras crenças. Nós E de uma *árvore E/OU* representam situações (resultantes da divisão do problema original em subproblemas a resolver, justificativas associadas a lista DENTRO, etc.) que devem ocorrer em conjunto e os nós OU representam maneiras alternativas em que uma situação pode ser representada (p.ex. formas pelo qual o problema original pode ser resolvido, justificativas associadas a lista FORA, etc.).

Com os dados da rede da “História do Assassinato ABC” o JTMS construiria uma árvore OU para a declaração *Contradição*, composta de quatro nós OU (*Suspeito Alberto*, *Suspeito Beto*, *Suspeito Carlos* e *Outros Suspeitos*). Nesta situação, as implementações iniciais de JTMS escolhiam um dos quatro nós ao acaso para levantar a contradição. As arquiteturas mais recentes deixam a escolha a cargo do resolvidor de problemas, o qual pode recorrer ao usuário para resolver tal questão [RIC 93]. Suponha que uma alternativa tenha sido escolhida (p.ex. Beto mentiu). Então qual deve ser a justificativa para ela? Um JTMS pode ter algoritmos que criam tais *justificativas abduativas* (assim denominadas por serem criadas através de raciocínio abduativo). Se mais tarde se descobrir uma outra maneira de levantar a contradição, essa justificativa abduativas deve ser revogada.

Resumindo a teoria apresentada sobre JTMS vimos que as suas principais atividades são fazer uma *marcação consistente* e *resolver contradições*, enquanto que as principais tarefas do resolvidor de problemas são: *aplicar regras para derivar conclusões*, *criar justificativas para os resultados da aplicação das regras* (como parte da resolução das contradições), *escolher uma alternativa dentre as possíveis para resolver uma contradição* e *detectar contradições lógicas*.

**Sistema de manutenção da verdade baseado em lógica.** Em um JTMS um nó de contradição não representa uma contradição lógica, mas um estado da base de conhecimentos *explicitamente* declarado como indesejável. Os nós da rede de dependência são tratados como átomos por um JTMS, o qual assume que não existe nenhum entre eles, exceto aquele explicitamente definido nas justificativas. Um LTMS, proposto por McAllester [Apud RIC 93], em 1980, é similar ao JTMS no que se refere a marcação de nós e busca em profundidade, mas difere desse por controlar os relacionamentos lógicos entre os nós da rede de justificativas, o que permite a detecção automática de contradições. Se a base de conhecimentos da “Historia do Assassinato ABC” fosse representada em um LTMS, não teria sido necessário criar a declaração explícita para detectar contradição (*Contradição()* ← *VERDADEIRO EXCETO*  $\exists x / \text{Suspeito}(x)$ ). Em vez disso substituiríamos esse nó por um que afirmasse *Nenhum suspeito* e definiríamos outro nó associado a *Suspeito*. A contradição lógica seria detectada automaticamente quando *Nenhum suspeito* e *Suspeito* fossem marcados com DENTRO.

**Sistema de manutenção da verdade baseado em suposições.** Nos sistemas JTMS e LTMS segue-se sempre uma única linha de raciocínio de cada vez, devido ao fato da busca ser em profundidade. Caso seja necessário mudar as suposições do sistema, é utilizado o raciocínio por dependências. Já o sistema ATMS, proposto por De Kleer [Apud RIC 93], em 1986, também marca nós (correspondendo a declarações), mas utiliza *busca em amplitude*. Portanto segue diversas linhas alternativas de raciocínio (múltiplos contextos), cada uma das quais correspondendo a um grupo de suposições consistentes. Durante o processo de raciocínio, o universo de contextos consistentes é podado quando são descobertas inconsistências em cada uma das justificativas. Os contextos restantes são utilizados para marcar declarações, indicando assim em quais contextos cada declaração tem uma justificativa válida. Como no caso do JTMS, esse TMS não considera existir nenhum relacionamento entre os nós da rede de justificativas, exceto aqueles explicitamente definidos nas justificativas, não possuindo, em consequência, nenhum mecanismo automático para detecção de contradições. No caso de um JTMS ou um LTMS uma declaração era marcada com FORA ou DENTRO. Em um ATMS, uma declaração será marcada com um *contexto consistente* se ela tiver pelo menos uma justificativa que possa ocorrer em um contexto consistente.

## 7.4 Vantagens e Desvantagens

As técnicas apresentadas nessa seção permitem raciocínio com modelos incompletos, inconsistentes e mutável do mundo quando se pode atribuir status de crença indiscutível (acreditar em ou não acreditar em) a declarações, ao invés de diversos graus de crença, o que as vezes é muito limitante. Esses formalismos representam incerteza somente se predicados com interpretação incerta forem utilizados [BHA 86].



Essas técnicas estão associadas ao esquema de representação de conhecimento baseado em lógica ou em regras de produção. São de complexidade elevada e requerem maior esforço computacional. Podem torna-se proibitivas se forem utilizadas com bases de conhecimento de grande porte.

Um TMS pode ser visto como um procedimento de prova para sistema não-monotônico que mantém a base de conhecimentos na forma de uma árvore onde um nó representa uma hipótese e seus filhos representam várias justificativas. Quando ocorre uma contradição o TMS revê as justificativas e formula novas hipóteses sobre valores de variáveis de forma a manter a consistência. A decisão final a qual o TMS chega não contém nenhuma informação sobre a incerteza a ela associada. Jon Doyle [DOY 83] reconhece que não possuem facilidades de cálculo de crença, “que pode ser necessário para resumir a estrutura de grandes conjuntos de extensões admissíveis assim como quantificar níveis de confiança”. As abordagens simbólicas modelam apenas a *incompletude* da informação e falham ao representar a *ignorância* sobre o valor de uma variável. Essa ignorância pode ser representada através da teoria da evidência de Dempster-Shafer.

Usar JTMS é melhor quando se deseja uma única solução ao passo que um ATMS é mais eficiente quando houver necessidade de todas as soluções. Por outro lado, como o ATMS considera todas as possíveis combinações de suposições, todas elas precisam ser conhecidas desde o início da solução do problema, já que a alternativa de recompilação sempre que uma nova suposição for acrescentada à rede de dependências é muito cara computacionalmente. Já o JTMS permite a adição gradual de novas suposições, mas pode passar muito tempo mudando de um contexto para outro quando há necessidade de retroceder dirigido por dependências lógicas. No caso do ATMS não há mudança de contextos, visto que todos eles são analisados em paralelo.

Até o presente momento não temos conhecimento da proposição de sistemas de manutenção da verdade baseado em lógica que realizem busca em amplitude (i.é que considerem múltiplos contextos paralelamente).

Sistemas de raciocínio não-monotônico baseados em LD foram implementados na área de diagnósticos. Já aplicações baseadas na LNM foram implementadas na área de raciocínio sobre ações, com ênfase especial no problema do modelo. O *problema do modelo* pode ser descrito como “assumir que tudo fica igual depois de uma ação, com exceção das coisas que necessariamente mudam” [RIC 93, p.263].

O raciocínio não-monotônico também tem sido utilizado na extração de intenções em expressões em linguagem natural.

## 8 Teoria do Endosso

Paul Cohen propôs uma abordagem simbólica informal (heurística) para representação da incerteza baseada na observação de que “a força da evidência em qualquer situação é uma soma dos diversos fatores relacionados com a certeza. As abordagens numéricas resumem todas as evidências pró e contra em um único número. Um raciocínio inteligente em geral discrimina entre esses fatores. Logo a representação resumida é inadequada.” [BHA 86]. A abordagem de Cohen, denominada teoria do endosso, é centrada na idéia de se lidar com as razões para se acreditar ou desacreditar em uma hipótese. *Endossos* são registros de fontes de incerteza e representam conhecimento sobre situações incertas [Cohen 1983 Apud SUL 85]. Tal conhecimento pode incluir, mas não se limita a, razões para acreditar e para desacreditar em proposições incertas [SUL 85].

Embora endossos possam ter nomes mnemônicos, são apenas “*tokens*”, cujos significados são obtidos através da especificação das situações às quais são aplicados, formas de combinação e critérios de ordenação. Portanto a construção de um sistema baseado em endossos requer identificar e nomear as fontes de certeza e incerteza em um domínio; especificar como essas fontes interagem de forma tal que se possa combinar as evidências; e especificar regras para ordenar combinações de endossos, de maneira a permitir tomar decisões com base nessa classificação. Os nomes atribuídos a tais fontes são denotados endossos. Eles podem ser *positivos*, significando que suportam a hipótese a que estão associados ou *negativos*, representando razões para desacreditar nela.

Para Shafer [SHA 90, p.627], endossos são “representações simbólicas de diferentes itens de evidência, das questões que suportam e das suas relações. Eles podem atuar sobre outros endossos, conduzindo a não validade de conclusões prévias. Como não há consideração formal de conclusões finais, o processo pode ser visto mais como implementação procedimental de padrão não-monotônico de raciocínio do que como lógica”.

Paul Cohen desenvolveu o programa SALOMÃO (do inglês, SOLOMON) para testar a teoria do endosso. Esse programa tomava decisões sobre portfólio de investimentos, sendo projetado para evitar totalmente o uso de números na representação de incerteza. A descrição deste programa, apresentada a seguir, está baseada em [BHA 86].

No SALOMÃO, endossos são representados através de regras utilizando lógica de predicados de primeira ordem, sendo o encadeamento reverso (“*backward*”) a estratégia de controle utilizada. A base de conhecimentos do sistema difere dos sistemas tradicionais no sentido de que suas proposições não podem ser simplesmente assumidas como verdadeiras. Cada uma delas deve ser analisada com base em seu endosso, sendo que mesmo declarações com endosso negativo podem ser adicionadas à base de conhecimento. É de responsabilidade de quem fornece os dados da base de conhecimentos, também fornecer os endossos deles. Esses endossos refletem a incerteza associada ao conhecimento disponível e podem ser relativos a origem, tipo e precisão dos dados.

O sistema opera com tarefas deduzidas a partir da aplicação de regras de inferências a proposições da base de conhecimentos. O endosso dessas regras são usados para construir endossos para a tarefa sendo criada. Todas as tarefas são colocadas em uma *agenda* em uma ordem determinada pelo seu respectivo endosso, o qual reflete uma medida da

incerteza associada à tarefa. Ao selecionar uma tarefa da agenda para execução, SALOMÃO verifica se ainda é necessário executá-la, pois ela pode ter se tornado redundante. O sistema altera a confiança em tarefas, em decorrência da análise de seus endossos. Por exemplo, se as conclusões de diversas tarefas são fortalecidas entre si, os seus endossos podem ser combinados; ou se uma cláusula de um endosso não é mais necessária ou o sistema está tendo dificuldades em satisfazê-la, ela pode ser retirada. Em um sistema baseado em regras, a conclusão de uma regra é aceita se a sua condição (antecedente) for satisfeita (i.e. está na base de conhecimentos ou pode ser derivada). O SALOMÃO aceita uma conclusão de uma regra mesmo se a sua conclusão não for satisfeita. O endosso associado à conclusão reflete a satisfação ou não da condição da regra.

Além dos endossos, o sistema utiliza os seguintes tipos de conhecimentos, expressos na forma de *regras* para: inferências do domínio, incluir tarefas na fila da agenda, selecionar a tarefa a executar da agenda, propagar endossos através das inferências, determinar se uma proposição pode ser considerada como certa, e resolver tarefas. As cláusulas individuais das condições de uma regra do domínio podem ter os seguintes tipos de *endosso de condição* (os quais especificam a relação da condição no antecedente com o conseqüente da regra): talvez muito geral, talvez muito específico, exato, confirmativo, necessário, conjectural, hipótese do mundo fechado, flexível, inflexível.

Além dos endossos associados às condições, as regras podem ter *endossos de regra de inferência*, sendo que os principais são:

- *baseado em modelo*: quando é possível fornecer uma explicação cabal de porque o estado representado na condição conduz a acreditar no estado representado na conclusão da regra;
- *causal*: quando o estado descrito na condição da regra causa o estado descrito na sua conclusão (que nesse caso representa um ação);
- *correlacional*: quando o estado descrito na condição está associado com o estado descrito na conclusão, mas o relacionamento não é causal.

Quando a tarefa é incluída na agenda, além dos endossos decorrentes da aplicação das regras de inferência do domínio, também recebe um dos seguintes *endossos de tarefa*:

- *p-corroboração*: a conclusão da tarefa fortalece a evidência na conclusão de outra tarefa incluída na agenda;
- *p-conflito*: a conclusão da tarefa conflita com a conclusão de outra tarefa incluída na agenda;
- *p-conflito-potencial*: a conclusão da tarefa pode conflitar com a conclusão de outra tarefa na agenda;
- *p-redundante*: a conclusão da tarefa é idêntica a conclusão de outra tarefa, derivada através da aplicação da mesma regra e que tem a mesma regra de endosso.

Ao concluir uma tarefa é gerado um *endosso de conclusão de tarefa* (em geral refletindo o endosso de tarefa) contendo um dos seguintes “*tokens*”: corroboração, conflito, conflito potencial, redundante, improvável, modal, ou injustificável. Quando o sistema raciocina com as regras de inferências, os endossos dos antecedentes (endosso de condições) são transferidos para o conseqüente. O endosso associado a conclusão derivada inclui ainda os endossos da regra de inferência, das tarefas e da conclusão da tarefa. Como a conclusão derivada pode ser utilizada para provar novas tarefas, após poucas inferências, podemos ter conclusões com muitos endossos associados.

HMMM é uma aplicação desenvolvida por Cohen, em conjunto com Sullivan [SUL 85], baseada na teoria do endosso, para o reconhecimento de planos. Ela combina evidências fornecidas por ações sucessivas do usuário, para inferir, dentre um conjunto de planos conhecidos, qual(is) o usuário está seguindo. Nessa aplicação a fonte de incerteza reside em ações erradas ou ambíguas (no sentido de que possam ser consistentes com diversos planos) tomadas pelo usuário. Quando uma ação pode ser interpretada como erro ou ambígua, existe uma incerteza com relação a intenção do usuário e o HMMM gera endossos para as interpretações possíveis. O sistema usa o conhecimento sobre as ações anteriores do usuário para restringir as interpretações da ação corrente e associa um endosso a cada interpretação possível (de todas as ações tomadas até o momento).

Para ilustrar as suas idéias, Sullivan e Cohen apresentam o seguinte exemplo. Suponha que saibamos que o usuário possui apenas dois planos, p1 e p2, sendo p1 composto pelos passos  $(a,b,c)$  e p2 pelos passos  $(b,d,e)$ , nesta ordem. Suponha ainda que o usuário informou ao HMMM que as suas ações são  $a$ ,  $b$  e  $d$ , nesta ordem. A Tabela 8.1 apresenta as interpretações possíveis e seus endossos.

**Tabela 8.1** - Exemplo de Aplicação de Endossos em Reconhecimento de Planos

Passo	Interpretação	Endosso
1. $a$	(inicia plano p1: $a$ )	( $a$ única possibilidade pela gramática +) ( $a$ pode ser um erro -)
2. $b$	(continua plano p1: $b$ )	( $a b$ continuidade é desejável +) ( $b$ outra possibilidade pela gramática -) ( $b$ pode ser um erro -)
$b$	(inicia plano p2: $b$ )	( $a b$ continuidade é indesejável -) ( $b$ outra possibilidade pela gramática -) ( $b$ pode ser um erro -)
3. $d$	(continua plano p2: $d$ )	( $b d$ continuidade é desejável +) ( $d$ única possibilidade pela gramática +) ( $d$ pode ser um erro -)

Fonte: Sullivan e Cohen 1985. An Endorsement-Based Plan Recognition Program.

Assim  $a$  representa evidência para p1. Já  $b$  pode ser interpretado como mais evidência para p1 (seqüência  $a b$ ) ou evidência para o plano p2. A entrada  $d$  reforça associar  $b$  ao plano p2. Os endossos são associados às interpretações através das *regras de condições de aplicabilidade*: “única possibilidade pela gramática” é aplicável quando a ação só pode estar associada a um único plano, caso contrário se aplica “outra possibilidade pela gramática”; “pode ser um erro” sempre é aplicável; “continuidade é desejável” se aplica se a ação é interpretada como a continuidade de um plano; “descontinuidade é indesejável” está associada a interpretada de descontinuidade de um plano em execução e início de um novo plano; “+” se o endosso reforça a interpretação e “-”, caso contrário.

São utilizadas regras de *combinações semânticas* para combinar os endossos fornecidos por diversas evidências. HMMM implementa apenas as regras:

- Se no plano N, (passo i *pode ser um erro* -) e (passos i j *continuidade é desejável* +)  
Então elimine: (passo i *pode ser um erro* -).
- Se no plano N, (passos i j *descontinuidade é indesejável* -) e (passos j k *continuidade é desejável* +) e nos planos N, M, (passo j *outra possibilidade pela gramática* -)  
Então elimine, no plano N, (passos i j *descontinuidade é indesejável* -).

Essas regras estão baseadas na idéia de que passos consecutivos em um mesmo plano eliminam a incerteza sobre a interpretação dos passos iniciais. Assim elas utilizam a ocorrência de dois passos consecutivos como base para remover endossos negativos associados ao primeiro passo. A aplicação destas regras ao nosso exemplo conduz aos seguintes endossos combinados:

Interpretação de p1 com base em a b:

(a única possibilidade pela gramática +)  
 (a b continuidade é desejável +)  
 (b outra possibilidade pela gramática -)  
 (b pode ser um erro -)

Interpretação de p2 com base em a b d:

(b outra possibilidade pela gramática -)  
 (d única possibilidade pela gramática +)  
 (b d continuidade é desejável +)  
 (d pode ser um erro -)

Para ordenar conjuntos de endossos são implementados dois métodos. O primeiro é baseado no uso de pesos numéricos. Os autores argumentam que as regras de combinação vistas não são intuitivas porque elas eliminam inteiramente um endosso ao invés de aumentar ou reduzir o seu peso. Assim eles propõem o uso de pesos numéricos para representar variação da força do endosso resultante da combinação de diversos endossos. O HMMM utiliza apenas uma única forma de combinação, denominada *corroboração* de endossos, na qual dois endossos são combinados produzindo um endosso de maior peso. Foram identificadas três situações possíveis: a) múltiplas instâncias do mesmo endosso dentro de um único passo de plano (p.ex. o peso de “continuidade é desejável” é maior quando um passo ambíguo pode continuar um plano e iniciar numerosos outros); b) instâncias de diferentes endossos de mesmo sinal dentro de um único passo de plano (p.ex. o peso conjunto de “discontinuidade é indesejável” e “outra possibilidade pela gramática” é maior do que a soma de seus pesos individuais); e c) múltiplas instâncias do mesmo endosso em passos consecutivos (p.ex. o peso do endosso associado a um plano é aumentado ao se observar diversas ocorrência de “continuidade é desejável”). O segundo método de ordenação de conjuntos de endossos é baseado na obtenção de uma ordenação parcial, visando separar alternativas provável das improváveis. Cada alternativa é classificada como membro de uma das classes “provável”, “improvável” e “neutra” em função dos endossos associados. Será considerada membro de “provável” se seu conjunto de endosso possuir dois endossos positivos e de “improvável” se ocorrer qualquer endosso negativo.

## 8.1 Vantagens e Desvantagens

A vantagem de endosso é explicitar as fontes de incerteza, permitindo raciocinar com elas diretamente, ao invés de implicitamente como é o caso da representação numérica de incerteza. Esse formalismo facilita a explicação, pois além dos endossos armazena a relevância de cada proposição com relação a um objetivo específico. O raciocínio baseado em endosso não é normativo ou prescritivo: assim não existe um conjunto correto de endossos para um domínio nem um método correto de combina-los. O problema de selecionar uma alternativa dentre algumas concorrentes que possuem diferentes conjuntos de endossos não é fácil de ser resolvido, pois em geral não se consegue obter uma ordenação total, com base nos conjuntos de endossos. Cohen sugere apenas que o conhecimento do domínio e da situação, sobre a incerteza, seja utilizado.

Não há uma interpretação definida para endossos, pois eles assumem diferentes papéis e com significados diversos em contextos diferentes. Sua semântica é especificada pelas suas condições de aplicabilidade, regras de combinação e regras de ordenação.

## 9 Teoria de Probabilidades e Fatores de Certeza

Neste capítulo são apresentados os formalismos de teoria de probabilidades (estatística bayesiana) e teoria da confirmação (fatores de certeza) para representar e tratar incerteza. A exposição sobre estatística bayesiana é concluída com a apresentação dos problemas de complexidade decorrentes do uso dessa técnica em aplicações reais de porte, o que inviabiliza a sua utilização pura e simples em sistemas com incerteza. Então é apresentado o formalismo de anexação de fatores de certeza a regras, o qual constitui um dos mecanismos desenvolvidos para explorar o potencial da estatística bayesiana, com abordagem tratável dos problemas citados acima, em sistemas de raciocínio com incerteza. Os mecanismos de redes bayesianas e teoria de Dempster-Shafer, também desenvolvidos com o mesmo objetivo, serão apresentados nos próximos capítulos.

Essas técnicas numéricas não modificam os fundamentos matemáticos fornecidos pela lógica e pela teoria dos conjuntos, mas ampliam essas idéias com construções adicionais fornecidas pela teoria de probabilidades.

Para exemplificar a apresentação de fatores de certeza, vamos focar nossa atenção na incorporação de incerteza no raciocínio para trás, utilizado pela linguagem Prolog. Para tanto é necessário representar incerteza na base de conhecimentos e criar um meta-interpretador para combinar e resumir a incerteza anexada as regras Prolog.

### 9.1 Teoria de Probabilidades

Os esquemas de representação de conhecimento vistos até o momento, são aplicados a informação *categórica*: os itens estão ou não armazenados na base de conhecimento e as conclusões são referendadas ou não pelo procedimento de inferência do sistema. Na prática muitas vezes existem diversas gradações de certeza. Pode-se estar quase certo sobre alguns fatos, menos certo sobre outros e sem nenhuma evidência sobre outros. Combinando informações probabilísticas com o conhecimento representado (principalmente por regras de produção ou lógica de predicados) é possível descrever declarações com graus de crenças associados, para as quais existem evidências de apoio.

Probabilidade é uma medida atribuída a eventos. Seja o evento  $A \subseteq \Omega$ , onde  $\Omega$  denota o espaço amostral (no caso o conjunto universo de discurso) e  $\subseteq$  a inclusão não estrita. Então a probabilidade de ocorrência de  $A$  é dada por  $P : 2^\Omega \rightarrow [0,1]$ , satisfazendo os seguintes axiomas, sendo  $2^\Omega$  é o conjunto das partes de  $\Omega$ :

- i)  $\forall A \subseteq \Omega, 0 \leq P(A) \leq 1$  (0 e 1 são limitantes)
- ii)  $P(\Omega) = 1$  (prob. do evento certo)
- iii)  $\forall (A,B) \subseteq \Omega$  se  $A \cap B = \emptyset$  então  $P(A \cup B) = P(A) + P(B)$  (aditividade),

onde  $\emptyset$  é o conjunto vazio. Pelo axioma da aditividade, a probabilidade de ao menos um dentre dois eventos mutuamente exclusivo se realizar é dada pela soma de suas probabilidades individuais. Podemos então concluir que:

- i)  $P(\emptyset) = 0$  (prob. do evento impossível)
- ii)  $\forall A \subseteq \Omega, P(A) = 1 - P(\bar{A})$
- iii)  $\forall (A,B) \subseteq \Omega, P(A \cup B) = P(A) + P(B) - P(A \cap B),$

onde  $\bar{A}$  é o evento complementar de  $A$  em relação a  $\Omega$ . Observe que estes axiomas não facilitam encontrar o valor numérico da probabilidade para um dado evento. Esses valores devem ser encontrados com base na experimentação (enfoque freqüencista) ou no julgamento *a priori* (enfoques de resultados igualmente prováveis e subjetivo).

Na interpretação *freqüencista* a probabilidade do evento aleatório  $A$  é o valor limite da freqüência de ocorrência de  $A$  em relação ao número total de tentativas, em uma sucessão infinita e independente de repetições do experimento para verificação de  $A$ . Por exemplo, a probabilidade de ocorrência de um acidente aéreo em um certo aeroporto pode ser estimada por esse enfoque, analisando as estatísticas dos acidentes ocorridos.

Na interpretação de *resultados igualmente prováveis*, se todos os resultados possíveis de um experimento são igualmente prováveis, então a probabilidade de um evento é a razão entre o número de ocorrências favoráveis e o número total de resultados possíveis. Assim, em um dado não viciado, a probabilidade de ocorrência dos números 4 ou 5 é  $2/6$ . Essa interpretação e a freqüencista são muito utilizadas em engenharia e na estatística.

Na interpretação *subjetiva* a probabilidade de ocorrência do evento  $A$  representa a *crença* de um determinado agente na ocorrência de  $A$ . Esse agente deve ser capaz de fornecer estimativas de probabilidades consistentes com os axiomas acima. Essa interpretação é normalmente utilizada em sistemas inteligentes. As informações probabilísticas são úteis para modelar aleatoriedade e para resumir exceções possíveis, quando a enumeração explícita (usando regras *default*, etc.) das mesmas for inviável. Esse resumo numérico informa com que freqüência podemos esperar a ocorrência de uma determinada exceção. A estatística bayesiana permite coletar evidências, durante o processo de raciocínio e usa-las para modificar o comportamento do sistema.

A utilização de probabilidades *a priori* e *a posteriori* da teoria de Bayes constitui umas das primeiras tentativas para se lidar com incerteza numérica. A noção básica da estatística bayesiana é a probabilidade *condicional*  $P(H|E)$  da hipótese  $H$ , *dado* que observou-se a evidência  $E$ . Para calculá-la é necessário levar em conta a probabilidade *a priori* da hipótese  $H$ , i.e a probabilidade associada a ocorrência de  $H$ , na ausência de qualquer evidência. Também é necessário estimar até que ponto  $E$  fornece evidências da ocorrência de  $H$ , dividindo-se o universo em um conjunto *exaustivo* e *mutuamente exclusivo* de hipóteses que se deseja discriminar. Seja:

$P(H_i|E)$  = prob. da  $i$ -ésima hipótese ser verdadeira, dado a evidência  $E$ ,

$P(E|H_i)$  = prob. da evidência  $E$  ser observada, sendo a  $i$ -ésima hipótese verdadeira,

$P(H_i)$  = prob. *a priori* da  $i$ -ésima hipótese ser verdadeira, na ausência de qualquer evidência específica.

$m$  = número de hipóteses possíveis,

então o teorema de Bayes afirma que  $P(H_i|E) = P(E|H_i).P(H_i) / (\sum_{k=1}^m P(E|H_k).P(H_k))$ .

Como  $P(E) = \sum_{k=1}^m P(E|H_k).P(H_k)$ , visto que as hipóteses  $H_k$  são exaustivas e mutuamente exclusivas,  $P(H_i|E) = P(E|H_i).P(H_i) / P(E)$ . O subjetivismo Bayesiano vê essa fórmula como uma regra para atualizar crenças, face a ocorrência de  $E$ , ou seja ela estabelece que a crença que atribuímos a hipótese  $H_k$ , após obtermos a evidência  $E$ , pode ser calculada multiplicando nossa crença prévia  $P(H_k)$ , pela verossimilhança  $P(E|H_k)$  que  $E$  irá ocorrer se a hipótese  $H_k$  for verdadeira. Os filósofos bayesianos consideram a relação condicional como um conceito mais básico do que a conjunção de eventos e mais com-

patível com a organização do conhecimento humano [PEA 87, p.31]. Nessa visão  $A|B$  representa o evento  $A$  no contexto (*frame* de conhecimento) especificado por  $B$ . Probabilidades condicionais são consideradas primitivas da linguagem e uma tradução útil da expressão “..., visto que eu sei  $A$ ”. Elas armazenam conhecimento empírico invariante. Se existirem mais de uma evidência, é preciso levá-las em conta na atualização das nossas crenças, ou seja:  $P(H_i|E_1,...,E_n) = P(E_1,...,E_n|H_i).P(H_i) / (\sum_{k=1}^m P(E_1,...,E_n|H_k).P(H_k))$ . Se a seqüência de evidências  $\{E_1, E_2, ..., E_n\}$  existente é tal que cada porção de evidência  $E$  é condicionalmente independente sob cada hipótese  $H_k$ , pode-se somar os seus efeitos, i.é  $P(E_1,...,E_n|H_i) = \prod_j P(E_j|H_i)$ . Caso as evidências *não* sejam condicionalmente independentes, é preciso representar explicitamente a probabilidade condicional resultante da conjunção delas, i.é  $P(E_1,E_2,...,E_n|H_i) = P(E_n|E_{n-1},...,E_1,H_i) .. P(E_2|E_1,H_i).P(E_1|H_i)$ .

Dada uma seqüência de porções de evidências observadas no passado  $E^n = \{E_1,...,E_n\}$  e uma nova evidência  $E$ , a probabilidade condicional da ocorrência da hipótese  $H$ , face a ocorrência simultânea das evidências  $E^n$  e  $E$ , é  $P(H|E^n,E)=P(H|E^n).P(E|H,E^n) / P(E|E^n)$ , desde que elas sejam condicionalmente independentes de  $H$  [PEA 87][SHO 75].

### 9.1.1 Vantagens e Desvantagens

O enfoque Bayesiano utiliza a interpretação subjetiva da probabilidade e possui uma semântica definida: probabilidade representa uma crença (i.é grau de confirmação de uma hipótese). É possível representar o relacionamento entre evidências e hipóteses através de uma estrutura de grafo, denominada rede bayesiana, o que permite explorar relações de causalidade e dependência condicional localizada (veja Capítulo 10).

As desvantagens desse enfoque estão centradas no fato das premissas de a) independência condicional das evidências em relação as hipóteses e b) hipóteses serem exaustivas e mutuamente exclusivas, não serem muito realistas, pois não são facilmente satisfeitas em domínio reais. Se uma rede bayesiana contém múltiplos caminhos entre o nó que representa uma dada evidência e aquele representando uma hipótese, a independência condicional não se mantém. Por outro lado a impossibilidade da ocorrência simultânea de mais de uma hipótese implica que o especialista que as elaborou possui perfeito conhecimento do domínio, sendo este compatível com a utilização da hipótese HMF.

As probabilidades condicionais conjuntas são difíceis de obter e tratar computacionalmente em problemas reais. Essa abordagem apresenta problemas de complexidade para aplicações de porte, pois requer que uma enorme matriz de probabilidades condicionais seja estimada e fornecida para o sistema, inviabilizando a aquisição de conhecimentos e implicando elevados requisitos de tempo, armazenamento e capacidade computacional para processar todas essas informações, levando a procura de novos métodos [RIC 93] [BON 90]. O renascimento do interesse em métodos probabilísticos foi em parte motivado pelos trabalhos de Judea Pearl, em especial o livro *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, sendo que a partir de 1991 a capa da revista Artificial Intelligence passou a incluir uma rede de decisão estilizada.

## 9.2 Representação de incerteza

A incorporação de medida de certeza na base Prolog pode ser feita de duas formas [SHO 94], função do fato da implementação da linguagem suportar ou não referências aos ín-



indices internos utilizados para acesso e armazenamento de cláusulas na base de conhecimentos. Se a implementação suportar essas referências (ref), para cada cláusula pode ser armazenado o fato `cert(ref, certeza)`, utilizando o predicado (Prolog padrão Edimburgo):

`cassert(+Cláusula, +Cert):-assert(+Cláusula, -Ref), assert(cert(+Ref, +Cert)).`

onde o sinal '+' indica que o parâmetro deve estar instanciado e '-' indica que não deve estar instanciado, quando o predicado for chamado. O predicado `assert` de aridade 1 (`assert/1`) inclui cláusula no final da cadeia de cláusulas para o predicado especificado pela *cabeça* da cláusula sendo incluída na base de conhecimentos. Já `assert/2` atua de forma similar, mas a inclusão é feita utilizando o índice de referência interna *Ref*. Desta forma se garante que ambas cláusula e predicado `cert` poderão ser recuperados através da mesma referência interna *Ref*.

A medida de certeza associada a uma cláusula pode ser obtida com o predicado:

`cclause(+Head, -Body, -Cert):-clause(+Head, -Body, -Ref), cert(+Ref, -Cert).`

Se a implementação do Prolog não suportar referências internas, pode-se armazenar a medida de certeza diretamente no corpo da cláusula, como, por exemplo:

`head:-cert(cert), body.`

e adaptar os predicados `cassert/2` e `cclause/3`, de forma compatível.

Os valores da certeza, por exemplo, podem ser representados simbolicamente, com valores retirados de {impossível, muito\_improvável, improvável, possível, provável, muito\_provável, certo} ou numericamente, com valores em [-1,1] (p.ex. fatores de certeza utilizados pelo MYCIN) ou com valores em [0,1] (p.ex. representação de crença e plausibilidade na teoria de Dempster-Shafer, a ser vista no Capítulo 11).

### 9.3 Meta-interpretador com incerteza

Na ausência de incerteza, cada regra pode ser tratada de uma forma *livre de contexto*. Assim se o *antecedente* da regra é satisfeito, pode-se concluir que o *consequente* é disparado, independentemente das outras regras e fatos da base de conhecimentos.

Existindo medida de certeza associada com fatos e regras, a medida da certeza da *ação* de uma regra não pode ser calculada apenas com base na regra em si, mesmo se a parte da *condição* for satisfeita, pois podem existir outras evidências de apoio ou refutação.

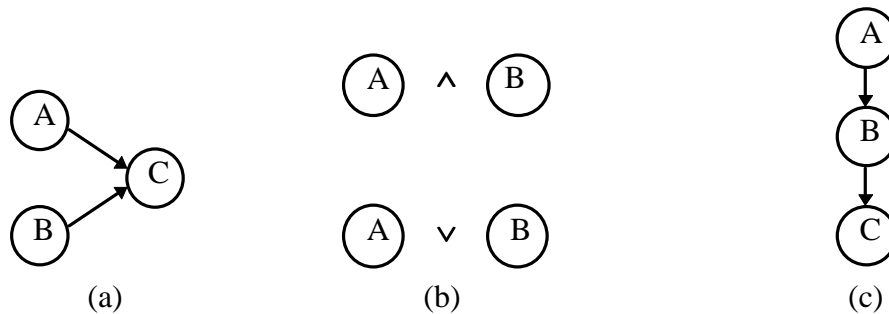
Shoham [SHO 94] apresenta o seguinte exemplo de regras Prolog com incerteza.

malária:-febre_alta, recente_selva, not(tomou_pílulas).	(muito_provável).
malária:-disenteria.	(improvável).
disenteria:-recente_3mundo.	(possível)
recente_3mundo:-recente_guatemala.	(certo)
recente_selva:-recente_guatemala	(muito_provável)
recente_guatemala.	(certo)
tomou_pílulas.	(muito_improvável).

**Figura 9.1** - Exemplo de Incerteza Simbólica em Regras

Pela primeira regra concluímos que “dada apenas a informação que o paciente tem febre alta, esteve recentemente na selva e não tomou pílulas contra malária, é altamente provável que sofra de malária”. Com apenas essa regra não podemos inferir se o paciente está sofrendo de malária; é preciso considerar outras regras relevantes. A operação do meta-interpretador permite combinar as medidas de certeza em vários estágios e estimar um valor de certeza a ser associado com o diagnóstico do paciente.

A Figura 9.2, extraída de [RIC 93], ilustrar as formas de combinação de evidências (condições) e de um grupo de regras correlacionadas.



**Figura 9.2** - Combinando Regras Incertas

Na Figura 9.2(a), várias regras fornecem evidências que se relacionam com a mesma hipótese, i.e são diversas regras que possuem a mesma conclusão, caracterizando várias fontes de evidência para uma única hipótese. Já a Figura 9.2(b) representa o encadeamento de premissas de uma mesma hipótese, correspondendo a uma regra cujo *antecedente* (condições) é um conjunto de premissas que devem ser consideradas simultaneamente. A Figura 9.2(c) ilustra o encadeamento de regras em cascata, no qual uma regra que dispara fornece informações para o disparo da seguinte, representando a situação em que o *consequente* de uma regra é parte do *antecedente* de outra.

No meta-interpretador proposto por Shoham [SHO 94], Figura 9.3, são considerados três predicados para essas formas de combinação de incerteza:

- combinação paralela ou ‘or’: `comb_fn_parallel(+ListCert,-Cert)` - estima o valor de certeza de uma hipótese disparada por diversas regras. (Figura 9.2(a)).
- combinação serial ou ‘and’: `comb_fn_serial(+ListCert,-Cert)` - calcula o valor de certeza do *corpo* da cláusula, face aos valores de certeza de cada condição. (Figura 9.2(b)).
- combinação da regra: `comb_fn_rule([+Crule,+Cbody],-Chead)` - estima o valor de certeza da *cabeça* da regra em função dos valores de certeza associados à regra em si e ao seu *corpo*. (Figura 9.2(c)).

```
certainty(not(+A), -Cert):-!, certainty(A, Cert1), complement(Cert1, Cert).
certainty(true, -Cert):-highest_certainty(Cert),!.
certainty((+A, +B),-Cert):-!,
    findall(Cert1, ( amember(X, (A, B)), certainty(X, Cert1) ), ListCert),
    comb_fn_serial(ListCert, Cert).
certainty(+A, -Cert):- findall(Cert3, (cclause(A, B, Cert1), certainty(B, Cert2),
    comb_fn_rule( [Cert1, Cert2], Cert3) ), ListCert),
    comb_fn_parallel(ListCert, Cert).
```

**Figura 9.3** - Meta-interpretador Prolog com Incerteza

*certainty(+Meta,-Cert)* recebe uma meta Prolog, a avalia e retorna uma medida de certeza. Os predicados de combinação devem ser substituídos, em função dos esquemas de representação e combinação de incerteza usados. Também são utilizados os predicados *highest\_certainty/1*, *complement/2* e *amember/2*. *Highest\_certainty(+Cert)* retorna *true* se *Cert* for o maior valor na escala de valores de certeza utilizada. *Complement(+Cert, -CmplCert)* retorna o complemento do valor *Cert*. *Amember(Membro, +ListAnd)* verifica se *Membro* está na lista *and*. Uma lista *and* é uma lista onde os elementos são separados por vírgula, i.é com o formato (a,2,d,4,5,8,10). Se *Membro* não estiver instanciado, *amember* o instancia com um elemento da lista, iniciando com o primeiro. Seu código é:

```
amember(X, (A,B)) :- !, (X = A; amember(X,B)).
amember(A,A).
```

Já os códigos de *highest\_certainty/1* e *complement/2* dependem da escala de valores de certeza utilizada. Para a escala do exemplo da Figura 9.1, seus códigos são:

```
highest_certainty(certo).
number(impossivel,1).           % converte valor simbólico em numérico
number(muito_improvavel,2).
number(improvavel,3).
number(possivel,4).
number(provavel,5).
number(muito_provavel,6).
number(certo,7).
complement(+A,-B) :- number(A, Na), Nb is 8 - Na, number(B,Nb).
```

A Tabela 9.1 apresenta exemplos de funções de combinação e respectivos códigos Prolog, utilizados para manipulação de incerteza simbólica, com a escala de valores vista.

**Tabela 9.1** - Combinação de Incerteza Simbólica

Combinação	Algoritmo	Código
Regra	Distribua a certeza da <i>cabeça</i> da regra ao redor do valor possível, em proporção a certeza do <i>corpo</i> e da <i>regra</i> propriamente dita.	comb_fn_rule([+Crule, +Cbody], -Chead) :- number(Crule, Nr), number(Cbody, Nb), Nh is 4 + ((Nr-4) * Nb // 7), number(Chead, Nh).
Serial	Selecione o valor de certeza mínimo dentre os valores de associados a cada <i>condição</i> , no <i>corpo</i> da regra.	comb_fn_serial([], certo). comb_fn_serial([+Ca   +Cb], -Cmin) :- number(Ca, Na), comb_fn_serial(Cb, Cb1), number(Cb1, Nb), (Na < Nb, !, Cmin = Ca; Cmin = Cb1).
Paralela	Selecione o valor de certeza máximo dentre os valores associados à <i>cabeça</i> comum de diversas regras.	comb_fn_parallel([], possível) :- !. comb_fn_parallel([+Ca], -Ca) :- !. comb_fn_parallel([+Ca   +Cb], -Cmax) :- number(Ca, Na), comb_fn_parallel(Cb, Cb1), number(Cb1, Nb), (Na < Nb, !, Cmax = Cb1; Cmax = Ca).

Fonte: Yoav Shoham, 1994. Artificial Intelligence Techniques in Prolog.

Ao perguntar ao sistema se o paciente do exemplo da Figura 9.1 tem malária, através da meta ‘?- *certainty(malaria, Cert)*.’, o meta-interpretador responderá com *Cert = provável*. Caso se altere o valor de certeza de  *muito provável* para *provável*, da regra relacionando malária com ter tido febre alta, ter estado na selva e não ter tomado pílulas, o sistema responderá com *Cert = possível*. Esses casos são ambos coerentes. Entretanto se forem alteradas as certezas da regra citada e do fato ‘recente\_guatemala’ para *impossível* e de todas as outras cláusulas para *certo*, o sistema responderá com *Cert = muito provável*. Esse último resultado é *incoerente*, pois a única evidência positiva para malária, o paciente ter tido disenteria, tem medida de certeza apenas *provável*! Observe que a especificação das regras de combinações foram baseadas apenas na intuição e estão sujeitas a erros!

#### 9.4 Combinações Heurísticas

Embora as combinações a serem vistas nessa seção utilizem medidas de certeza no intervalo [0,1], não representam uma medida de probabilidade. Assim como os fatores de certeza do MYCIN, elas constituem heurísticas intuitivas, sem uma justificativa formal.

As heurísticas mais freqüentes são:

- *independência* (indep): é assumido que as medidas de certeza de diferentes metas sejam independentes entre si;
- *máxima separação*<sup>12</sup>(maxdis): adota funções de combinação que minimizam a possibilidade de metas relacionadas serem disparadas ao mesmo tempo;
- *inclusão* (inc): as metas são disparadas em cascata, uma vinculando a outra.

A Tabela 9.2 apresenta as funções de combinações obtidas com as heurísticas acima.

**Tabela 9.2** - Combinações Heurísticas de Incerteza Numérica

Tipo	Regra (Chead regra)	Serial (Cbody)	Paralela (Chead diversas regras)
Indep	= Crule*Cbody	= $C(A_1) * \dots * C(A_n)$	= $1 - (1 - C(H_1)) * \dots * (1 - C(H_n))$
Maxdis	= Crule*Cbody	= $\max\{C(A_1) + \dots + C(A_n) - n + 1, 0\}$	= $\min\{C(H_1) + \dots + C(H_n), 1\}$
Inc	= Crule*Cbody	= $\min\{C(A_1), \dots, C(A_n)\}$	= $\max\{C(H_1), \dots, C(H_n)\}$

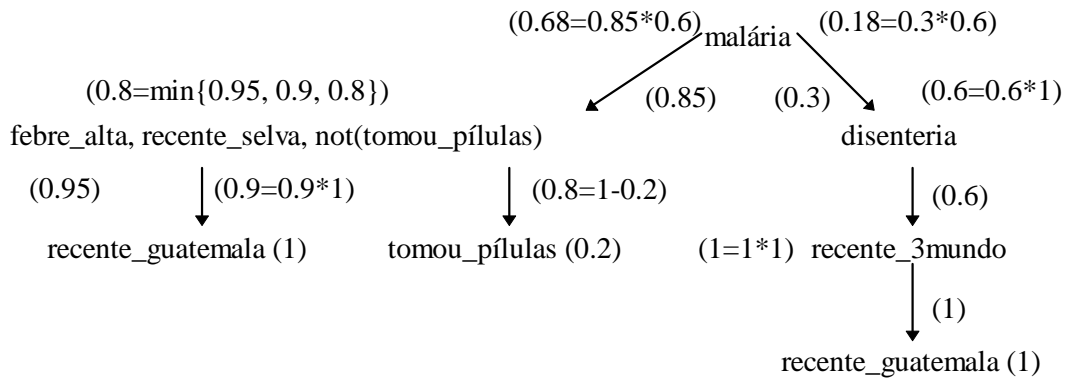
Fonte: Yoav Shoham, 1994. Artificial Intelligence Techniques in Prolog

Essas heurísticas utilizam a mesma combinação para regra, admitindo a independência entre as medidas de certezas da regra em si e a do seu *corpo*. Elas calculam a combinação serial com as medidas de certeza das *n* condições que compõem o antecedente da regra. A paralela é calculada sobre as medidas de certeza das *cabeças* (iguais) de *n* regras. Os predicados correspondentes estão ilustrados na Tabela 9.3, obtida de [SHO 94].

Com a heurística *inclusão* e alterando os predicados *highest\_certainty/1* e *complement/2* para trabalhar com valores numéricos de 0 a 1, o meta-interpretador responde com *Cert = 0,68* à demanda ‘*certainty(malaria, Cert)*’, com as medidas de certeza 0,85, 0,3, 0,6, 1,

<sup>12</sup> Maximal disjointness em Inglês.

0,9, 1, 0,95 e 0,2, respectivamente, para as cláusulas do problema da Figura 9.1. Para facilitar o entendimento, a Figura 9.4 apresenta a prova Prolog de forma esquematizada.



**Figura 9.4** - Avaliação de Incerteza com Heurística Inclusão

**Tabela 9.3** - Predicados Heurísticas de Incerteza Numérica

Combinação	Serial	Paralela
Independente	<code>serial_indep([A], B) :- !, A=B.</code> <code>serial_indep([A B], Cert) :-</code> <code>  serial_indep(B, BB),</code> <code>  Cert is A * BB.</code>	<code>parallel_indep([A], Cert) :- !, Cert=A.</code> <code>parallel_indep([A B], Cert) :-</code> <code>  multiply_indep([A B], Prod),</code> <code>  Cert is 1-Prod.</code> <code>multiply_indep([A], 1-A).</code> <code>multiply_indep([A B], Prod) :-</code> <code>  multiply_indep(B, P1),</code> <code>  Prod is (1-A) * P1.</code>
Inclusão	<code>serial_inc(ListCert, Cert) :-</code> <code>  minimal(ListCert, Cert).</code> <code>minimal([X], X).</code> <code>minimal([X Y], Min) :-</code> <code>  minimal(Y, M1),</code> <code>  ( X&lt;M1, !, Min=X; Min=M1 ).</code>	<code>parallel_inc(ListCert, Cert) :-</code> <code>  maximal(ListCert, Cert).</code> <code>maximal([X], X).</code> <code>maximal([X Y], Max) :-</code> <code>  maximal(Y, M1),</code> <code>  ( X&gt;M1, !, Max=X; Max=M1 ).</code>
Máxima se- paração	<code>serial_maxdis(ListCert, Cert) :-</code> <code>  length(ListCert, N),</code> <code>  add(ListCert, Soma),</code> <code>  S1 is Soma - N + 1,</code> <code>  maximal([0, S1], Cert).</code> <code>add([], 0).</code> <code>add([A B], Soma) :-</code> <code>  add(B, S1),</code> <code>  Soma is S1 + A.</code>	<code>parallel_maxdis(ListCert, Cert) :-</code> <code>  add(ListCert, Soma),</code> <code>  minimal([Soma, 1], Cert).</code>

Fonte: Yoav Shoham, 1994. Artificial Intelligence Techniques in Prolog

## 9.5 Fatores de Certeza do MYCIN

Fatores de certeza constituem uma abordagem prática de uso de conceitos derivados da estatística bayesiana para provê capacidade de representação de conhecimento incerto e incompleto, associado ao esquema de regras de produção. O sistema especialista MYCIN, foi o pioneiro no uso de fatores de certeza. O MYCIN tenta descobrir organismos significativos causadores de doenças infecciosas bacteriológicas e recomendar tera-

pias para combatê-las [DAV 77]. Ele interage com o médico para adquirir os dados clínicos necessários. A maioria de seus conhecimentos de diagnóstico está representada na forma de regras de produção, cada qual com um *fator de certeza*, que mede até que ponto a evidência descrita pelo *antecedente* da regra suporta a conclusão presente no seu *consequente*. Um regra típica apresenta a seguinte forma [SHO 75, p.238]:

- se (1) o pigmento do organismo é gram-positivo, e  
 (2) a sua morfologia é *coccus*, e  
 (3) a sua conformação do crescimento é de blocos  
 então existe evidência sugestiva (0,7) de que a sua identidade é *estafilococo*.

A principal simplificação, em relação a estatística bayesiana, adota pelo MYCIN é supor regras com ***independência condicional*** das evidências. Assim a atualização de uma crença, face a uma evidência (expressa em uma regra), independe da existência de evidências expressas em outras regras. Essa propriedade é conhecida como *modularidade* de regras. Recai sobre o autor da regra a responsabilidade de garantir tal independência.

O *fator de certeza* FC é a diferença entre a medida de crença MC e a medida de descrença MD, na hipótese H, dada a evidência E, ou seja:  $FC[H|E] = MC[H|E] - MD[H|E]$ , onde MC (valor entre 0 e 1) mede até que ponto a evidência E sustenta a hipótese H e MD (valor entre 0 e 1) mede até que ponto a evidência E sustenta a *negação* de H [BON 90]. Se MC for zero, a evidência não consegue sustentar H e se MD for zero, a evidência a sustenta. Portanto FC varia de -1 a 1. Informalmente o fator de crença pode ser visto como um *grau de confirmação* da hipótese H, dada a evidência E, mas na verdade ele representa uma mudança na crença em H, dado E. Assim um valor positivo corresponde a um aumento na crença em H e um valor negativo corresponde a uma diminuição nessa crença, e não uma medida absoluta de crença em H [HEC 86][BON 90]. No MYCIN, cada regra corresponde a uma evidência, embora essa evidência possa ser composta. Como qualquer evidência *sustenta* ou *nega* uma hipótese, mas não ambas as coisas, basta apenas um único número, por regra, para definir o FC. Se a regra sustenta uma hipótese, esse número representa a MC e o FC, pois a MD é nula. Se sustenta a negação da hipótese, então esse valor representa a MD e o FC, sendo a MC nula. Os FC das regras do MYCIN são fornecidos pelos especialistas que escrevem as regras. Eles refletem as avaliações deles da força com que as evidências sustentam as hipóteses codificadas em cada regra. Quando o sistema raciocina, visando identificar um organismo e prescrever uma terapia, os FC são combinados para refletir a aplicação das várias evidências e das várias regras aplicadas à situação específica.

Independente da forma de combinação dos FC, as funções de combinação do MYCIN devem satisfazer as seguintes propriedades [SHO 75][HEC 86][RIC 93]:

- FC depende somente da hipótese H e da evidência E, não dependendo de outras evidências e já conhecidas; assim  $FC(H,E,e) = FC(H,E,0) \equiv FC(H,E)$ ;
- se  $FC(H|E_1) > FC(H|E_2)$  então  $FC(H|E_1,E_3) > FC(H|E_2,E_3)$ ;
- ser *comutativas* e *associativas*, pois a ordem de coleta de evidências é arbitrária;
- evidências adicionais de confirmação devem aumentar a MC e evidências adicionais de contestação devem aumentar a MD, até que a certeza seja alcançada (valor limite 1);
- se inferências incertas forem encadeadas juntas, o resultado deverá ser menos certo do que cada inferência em separado.
- se  $FC(H|E_1) = 1$  e  $FC(H|E_2) \neq -1$  então  $FC(H|E_1,E_2) = 1$ , i.e  $E_1$  prova H;

- se  $FC(H|E_1) = -1$  e  $FC(H|E_2) \neq 1$  então  $FC(H|E_1, E_2) = -1$ , i.é  $E_1$  refuta  $H$ ;
- se  $FC(H|E_1) = 1$  e  $FC(H|E_2) = -1$  então  $FC(H|E_1, E_2)$  é indefinida.

São as seguintes as fórmulas associadas às Figuras 9.2(a) a 9.2(c), respectivamente:

$$a) FC(H|S_1, S_2) = \begin{cases} FC(H|S_1) + FC(H|S_2) - FC(H|S_1) * FC(H|S_2), & \text{se } FC(H|S_1), FC(H|S_2) \geq 0, \\ \frac{FC(H|S_1) + FC(H|S_2)}{1 - \min\{|FC(H|S_1)|, |FC(H|S_2)|\}}, & \text{se } FC(H|S_1), FC(H|S_2) \text{ são} \\ & \text{de sinais opostos,} \\ FC(H|S_1) + FC(H|S_2) + FC(H|S_1) * FC(H|S_2), & \text{se } FC(H|S_1), FC(H|S_2) < 0 \end{cases}$$

Essa fórmula [HEC 86, p.169] expressa uma medida da variação da crença em uma hipótese  $H$  dadas duas observações de evidências independentes. A partir de MC e MD, o valor do FC pode ser calculado. Se várias fontes de evidências de corroboração da hipótese (regras  $S$  com a mesma conclusão  $H$ ) forem combinadas, o valor de FC aumentará. Se forem introduzidas evidências conflitantes (regras que negam a conclusão anterior), o valor de FC diminuirá.

Essa função é uma variação da função de combinação paralela da heurística *independência*, complicada pela presença de valores negativos [SHO 95, p.177].

$$b) \begin{aligned} FC(H_1, H_2|E) &= \min \{FC(H_1|E), FC(H_2|E)\} \\ FC(H_1 \vee H_2|E) &= \max \{FC(H_1|E), FC(H_2|E)\} \end{aligned}$$

Permite calcular o FC do *antecedente* de uma regra  $E$  que é composto por várias hipóteses  $H$  [SHO 75, p.270]. Essa função adota a premissa de dependência entre essas hipóteses, sendo idêntica a heurística *inclusão* [SHO 95, p.177].

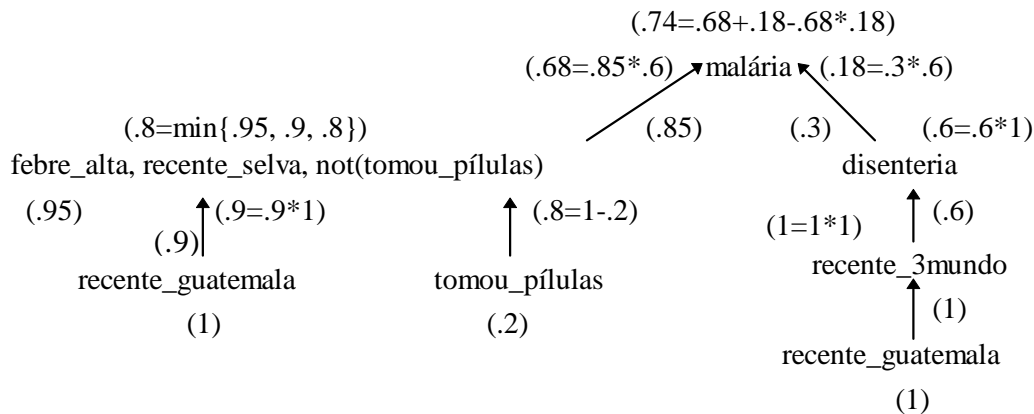
$$c) FC(H|S) = \begin{cases} FC(S'|S) * FC(H|S'), & \text{se } FC(S'|S) \geq 0 \\ -FC(S'|S) * FC(H|\neg S'), & \text{se } FC(S'|S) < 0 \end{cases}$$

$FC(H|S)$  é a variação da medida de crença em  $H$ , dada a evidência  $S$ . Essa evidência suporta  $S'$  que por sua vez suporta  $H$ , permite o encadeamento de uma regra com conclusão  $S'$  e premissas  $S$ , com outra cuja condição para a conclusão  $H$  é a evidência  $S'$ . Se  $S$  não suporta  $S'$  ( $FC(S'|S)$  negativo), então o fator de certeza de  $H$ , dado  $S$  depende do fator de certeza da  $H$ , dado que  $S'$  é *falsa*. Se esse valor ( $FC(H|\neg S')$ ) não estiver disponível, pode ser assumido *nulo* [HEC 86, p.170]. Caso se considere somente as regras com fator de certeza positivo, essa função de combinação torna-se a *combinação de regras* vista na abordagem heurística.

Uma *rede de inferência* é uma estrutura que mostra o relacionamento entre evidências e hipóteses. A Figura 9.5 mostra a rede de inferência da Figura 9.1, considerando FC os valores numéricos de certeza da abordagem heurística. Também é mostrada a propagação da medida de certeza utilizando fatores de crença. Se as regras não são independentes, resultados imprevisíveis poderão ocorrer, como em: *se* o esguicho do jardim (irrigador) estava ligado à noite *então* há evidência (0,9) de que a grama estará molhada hoje de manhã e *se* a grama do jardim está molhada hoje de manhã *então* há evidência (0,8) de que choveu ontem a noite [RIC 93, p.275]. Cada uma dessas regras, tomada isoladamente, pode descrever com precisão o “mundo”. Mas, como não são independentes, se tomadas em conjunto, aplicando-se a fórmula de combinação associada ao encadeamento ilustrado na Figura 9.2(c), tem-se:

$$\begin{aligned} FC(\text{grama molhada} | \text{irrigador}) &= 0,9 \\ FC(\text{chuva} | \text{irrigador}) &= 0,8 * 0,9 = 0,72 \end{aligned}$$

ou seja acredita-se, com fator de certeza 0,72, que choveu ontem a noite porque o irrigador da grama estava ligado!



**Figura 9.5** - Propagação de Fatores de Certeza

A situação ilustra o uso de estrutura de regra que quase sempre causa problemas e que deve ser evitado: relacionamento *causal* (irrigador causa grama molhada) seguido de um relacionamento de causalidade *inversa* (grama molha é causada pela chuva; sendo evidência de sua ocorrência). É possível derivar evidências de um sintoma a partir da sua causa e de uma causa a partir da observação do sintoma, mas é importante não usar novamente a evidência derivada no sentido inverso, sem novas informações. Para evitar esse problema, alguns sistemas limitam as suas regras a uma estrutura apenas ou separam os dois tipos para que as de um tipo não interfiram com as regras do outro tipo.

### 9.5.1 Vantagens e Desvantagens

Fatores de certeza produzem bons resultados em sistemas de regras que são puramente diagnósticos (MYCIN) ou puramente causal. Em outras circunstâncias, podem produzir variações de crença incorretas, com super valorização de evidências [RUS 95]. MYCIN obteve muito sucesso porque a sua base de regras foi construída de forma a refletir o comportamento do especialista na área. A sua performance não é muito sensível à variações nos fatores de certeza e a problemas que apresenta no relacionamento entre algumas regras [HEC 86]. Trata-se de um caso específico: o uso de fatores de certeza *não* é mais recomendado, em favor de abordagens que permitem explicitar as representações de dependência e independência em uma rede de inferências (p.ex. rede bayesiana).

Essa abordagem faz fortes suposições de independência condicional que facilitam o seu uso e, ao mesmo tempo, criam perigos se as regras não forem escritas com o cuidado de captar dependências importantes: deve ser evitada a causalidade *inversa*; os *antecedentes* de uma regra devem relacionar todas as condições dependentes; as regras devem ser independentes e a rede de inferências ter a forma de uma árvore. Portanto a base de regras deve possuir *modularidade* (o que permite considerar os relacionamentos antecedente/consequente individualmente, independentemente uns dos outros). Embora Heckerman [HEC 86] tenha proposto uma interpretação probabilística para fatores de certeza que antes eram estritamente intuitivos, a grande crítica a eles é que não possuem semântica definida [BHA 86]: somam todos os tipos de incerteza sobre a asserção a qual se referem, sem especificar qual a fonte de tal incerteza. Grosz [GRO 86] demonstrou que a abordagem de fatores de certeza, revisados por Heckerman, é isomórfica ao caso



especial da teoria de Dempster-Shafer com crença no intervalo de  $[-1, 1]$ . Não é possível distinguir entre ignorância e conflito (ambos representados por FC nulo) [BON 90].

## 10 Redes Bayesianas

Ao invés de supor independência dos eventos (como nos fatores de certeza do MYCIN), *redes bayesianas*<sup>13</sup> conservam o formalismo das estatísticas bayesianas. Como a maioria dos eventos no mundo real é condicionalmente independente da maioria dos outros, suas interações não precisam ser consideradas. Portanto pode-se usar uma representação mais local, na forma de uma *rede de restrições*, para descrever agrupamentos dos eventos que interagem entre si.

Devido a essa independência condicional, não é preciso representar as probabilidades de todas as combinações possíveis de eventos, na matriz de *probabilidades condicionais conjuntas*, resultando em problema mais tratável computacionalmente.

Redes bayesianas fornecem uma solução semântica para o problema de inter-relação de estruturas de *causalidade* (causas influenciam a probabilidade de seus sintomas) com as de *causalidade inversa* (observação de um sintoma afeta a probabilidade de todas as suas causas possíveis), discutido no Capítulo 9, fazendo uma distinção bastante clara entre esses tipos de influência, através da *rede de restrições*. Uma rede bayesiana representa as dependências entre variáveis e fornece uma representação concisa das suas distribuições de probabilidades conjuntas.

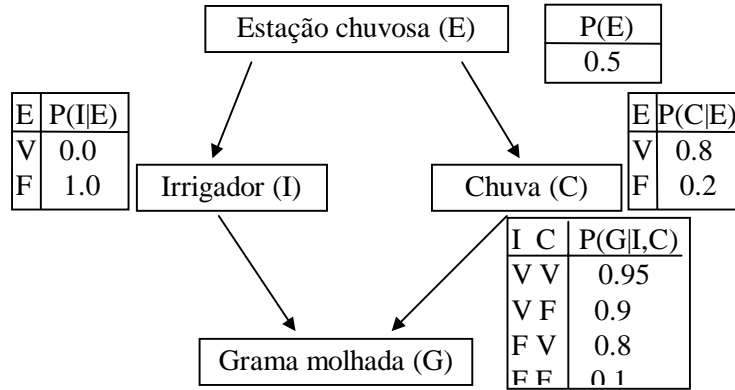
Russel e Norvig [RUS 95] definem **rede bayesiana** como um grafo em que: a) um conjunto de variáveis aleatórias, que podem ser *proposicionais* (valores *verdadeiro* ou *falso*) ou não (valores de tipos diversos), compõem os nós da rede; b) um conjunto de arcos direcionados conectam pares de nós e um arco de X para Y indica intuitivamente que X tem uma influência direta em Y (relações de *causalidade* entre as variáveis); c) cada nó X tem uma *tabela de probabilidade condicional* (CPT) que quantifica os efeitos que os nós pais (aqueles com arcos que chegam em X) têm sobre ele; d) o grafo não tem ciclos, é portanto um *grafo acíclico direcionado* (DAG).

Um rede não tem nós correspondendo a todas as possíveis causas de um evento. Alguns fatores não relevantes são resumidos na incerteza associada às probabilidades de algumas variáveis. Assim essas probabilidades resumem um conjunto potencialmente infinito de causas possíveis. Desta forma, com poucas variáveis aleatórias podemos lidar com um universo muito grande, ao menos aproximadamente. O grau de aproximação pode ser melhorado se nós introduzirmos informação adicional relevante.

O **procedimento geral para construção incremental** da rede bayesiana é: a) selecione o conjunto de variáveis relevantes  $X_i$  que descrevem o domínio; b) ordene esse conjunto, pelo *relacionamento causal*, começando com as causas principais, variáveis influenciadas pelas causas principais e assim por diante, até aquelas que não têm influência causal direta em outras variáveis; c) enquanto existirem variáveis no conjunto ordenado: c1) pegue uma variável  $X_i$  e acrescente um nó na rede para ela; c2) crie um arco de cada elemento de  $\text{pais}(X_i)$  para  $X_i$ , onde  $\text{pais}(X_i)$  é o conjunto mínimo de nós que já estão na rede e que têm uma influência causal direta sobre  $X_i$ ; c3) defina a tabela CPT para  $X_i$ .

<sup>13</sup> outros nomes: rede de crenças, rede probabilística, rede causal e mapa de conhecimentos [RUS 95].

A rede bayesiana (adaptada de [RIC 93]) para o exemplo irrigador-grama molhada-chuva (Capítulo 9), está mostrada na Figura 10.1. Esse grafo contém uma variável proposicional, *estação chuvosa*, que informa se é época das chuvas ou não. Observe a existência de mais de um caminho para o nó grama molhada.



**Figura 10.1** - Rede Bayesiana Típica

Um especialista pode decidir facilmente quais são as relações de dependência condicional direta em um certo domínio e montar a rede de restrições. Seja  $X = \{X_i\}$ , Desc( $X_i$ ) os descendentes de  $X_i$ , e  $Y$  um conjunto de variáveis aleatórias tal  $Y \subseteq X - \text{Desc}(X_i)$ . Pode ser provado que  $P(X_i | \text{Pais}(X_i) \cup Y) = P(X_i | \text{Pais}(X_i))$  [SHO 94]. Logo para especificar a distribuição conjunta de todas as variáveis da rede, é suficiente especificar a probabilidade de cada nó condicionado ao seus pais.

Do ponto de vista semântica, rede bayesiana pode ser vista como uma representação da distribuição de probabilidades conjunta das variáveis aleatórias  $X_i$  que compõem os seus nós ou como uma coleção de assertivas condicionalmente independentes.

Como representação da distribuição conjunta, a rede bayesiana provê uma descrição completa do domínio sendo representado. Qualquer valor da distribuição conjunta pode ser calculado a partir das informações na rede, como se segue:

$$P(X_1, \dots, X_n) = P(X_1 = x_1 \wedge \dots \wedge X_n = x_n) = \prod_{i=1}^n P(x_i | \text{Pais}(X_i)) \quad (10.1)$$

Essa equação implica um certo relacionamento de independência condicional dos nós, já mencionado, o qual vamos detalhar. Aplicando a definição de probabilidade condicional, repetidas vezes, temos:

$$\begin{aligned} P(X_1, \dots, X_n) &= P(X_n | X_{n-1}, \dots, X_1) * P(X_{n-1}, \dots, X_1) \\ P(X_1, \dots, X_n) &= P(X_n | X_{n-1}, \dots, X_1) * P(X_{n-1} | X_{n-2}, \dots, X_1) * \dots * P(X_2 | X_1) * P(X_1) \\ P(X_1, \dots, X_n) &= \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1), \end{aligned}$$

Comparando essa última equação com a Equação (10.1), resulta:

$$P(x_i | x_{i-1}, \dots, x_1) = P(x_i | \text{Pais}(X_i)) \quad (10.2)$$

onde  $\text{Pais}(X_i) \subseteq \{x_{i-1}, \dots, x_1\}$ , deve ser garantido na construção da rede.

Podemos concluir que a distribuição de probabilidades condicionais de  $X_i$ , dado as variáveis  $X_{i-1}, \dots, X_1$ , depende apenas dos nós-pais de  $X_i$ , sendo condicionalmente independente das outras variáveis.

A ação básica de um sistema de inferência probabilístico é calcular a distribuição de probabilidades *a posteriori* para um conjunto de variáveis, dados valores exatos para algumas variáveis de evidência. O enfoque semântico da rede bayesiana como uma coleção de assertivas condicionalmente independentes é útil na realização desse cálculo.

Por último, as redes bayesianas incorporam um mecanismo que permite o cálculo da influência de um nó do grafo de causalidades sobre qualquer outro nó. Esse mecanismo é capaz de garantir que as probabilidades vão ser calculadas corretamente, evitando ciclos como, por exemplo, fazer com que a grama molhada seja evidência de chuva, o que por sua vez passa a ser evidência para grama molhada e assim por diante.

De forma geral estamos interessados no cálculo de  $P(x_i \mid \text{evidência } S)$ . Se  $S \subseteq \text{Pais}(X_i)$ , então basta consultar a tabela CPT de  $X_i$ . A complexidade desse cálculo depende da estrutura da rede. Para  $S$  qualquer, existem três grandes classes de algoritmos para a execução dos cálculos citados: método de *triangulação de grupos* [Lauritzen e Spiegelhalter, 1988 Apud RIC 93], método de *transmissão de mensagens* [SHO 94][PEA 88] e *algoritmos estocásticos* [Chavez, 1989 Apud RIC 93].

Para tirar vantagem do fato de que os nós têm domínios restritos de influência, o método da *triangulação de grupos* introduz arcos explícitos entre pares de nós que têm um *descendente em comum*. Essa ligação dá suporte à avaliação do impacto da observação do evento associado a um nó sobre a hipótese associada ao outro nó. Isso é importante porque a ocorrência do descendente em comum poderia ser evidência de qualquer um dos dois nós ancestrais, mas a observação do descendente e de uma de suas causas não é evidência para a causa concorrente, porque já existe uma explicação alternativa para o fenômeno observado. Assim, no problema-exemplo irrigador-grama molhada-chuva, a grama molhada pode ser evidência de irrigador ligado *ou* de chuva, mas não de ambos.

O método de Pearl da *transmissão de mensagens* baseia-se em, face as independências condicionais, para calcular a probabilidade de um nó  $X$ , precisamos apenas conhecer:

- o total do suporte à hipótese associada ao nó  $X$  (causas possíveis),  $E_{X+}$ , que nele chega a partir de seus *nós-pais*, denominado mensagem  $\pi$  (variáveis de evidências ligadas a  $X$  através de seus pais);
- o total do suporte que chega em  $X$  (sintomas possíveis),  $E_{X-}$ , a partir de seus *nós-filhos*, denominado de mensagem  $\lambda$  (variáveis de evidências ligadas a  $X$  através de seus filhos);
- as entradas na tabela CPT que relacionam  $X$  com suas causas.

A estrutura da rede determina o método de propagação das mensagens  $\pi$  e  $\lambda$  a ser usado para a atualização das probabilidades nos nós. [SHO 94] apresenta algoritmo e programa Prolog para uma rede bayesiana com o formato de *árvore*, onde cada nó  $X$ , ao receber uma mensagem do nó  $N$  (seu pai ou um dos seus filhos), e sendo ela diferente da última recebida de  $N$ , atualiza sua probabilidades condicional (intensidade de crença) e envia uma mensagem para o seu pai e seus filhos, exceto  $N$ . Quando esse processo termina, a rede estabiliza com a distribuição de probabilidades correta, dada as evidências iniciais  $S$ .

A fórmula de cálculo da probabilidade para um nó  $X$  leva conta em que o conjunto de variáveis das evidências pode ser dividido na sub-árvore dos filhos de  $X$  e no restante da árvore. Esses conjuntos são condicionalmente independente, dado  $X$ . A fórmula é:

$$P(X | E_x^-, E_x^+) = P(E_x^- | X) * P(X | E_x^+) / \alpha = \lambda(X) * \pi(X) / \alpha, \text{ onde:}$$

$$\lambda(X) = \prod_{i=1}^n \lambda_{X_i}(X), \text{ sendo } \lambda_{X_i}(X) \text{ a mensagem recebida por } X, \text{ do seu filho } X_i;$$

$$\pi(X) = \sum_u P(X | u) \pi_X(u), \text{ sendo } \pi_X(u) \text{ a mensagem recebida por } X, \text{ do seu pai } u;$$

$P(X | u) = P(X | U=u)$  é retirada diretamente da tabela CPT de  $X$ , e

$$\alpha = \sum_x P(E_x^- | x) * P(x | E_x^+) \text{ é uma constante de normalização para garantir soma 1.}$$

Após esse cálculo, o nó  $X$  envia as mensagens  $\lambda_X(U)$  para o seu pai  $U$  e  $\pi_{X_j}(X)$  para cada um dos seus filhos  $X_j$ , sendo:

$$\lambda_X(U) = \sum_x \lambda(x) * P(x | U) \text{ e } \pi_{X_j}(X) = \beta * \left( \prod_{i \neq j} \lambda_{X_i}(X) \right) * \pi(X),$$

onde  $\beta$  também é uma constante de normalização. Cada mensagem é composta por um *par* de probabilidades condicionais: a probabilidade do fato associado ao nó em questão, dado que as condições (evidências) são verdadeiras e essa mesma probabilidade dado que as condições são falsas. Para indicar que um fato (representado pela *folha*  $X$  na árvore) ocorre, utilizamos a mensagem  $\lambda(X)=(1.0,0.0)$ ; para indicar que o fato não ocorre, é utilizado  $\lambda(X)=(0.0,1.0)$ . Se nada sabemos a respeito do fato,  $\lambda(X)=(1.0,1.0)$ . O nó raiz tem um tratamento especial, ao dos nós folhas: é propagada a mensagem  $(\pi(\text{raiz}), \pi(\text{raiz}))$ , onde  $\pi(\text{raiz})$  é a probabilidade *a priori* do evento associado ao nó raiz acontecer.

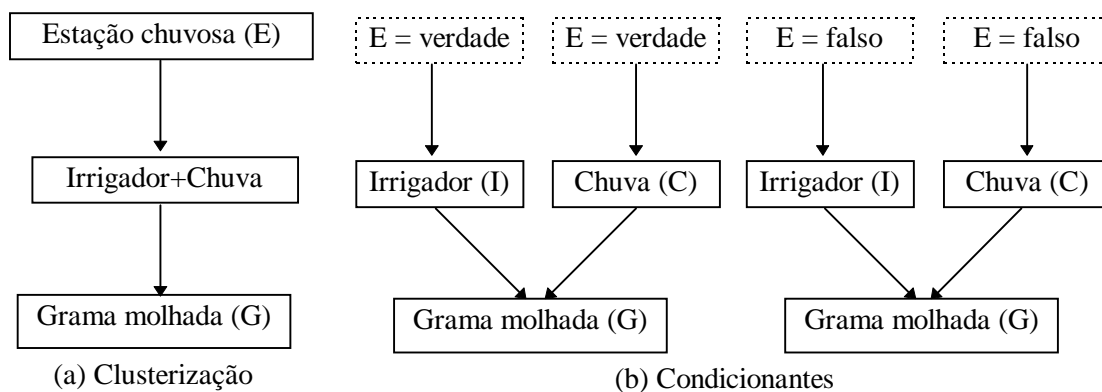
Já [RUS 95] apresenta um algoritmo com encadeamento para trás (*backward-chaining*), baseado na técnica de transmissão de mensagens de Pearl. Trata-se de algoritmo para rede bayesiana com formato de *polytree* (DAG conectado simplesmente) ou seja uma rede na qual cada nó pode ter *mais* de um pai, mas entre dois nós quaisquer existe, no máximo, um caminho direcionado. Ao contrário do método de transmissão de mensagens, esse algoritmo calcula a distribuição de probabilidade para uma variável apenas (aquela a qual está associada a pergunta feita ao sistema) e não para toda a rede. Russel e Norvig também sugerem abordagens para redes conectadas multiplamente (onde dois nós podem ser ligados por mais de dois caminhos direcionados) (Figura 10.1).

Os *algoritmos estocásticos* são mais rápidos que os métodos anteriores, mas na prática podem não produzir resultados corretos. Chavez [Chavez, 1989 Apud RIC 93] propôs um algoritmo que transforma uma rede bayesiana em uma cadeia de Markov. Uma *cadeia de Markov* é um processo estocástico sem *memória*, i.e a probabilidade de transição de estado depende apenas do estado atual e independe de toda a história passada (transições de estado que já ocorreram e que conduziram ao estado atual). Essa característica da cadeia de Markov reduz substancialmente a dimensão da matriz de probabilidades condicionais conjuntas, a qual passa a ser denominada *matriz de transição de estados*.

DAG multiconectados são aqueles nos quais dois nós podem ser conectados por mais de um caminho), p.ex. aqueles nos quais existem duas ou mais causas possíveis para uma mesma variável e as causas dividem um ancestral como [RUS 95], como a grama molhada no grafo representado na Figura 10.1.

Existem três classes básicas de algoritmos para avaliar DAG multiconectados. Todas elas utilizam como subrotinas algoritmos de avaliação de DAG com estrutura de árvores ou *polytrees*. Essas classes de algoritmos são compostas de métodos de:

- *clusterização* que através da substituição de um conjunto de nós por um único nó resultante, transformam a rede de conexões em uma *polytree* equivalente probabilisticamente, mas com topologia diferente. A CPT do nó resultante é definida para o produto cartesiano dos domínios das variáveis agregadas. Por exemplo, podemos transformar o DAG da Figura 10.1 em uma *polytree* fazendo a clusterização das variáveis I (irrigador) e C(chuva), obtendo I+C. Essas variáveis são proposicionais (i.é assumem apenas valores falso-F ou verdadeiro-V), a I+C assume os valores VV, VF, FV e FF (Figura 10.2(a)).
- *condicionantes* que instanciam variáveis (denotadas conjunto de corte) com valores definidos, obtendo uma *polytree* para cada instanciação possível, que então são avaliadas.  $P(x_i \mid \text{evidência } S)$  é calculada como a média ponderada sobre os valores obtidos em cada *polytree*. Para o DAG da Figura 10.1, o conjunto de corte é composto apenas pela variável E(Estação chuvosa), que após instanciação resulta nas duas *polytrees* da Figura 10.2(b).
- *simulação estocástica* (ou amostragem lógica) simula repetidas vezes o domínio descrito pela rede bayesiana e estima probabilidades contando a frequência dos eventos de interesse. Cada simulação começa com um sorteio aleatório dos valores das variáveis de evidência, considerando a sua probabilidade *a priori*. Então são feitos sorteios aleatórios para os nós sucessores, usando as suas CPT, e assim por diante.



**Figura 10.2** - Exemplo de Clusterização e Condicionantes de Redes Bayesianas

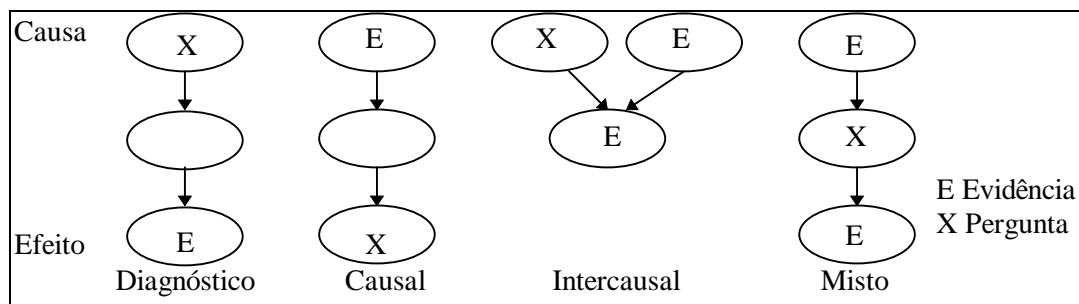
### 10.1 Vantagens e Desvantagens

Rede bayesiana é uma abordagem probabilística que tira partido da independência condicional de eventos e provê uma semântica bem definida para as medidas de certeza. A rede obtida constitui uma representação completa e não redundante do domínio e, muitas vezes, bem mais compacta do que a distribuição de probabilidades conjunta completa das variáveis aleatórias envolvidas no problema. Essa propriedade permite a manipulação de um grande número de evidências sem o crescimento exponencial asso-

ciado aos valores da distribuição de probabilidades condicional. Os algoritmos disponíveis são para *variáveis aleatórias discretas*. A extensão de redes bayesianas para manipular variáveis aleatórias contínuas constitui uma área de pesquisa atual.

A inferência em uma rede bayesiana depende da estrutura da rede. Para redes *polytrees* o tempo de computação é linear com o tamanho da rede. Para redes mais complexas, são necessárias técnicas mais elaboradas e o tempo pode se tornar exponencial no pior caso. As técnicas de simulação independem da estrutura da rede e produzem uma estimativa aproximada das probabilidades com menos computação, mas podem apresentar problemas na estimação de probabilidades de eventos que *raramente* ocorrem.

Redes bayesianas é o único mecanismo de raciocínio com incerteza que pode desenvolver todos os seguintes tipos de inferência: *diagnóstico* (de efeitos para as causas), *causal* (das causas para os efeitos), *intercausal* (discriminar entre causas de um efeito comum) e *misto* (combinação de dois ou mais tipos acima) [RUS 95] (Figura 10.3).



**Figura 10.3** - Padrões de Inferências em Redes Bayesianas

Redes bayesianas podem ser utilizadas a) no processo de tomada de decisão com base nas probabilidades da rede; b) para decidir quais variáveis de evidências adicionais devem ser observadas para obter informação útil; c) realizar análise de sensibilidade para entender quais aspectos do modelo têm o maior impacto nas probabilidades das variáveis de interesse (aquelas associadas às perguntas do usuário) e, portanto devem ter suas probabilidades estimadas com maior precisão; e d) explicar os resultados das inferências probabilísticas para o usuário.

Russel e Norvig [RUS 95] apresentam o sistema *Pathfinder* como um caso de estudo. *Pathfinder* é um sistema especialista para doenças do sistema linfático, desenvolvido por membros do programa de Ciência da Computação Médica da Universidade de Stanford. O sistema lida com mais de 60 doenças e com mais de 100 evidências (sintomas e resultados de testes). Foram desenvolvidas quatro versões, mostrando uma tendência a aumentar a sofisticação no raciocínio com incerteza.

*Pathfinder* I era baseado em regras e não tratava incerteza. *Pathfinder* II teste diversos métodos para lidar com incerteza, dentre eles fatores de certeza e teoria de Dempster-Shafer. Os resultados mostraram que um modelo Bayesiano simplificado (onde as evidências eram consideradas independentes) suplantava a performance dos anteriores. *Pathfinder* III usou o mesmo modelo Bayesiano simplificado, mas com um refinamento das probabilidades e maior atenção para com os eventos de baixa probabilidade. Já *Pathfinder* IV usou uma rede bayesiana para representar as dependências, eliminando a suposição de evidências independentes. Em uma avaliação inicial, utilizando 53 casos de

dificuldade acima da média, um grupo de especialistas atribuiu média 7,9 a versão III e 8,9 a versão IV, em um total de 10. Uma comparação recente mostra que a versão IV está atualmente suplantando a performance dos especialistas mundiais na área.



## 11 Teoria de Dempster-Shafer

As técnicas de fatores de certeza e redes bayesianas consideram proposições isoladas e atribuem a cada uma delas uma estimativa de ponto (i.é um único valor) do grau de crença garantido, face as evidências observadas.

A Teoria de Dempster-Shafer [Dempster, 1968; Shafer, 1976 Apud FUN 86 e RIC 93] considera grupos de proposições e atribui a cada um deles um intervalo [Crença, Plausibilidade] no qual está o grau de crença. Ela foi projetada para lidar com a distinção entre incerteza e ignorância [RUS 95], sendo que permite representar a *ignorância* sobre uma proposição. Ao invés de calcular a probabilidade de uma proposição, ela calcula a probabilidade das evidências suportarem a proposição.

A *crença* (representada por  $Cr$ ) mede a força da evidência em favor de um grupo de proposições, variando de 0 (zero), indicando falta de evidências, a 1 (um), denotando certeza.

A *plausibilidade* (representada por  $Pl$ ) é definida como  $Pl(s) = 1 - Cr(\neg s)$ , onde  $s$  é um grupo de proposições.  $Pl(s)$  varia de 0 a 1 e mede até que ponto a evidência em favor de  $\neg s$  deixa espaço para a crença em  $s$ . Assim, se existem evidências certas em favor de  $\neg s$ , então  $Cr(\neg s) = 1$  e, conseqüentemente  $Pl(s) = 0$ . Logo  $Cr(s)$  também será nula, pois está limitada por zero (que é o valor da sua plausibilidade).

O *intervalo crença-plausibilidade* mede o nível de crença em certas suposições e também a quantidade de informação que se possui. Ao analisar hipóteses concorrente, se não se tem nenhuma informação disponível (ignorância total), representa-se a situação dizendo que, para cada uma delas, as possibilidades reais estão no intervalo  $[0,1]$ . Quando as evidências começarem a se acumular, é de se esperar que esse intervalo diminua, representando o aumento de confiança associado ao fato de se saber a probabilidade de cada hipótese.

Pela Teoria de Dempster-Shafer, começa-se com um universo exaustivo de hipóteses mutuamente exclusivas (assim como no teorema de Bayes), representado por  $\theta$  e denotado *estrutura de discernimento*. A meta é anexar alguma medida de crença aos elementos de  $\theta$ .

Em geral, as evidências não sustentam diretamente elementos isolados mas sim grupos de elementos (subconjuntos de  $\theta$ ). Como os elementos de  $\theta$  são mutuamente exclusivos, as evidências em favor de alguns podem afetar a crença nos outros. Em um sistema puramente Bayesiano esses fenômenos são contornados listando-se todas as combinações de probabilidades condicionais conjuntas. A Teoria de Dempster-Shafer não atua dessa forma. Ela permite controlar as interações manipulando diretamente conjuntos de hipóteses (subconjuntos de  $\theta$ ), através da manipulação da massa básica de crença.

A *massa básica de crença*, denotada  $m$ , é definida para o conjunto das partes de  $\theta$ , ou seja o conjunto dos seus subconjuntos. A quantidade  $m(p)$  mede a quantidade de crença corretamente atribuída exatamente ao conjunto  $p$  de hipóteses, sendo que  $m(\phi)=0$ , onde

$\emptyset$  é o conjunto vazio. A soma de todos os valores  $m$  atribuídos aos subconjuntos de  $\theta$  deve ser 1 (um). Se  $\theta$  possui  $n$  elementos, então o seu conjunto das partes conterá  $2^n$  elementos. Em geral não é necessário lidar com todos, porque muitos deles nunca terão de ser considerados, visto que não têm significado para o domínio do problema; logo possuem  $m$  nulo.

A quantidade  $m(p)$  mede a quantidade de crença corretamente atribuída exatamente ao conjunto  $p$  de hipóteses. Pode ser vista como uma medida da probabilidade do conjunto de hipóteses  $p$  ser verdadeiro. Já a crença em  $p$  é dada pela fórmula [SIL 94, FUN 86]:

$$Cr(p) = \sum (m(q) \mid q \subseteq p),$$

representando a crença global da resposta correta estar em algum lugar do conjunto  $p$ .

A plausibilidade de  $p$  é dada por [SIL 94, BHA 86, FUN 86]:

$$Pl(p) = 1 - Cr(\neg p) = \sum (m(q) \mid q \subseteq \theta) - \sum (m(q) \mid q \subseteq \theta - p) = \sum (m(q) \mid p \cap q \neq \emptyset).$$

O intervalo  $[Cr(p), Pl(p)]$  pode ser interpretado com limites inferior e superior para a probabilidade de  $p$  ser verdade, i.é  $P(p)$  [RUS 95, SIL 94, FUN 86].

O problema a seguir, retirado de [RIC 93], exemplifica um diagnóstico simplificado feito com base em raciocínio estatístico baseado na Teoria de Dempster-Shafer.

Suponha que se deseje diagnosticar a doença de um paciente, que com certeza tem alergia (A), gripe (G), resfriado (R) ou pneumonia (P), admitidas exaustivas e mutuamente exclusivas. Portanto  $\theta = \{A, G, R, P\}$ .

Se não se tem nenhuma informação sobre como escolher entre as hipóteses quando se começa o diagnóstico,  $m$  pode ser definido como:  $m(\theta) = 1$ . Essa atribuição representa que, no momento não se dispõe de nenhuma informação que permita atribuir crença a qualquer um dos elementos de  $\theta$ , ou seja situação de ignorância total.

O aparecimento de febre sugere (em um nível estimado pelo médico em 0,6) que o diagnóstico correto está no conjunto  $\{G, R, P\}$ . Então  $m$  deve ser atualizada da seguinte forma:

$$m(\{G, R, P\}) = 0,6 \qquad m(\theta) = 0,4.$$

Ao  $\{G, R, P\}$  foi atribuída uma crença apropriada; o restante da crença foi atribuída ao conjunto maior  $\theta$ , sem o comprometimento de atribuí-la ao complemento de  $\{G, R, P\}$ .

Outras evidências que surjam podem resultar em novos valores para  $m$ . Assim é necessário dispor de uma maneira para combinar as diversas massas básicas de crenças que surgem de diversas fontes de evidência.

A *regra de combinação de Dempster* permite combinar quaisquer duas funções de massa básica de crença  $m_1$  e  $m_2$ , resultando na função de massa básica de crença  $m_3$ , quer  $m_1$  e  $m_2$  representem várias fontes de evidência para uma única hipótese ou várias fontes de evidência para diferentes hipóteses. Essa regra cobre as formas de combinação ilustradas nas Figuras 9.2(a) e 9.2(c). O recurso de combinar massas básicas de crenças associadas às várias hipóteses dependentes, consideradas em conjunto, (conforme Figura 9.2(b)), já está presente na capacidade do especialista de atribuir um valor de  $m$  ao grupo de hipóteses.

Seja  $X$  o grupo de subconjuntos de  $\theta$ , aos quais  $m_1$  atribui um valor não nulo, e  $Y$  o grupo correspondente de  $m_2$ . A regra de combinação de Dempster é dada por:

$$m_3(Z) = \frac{\sum_{Z=X \cap Y} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)},$$

onde  $\emptyset$  é o conjunto vazio. Essa fórmula garante que a soma de todos os valores de  $m_3(Z)$  seja igual a um, desde que essa restrição também seja verdade para os valores de  $m_1$  e de  $m_2$ .

Suponha que  $m_1$  represente a função de massa básica de crença após ter sido observada a presença de febre e que  $m_2$  represente a função de massa básica de crença após ter sido observado um nariz escorrendo, conforme a seguir:

$$\begin{aligned} m_1(\{G, R, P\}) &= 0,6 & m_2(\{A, G, R\}) &= 0,8 \\ m_1(\theta) &= 0,4 & m_2(\theta) &= 0,2 \end{aligned}$$

A função combinada de massa básica de crença,  $m_3$ , após a observação de febre e nariz escorrendo está mostrada na Tabela 11.1.

**Tabela 11.1** - Combinação Simples de Dempster

$m_3 = m_1 \cap m_2$		$\{A, G, R\}$	0,8	$\theta$	0,2
$\{G, R, P\}$	0,6	$\{G, R\}$	0,48	$\{G, R, P\}$	0,12
$\theta$	0,4	$\{A, G, R\}$	0,32	$\theta$	0,08

Observe que, no exemplo acima, não houve interseção vazia de  $X$  e  $Y$ . Assim o denominador da fórmula de Dempster assumiu valor 1. Também não houve mais de uma ocorrência do mesmo elemento  $Z$ . No caso da interseção resultar em elementos repetidos, os valores das suas massas básicas de crenças devem ser somados, como indica o somatório no numerador da fórmula de Dempster.

Se resultar da operação de interseção algum subconjunto vazio, valores de  $m_3$  serão atribuídos a eles. Como  $\theta$  foi assumido exaustivo, o valor real da hipótese precisa estar contido em algum dos seus subconjuntos não vazios. Assim, é preciso redistribuir qualquer massa de crença dos subconjuntos vazios, proporcionalmente entre os não vazios, utilizando o denominador da fórmula de Dempster como fator de escalamento.

Suponha que, motivado pelo fato de se saber que o problema do paciente desaparece quando ele viaja, a função de massa básica de crença passe a ser  $m_4$ , como abaixo:

$$m_4(\{A\}) = 0,9 \quad m_4(\theta) = 0,1$$

Combinando as funções de massas básicas de crenças associadas as evidências febre, nariz escorrendo e alergia, através da aplicação do numerador da fórmula de Dempster, obtém-se os dados apresentados na Tabela 11.2:

**Tabela 11.2** - Exemplo de Combinação de Dempster com Conjunto Vazio

$m_5 = m_3 \cap m_4$		$\{A\}$	0,9	$\theta$	0,1
$\{G, R\}$	0,48	$\phi$	0,432	$\{G, R\}$	0,048
$\{A, G, R\}$	0,32	$\{A\}$	0,288	$\{A, G, R\}$	0,032
$\{G, R, P\}$	0,12	$\phi$	0,108	$\{G, R, P\}$	0,012
$\theta$	0,08	$\{A\}$	0,072	$\theta$	0,008

A massa básica de crença associada ao conjunto vazio é de 0,54; sendo apenas 0,46 associado a eventos realmente possíveis. Para manter a soma de  $m_5$  unitária, os valores associados aos resultados possíveis devem ser escalonados. O evento alergia foi derivado mais de uma vez; logo as massas básicas de crenças devem ser somadas.

A Tabela 11.3 mostra a massa básica de crença, a crença e a plausibilidade resultantes, após aplicação do fator de escalamento 0,46 (denominador da fórmula de Dempster).

**Tabela 11.3** - Intervalo de Crença e Plausibilidade

Eventos	A	G,R	A,G,R	G,R,P	$\theta$
$m_5$	0,783	0,104	0,070	0,026	0,017
Cr	0,783	0,104	0,957	0,130	1,000
Pl	0,870	0,217	1,000	0,217	1,000

### 11.1 Vantagens e Desvantagens

A teoria de Dempster-Shafer permitir representar a ignorância e distingui-la da incerteza. A cada evidência está associado um intervalo  $[Cr(p), Pl(p)]$  para a proposição  $p$ . O tamanho de  $[Cr(p), Pl(p)]$  ajuda a decidir se precisamos de mais evidências ou não. Se o intervalo for grande, então precisamos de mais evidências para podermos avaliar a crença em  $p$ . Se os intervalos, originados por duas evidências distintas, possuem interseção vazia, então as evidências são inconsistentes [GRO86]. Usando  $Cr(p)$  e  $Pl(p)$ , vários níveis de incerteza sobre um conhecimento podem ser representados [BHA 86]: a) certo e preciso se  $Cr(p) = Pl(p) = 1$ ,  $m(q \neq p) = 0$  e  $p$  é um conjunto unitário distinto de  $\theta$ ; b) certo e impreciso se, no caso anterior,  $p$  não é um conjunto unitário; c) consonante (harmônico) se os valores de  $m$  são tais que os elementos  $p$  com  $m(p) \neq 0$  formam uma sequência encadeada  $p_1 \subseteq p_2 \subseteq \dots \subseteq p_n$ , toda a evidência é consistente e com mais evidência podemos esperar obter  $p_i$  com menos elementos; d) inconsistente se os elementos  $p$  com  $m(p) \neq 0$  são subconjuntos que podem ser divididos em pares disjuntos; se são conjuntos unitários então a situação é a mesma de ter-se uma distribuição de probabilidades sobre  $\theta$ , sendo a evidência inconsistente, mas precisa; e) ignorância total se a evidência é tal que  $m(\theta)=1$  e  $m(p)=0$  para  $p \neq \theta$ , logo a  $Cr(p) = 0$  e  $Pl(p) = 1$ . Sob a hipótese de independência condicional o caso especial de intervalo de um único ponto de Dempster-Shafer é idêntico a probabilidade condicional bayesiana.

A grande vantagem da teoria de Dempster-Shafer é permitir a representação explícita de ignorância e sua maior desvantagem é possuir pouco ou quase nenhum mecanismo para realizar raciocínio ou inferência com informações representadas na forma de crença [BRA 86]. Normalmente a evidência disponível é de natureza qualitativa e a sua conversão em valores numéricos (função  $m$ ) é subjetiva e reflete o conhecimento do domínio pelo analista que os fornece.

## 12 Técnicas Baseadas em Conjuntos Fuzzy

Para Bonissone [BON 86, 87 e 90] a abordagem simbólica é inadequada para representar e resumir medidas de incerteza. A numérica (i.é os formalismos numéricos vistos até o momento) impõe restrições sobre a estrutura da informação (p.ex. independência condicional das evidências e/ou exclusão mútua de hipóteses exaustivas) e exige que o usuário ou o especialista forneça uma estimativa precisa e consistente da incerteza dos dados atômicos e de suas relações, o que é irreal. Neste capítulo são descritos sistema fuzzy, teoria da possibilidade de Zadeh e o uso de probabilidades lingüísticas conjugado com normas triangulares para combinar e resumir medidas de incerteza. Eles são adequados para o tratamento de informação incerta e vaga, em especial a de origem lingüística, sem exigir do especialista a elicitação de precisas medidas numéricas de incerteza.

Sistema fuzzy é orientado para manipular dados vagos através da associação de um grau de pertinência gradual (entre 0 e 1) a um conjunto fuzzy (i.é com fronteiras mal definidas). A teoria da possibilidade associa a uma variável com valor incerto uma distribuição de possibilidades (i.é uma função de pertinência), a qual mede até que ponto é *possível* que a variável em questão assuma um certo valor (que é considerado o seu valor verdadeiro), sendo os valores supostos mutuamente exclusivos. A partir da distribuição de possibilidades, a verossimilhança de eventos pode ser descrita utilizando as medidas de possibilidade e de necessidade [BIT 96]. Para Zadeh [ZAD 78, p.3] “a imprecisão intrínseca na linguagem natural é, principalmente mais possibilística do que probabilística”.

Bonissone propõe o uso de probabilidades lingüísticas com semântica fuzzy e de normas triangulares para combinar e resumir as medidas de incerteza.

### 12.1 Sistema Fuzzy

A teoria dos conjuntos fuzzy foi proposta por Lofti Zadeh, em 1965<sup>14</sup> e serve de base para sistema fuzzy, formalismo mais tradicional para tratamento da informação imprecisa e vaga [BIT 96]. O uso da lógica fuzzy em IA normalmente se dá através da agregação de variáveis fuzzy a sistemas baseados em conhecimento (em especial a sistemas de regras), permitindo o gerenciamento de incerteza com a realização de raciocínio *aproximado qualitativo*. Este raciocínio lida com proposições fuzzy (conhecimento expresso por primitivas codificadas em uma forma de linguagem natural) e regras de inferências. Diversas regras fuzzy podem disparar simultaneamente, sendo necessário um mecanismo para agregar os resultados. Lógica fuzzy é utilizada na engenharia para a construção de FKBC - Fuzzy Knowledge Based Controllers. Em ambos os casos uma variável lingüística é modelada por uma variável fuzzy (através de uma função de pertinência). Os valores numéricos (da função de pertinência) são utilizados para disparar as regras fuzzy e na agregação dos resultados obtidos. No caso de FKBC, o valor agregado é utilizado para uma ação de controle em um sistema dinâmico. Em alguns casos, para facilitar o

---

<sup>14</sup>Zadeh, Lofti A. Fuzzy Sets. In: **Information and Control**. v.8, p.338-353

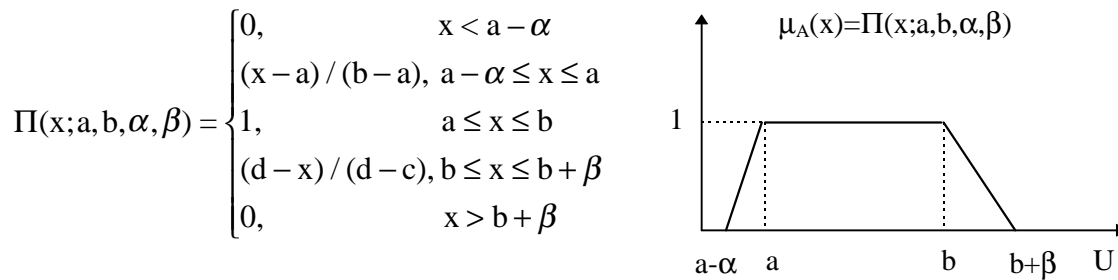
entendimento do usuário, pode ser necessário transformar o valor numérico agregado em uma variável lingüística, através de um processo chamado de “desfuzzificação”.

Lógica fuzzy (também denotada lógica nebulosa ou difusa) é uma lógica multi-valorada baseada no conceito de função de pertinência, operando com base em “propriedades” de elementos de um conjunto, sendo *qualitativa*. A teoria de probabilidade trata *aleatoriedade* (com base na teoria dos conjuntos clássicos e na lógica booleana) e sistema fuzzy trata a *imprecisão* relacionada com a pertinência a um conjunto (com base na teoria dos conjuntos fuzzy e na lógica fuzzy). Ao contrário da teoria de probabilidade, o arcabouço fuzzy dispõe de facilidades para representar o significado de proposições contendo:

- predicados fuzzy: pequeno, grande, jovem, ...
- quantificadores fuzzy: a maioria, vários, freqüentemente, poucos, ...
- probabilidade fuzzy: possível, bem possível, quase possível, ...
- valores “verdade” fuzzy: muito verdadeiro, pouco verdadeiro, ...
- modificadores de predicados: muito, ligeiramente, etc.

### 12.1.1 Teoria dos Conjuntos Fuzzy

Seja  $U$  o universo de discurso e o subconjunto  $A \subseteq U$ . A *função de pertinência*  $\mu_A$  associa a cada elemento  $x \in U$  um valor  $\mu_A(x)$  no intervalo  $[0,1]$ , que representa o grau de pertinência parcial de  $x$  ao subconjunto  $A$ , denotado *conjunto fuzzy*. Um conjunto fuzzy pode ser definido a) pela enumeração de todos os seus elementos e respectivos graus de pertinência (se o conjunto for finito); b) por uma função aritmética que represente a sua função de pertinência; e c) por funções especiais (formadas por semi-retas ou funções contínuas diferenciáveis) que expressam noções lingüísticas. A função trapézio é a função de semi-retas mais utilizada (Figura 12.1).



**Figura 12.1** - Função Trapézio

O *suporte* de  $A$  é definido por  $S(A) = \{x \in U | \mu_A(x) \neq 0\}$  e sua *altura* por  $\text{alt}(A) = \sup(\mu_A(x))$ , para  $x \in U$ , onde  $\sup$  é o supremo.  $A$  é *normal* se  $\text{alt}(A) = 1$ ; caso contrário  $A$  é *subnormal*. Os conjuntos fuzzy normais estão associados às noções lingüísticas e os conjuntos subnormais aos resultados de operações sobre conjuntos fuzzy.

$A$  é *convexo* sse para  $\forall (x,y) \in U$  e  $\forall \lambda \in [0,1]$ ,  $\mu_A(\lambda x + (1-\lambda)y) \geq \min(\mu_A(x), \mu_A(y))$ , ou seja o grau de pertinência da combinação linear de dois elementos  $x$  e  $y$  quaisquer de  $A$  for maior do que o menor dos graus de pertinência de  $x$  e  $y$ . Dois conjuntos fuzzy são *iguais* se têm o mesmo suporte e todo elemento do suporte tem o mesmo grau de pertinência.

As operações de interseção, união e complemento de conjuntos fuzzy permite a agregação de graus de pertinência e desempenham um papel importante no tratamento de incerteza (veja seção 12.3). Dubois e Prade [4 Apud BON 86] demonstraram que normas (T-normas) e conormas triangulares (T-conormas, também denotadas S-normas) são as famílias mais gerais de funções binárias satisfazem, respectivamente os requerimentos de operadores de conjunção (interseção) e disjunção (união) [BON 86, p.219]. As operações com conjuntos fuzzy utilizam T-normas e T-conormas aplicadas sobre graus de pertinência, para interseção e união respectivamente.

T-norma e T-conorma são funções binárias definidas em  $[0,1] \times [0,1]$  e associam valores em  $[0,1]$  e que são monotônicas, comutativas e associativas. Quando aplicadas aos extremos do intervalo  $[0,1]$  satisfazem a tabela verdade dos operadores lógicos AND e OR. A Tabela 12.1 apresenta os axiomas de T-normas e de T-conormas. Esses axiomas definem famílias de operadores, sendo que a Tabela 12.2 ilustra as principais.

**Tabela 12.1** - Axiomas de T-normas e de T-conormas

T-norma	T-conorma	Nome
$T(0,0) = 0$	$S(1,1) = 1$	Fronteira
$T(a,1) = T(1,a) = a$	$S(a,0) = T(0,a) = a$	Fronteira
$T(a,b) \leq T(c,d)$ se $a \leq c$ e $b \leq d$	$S(a,b) \leq S(c,d)$ se $a \leq c$ e $b \leq d$	Monotonicidade
$T(a,b) = T(b,a)$	$S(a,b) = S(b,a)$	Comutatividade
$T(a,T(b,c)) = T(T(a,b),c)$	$S(a,S(b,c)) = S(S(a,b),c)$	Associatividade

Fonte: Bonissone e Decker, 1986. Selecting Uncertainty Calculi and Granularity, p.220

Devido a associatividade é possível definir a T-norma para uma argumento n-dimensional, como:  $T(x_1, \dots, x_{n-1}, x_n) = T(T(x_1, \dots, x_{n-1}), x_n)$  e de forma análoga, a T-conorma.

**Tabela 12.2** - Principais T-normas e T-conormas

T-norma	T-conorma	Nome
$T_0(a,b) = \min(a,b)$ se $\max(a,b) = 1$ , = 0, caso contrário	$S_0(a,b) = \max(a,b)$ se $\min(a,b) = 0$ , = 0, caso contrário	Weber
$T_1(a,b) = \max(a+b-1, 0)$	$S_1(a,b) = \min(a+b, 1)$	Lukasiewicz
$T_{1.5}(a,b) = ab/[2-(a+b-ab)]$	$S_{1.5}(a,b) = (a+b)/(1+ab)$	Probabilista
$T_2(a,b) = ab$	$S_2(a,b) = a+b-ab$	
$T_{2.5}(a,b) = ab/(a+b-ab)$	$S_{2.5}(a,b) = (a+b-2ab)/(1-ab)$	Zadeh
$T_3(a,b) = \min(a,b)$	$S_3(a,b) = \max(a,b)$	

Fonte: Bonissone e Decker, 1986. Selecting Uncertainty Calculi and Granularity, p.221

Existe um conjunto de axiomas similares que definem uma família de operadores de negação [BON 86]. Para um operador de negação definido como  $N(x) = 1-x$ , as T-normas e T-conormas são *duais* no sentido de que satisfazem a seguinte generalização da lei de DeMorgan:  $S(a,b) = N(T(N(a), N(b)))$  e  $N(S(N(a), N(b)))$ . Esse relacionamento entre T-normas e T-conormas impede que sejam definidas de forma independente. Para o operador de negação tradicional  $N(x) = 1-x$ , as normas de mesmo subscrito da Tabela 12.2 são duais e formam pares denotados pares de DeMorgan. As normas acima obedecem a relação de ordem:  $T_0 \leq T_1 \leq T_{1.5} \leq T_2 \leq T_{2.5} \leq T_3$  e  $S_3 \leq S_{2.5} \leq S_2 \leq S_{1.5} \leq S_1 \leq S_0$ .

É muito comum a utilização das normas de Zadeh e da negação tradicional para implementar interseção, união e complementação de conjuntos fuzzy. Assim, para  $\forall x \in U$ ,  $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$ ;  $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$ ; e  $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$ .

Uma *relação* fuzzy  $R$  entre os conjuntos clássicos  $X$  e  $Y$  é um conjunto fuzzy definido no produto cartesiano  $X \times Y = \{(x,y) \mid x \in X \text{ e } y \in Y\}$  e caracterizado por uma função de pertinência  $\mu_R = X \times Y \rightarrow [0,1]$ . Por exemplo “é muito menor do que” é uma relação fuzzy em ‘ $x$  é muito menor do que  $y$ ’, onde  $x$  e  $y$  são números reais, cuja função de pertinência poderia ser caracterizada por:  $\mu_R(x,y) = 0$ , se  $x \geq y$ , e  $1/[1+(10/(y-x))^2]$ , se  $x < y$ . Este valor tende para 1 para  $y-x$  muito grande. Se  $R$  é uma relação fuzzy em  $X \times Y$  e  $S$  é outra relação fuzzy em  $Y \times Z$ , então a *composição* de  $R$  e  $S$ , denotada  $R \circ S$  é uma relação fuzzy em  $X \times Z$ , definida como:  $\mu_{R \circ S}(x,z) = S_i \{ \forall y, T_i(\mu_R(x,y), \mu_S(y,z)) \}$ , onde  $T_i$  e  $S_i$  são uma  $T$ -norma e uma  $T$ -conorma dual. Se  $T_i$  é o mínimo e  $S_i$  é o máximo, então  $R \circ S$  é a composição max-min. Por exemplo, para a relação max-min, seja  $\mu_R(x,y) = 0,4/x_1y_1 + 0,6/x_1y_2 + 0,9/x_1y_3 + 0,9/x_2y_1 + 1/x_2y_2 + 0,1/x_2y_3$  e  $\mu_S(y,z) = 0,5/y_1z_1 + 0,8/y_1z_2 + 0,1/y_2z_1 + 1/y_2z_2 + 0/y_3z_1 + 0,6/y_3z_2$ . Então a relação composta  $R \circ S$  é dada por  $\mu_{R \circ S}(x,z) = 0,4/x_1z_1 + 0,6/x_1z_2 + 0,5/x_2z_1 + 1/x_2z_2$ .

A extensão cilíndrica permitir a operação de conjuntos fuzzy com relações fuzzy. Assim estende conjuntos fuzzy em relações binárias fuzzy; relações binárias fuzzy em relações ternárias fuzzy, etc. Seja  $F$  um conjunto fuzzy definido em  $Y$ . A *extensão cilíndrica* de  $F$  em  $X \times Y$ , denotada  $ce(F)$  é o conjunto de todos os pares ordenados  $(x,y) \in X \times Y$ , com grau de pertinência  $\mu_F(y)$ , ou seja  $ce(F)(x,y) = \{\mu_F(y)/(x,y), (x,y) \in X \times Y\}$ . Se  $F$  for uma relação em  $U_1 \times \dots \times U_{m+1}$ , então sua extensão cilíndrica em  $U_1 \times \dots \times U_{m+1} \times \dots \times U_n$  é definida por:  $ce(F)(u_1, \dots, u_n) = \{\mu_F(u_1, \dots, u_{m+1})/(u_1, \dots, u_n), (u_1, \dots, u_n) \in U_1 \times \dots \times U_n\}$ . Por exemplo seja  $R$  um relação fuzzy em  $X \times Y$  e  $A$  um conjunto fuzzy em  $X$ . A interseção de  $A$  com  $R$ , é calculada como a interseção entre  $ce(A)$  e  $R$ . Supondo o uso da  $T$ -norma  $T_3(a,b) = \min(a,b)$  como operador de interseção, para  $A = 1/x_1 + 0,8/x_2 + 0,1/x_3$  e  $R = 0,8/x_1y_1 + 1/x_1y_2 + 0,1/x_1y_3 + 0,7/x_1y_4 + 0,8/x_2y_1 + 0,8/x_2y_2 + 0,9/x_3y_1 + 1/x_3y_2 + 0,7/x_3y_3 + 0,8/x_3y_4$  então  $A \cap R = 0,8/x_1y_1 + 1/x_1y_2 + 0,1/x_1y_3 + 0,7/x_1y_4 + 0,8/x_2y_1 + 0,8/x_2y_2 + 0,9/x_3y_1 + 1/x_3y_2 + 0,7/x_3y_3 + 0,8/x_3y_4$ . A operação de *projeção* reduz a dimensionalidade de uma relação fuzzy. Por exemplo, reduz uma relação binária a um conjunto fuzzy e um conjunto fuzzy a um único valor. Seja  $R$  uma relação em  $W \times Z$ , onde  $W = U_1 \times \dots \times U_m$  e  $Z = U_{m+1} \times \dots \times U_n$ . A projeção em  $Z$ ,  $\text{proj}_Z^R$  é dada por:  $\text{proj}_Z^R(u_{m+1}, \dots, u_n) = \{\sup_W \mu_R(w,z)/z, w \in W \text{ e } z \in Z\}$ . Se  $R$  é a relação binária acima,  $\text{proj}_X^R = 1/x_1 + 0,8/x_2 + 1/x_3$ ;  $\text{proj}_Y^R = 0,9/y_1 + 1/y_2 + 0,7/y_3 + 0,8/y_4$  e a projeção total de  $R = \sup_W \mu_R(w,z)/z = \sup_Z \mu_R(w,z)/w = \sup \mu_R(w,z)/(w,z) \in 1$ .

A *composição* de um conjunto fuzzy  $A$  (em  $X$ ) com uma relação binária fuzzy (em  $X \times Y$ ), resulta em um conjunto fuzzy  $B$  (em  $Y$ ), dado por:  $B = A \circ R = \text{proj}_Y^{(ce(A) \cap R)}$ . As composições mais comuns são a max-min,  $\mu_B(y) = \max_x [\min(\mu_A(x), \mu_R(x,y))]$  e a max-produto,  $\mu_B(y) = \max_x (\mu_A(x) \cdot \mu_R(x,y))$ . Essa composição de conjunto com relação fuzzy permite responder questões do seguinte tipo: a partir da relação  $X \times Y$  (cuja descrição lingüística é “Agnes é um pouco maior do que Olga”) e do conjunto  $A$  em  $X$  (“Agnes é um pouco alta”), estimar o conjunto fuzzy  $B$  em  $Y$  (“Qual a altura de Olga?”). Para o domínio (alturas)  $X = Y = \{172,5 \ 175 \ 177,5 \ 180\}$ , seja  $R = 0,4/(172,5;172,5) + 0,1/(172,5;175) + 0/(172,5;177,5) + 0/(172,5;180) + 0,7/(175;172,5) + 0,4/(175;175) + 0,1/(175;177,5) + 0/(175;180) + 1/(177,5;172,5) + 0,7/(177,5;175) + 0,4/(177,5;177,5) +$



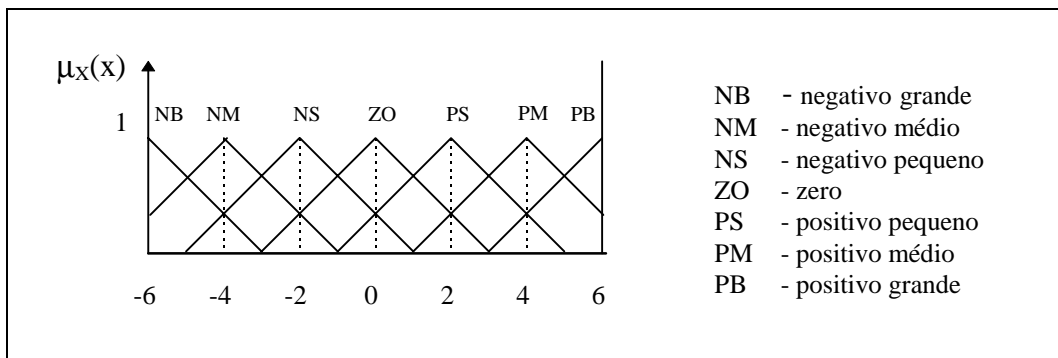
$0,1/(177,5;180) + 0,7/(180;172,5) + 1/(180;175) + 0,7/(180;177,5) + 0,4/(180;180)$  e  $A = 0,1/172,5 + 0,4/175 + 0,7/177,5 + 0,9/180$ . Então calculando B através da expressão  $\text{proj}^{(ce(A)\cap R)}_Y$  obtemos  $0,7/172,5 + 0,9/175 + 0,7/177,5 + 0,4/180$ , para  $\cap$  computada como a função mínimo.

O *produto cartesiano* de dois conjuntos fuzzy  $A \times B$  definidos, respectivamente nos universos de discurso U e V, é dado por  $\mu_{A \times B}(x,y) = \mu_A(x) \wedge \mu_B(y) / (x,y)$ , onde  $\wedge$  é o mínimo.

### 12.1.2 Raciocínio Aproximado

O raciocínio aproximado é baseado na lógica fuzzy: com base em proposições fuzzy e em regras de inferência (do tipo  $X \text{ é } A \rightarrow Y \text{ é } B$ ), dado um antecedente, avalia consequentes. Esse tipo de raciocínio lida com variáveis lingüísticas que representam conhecimento expresso em primitivas, por sua vez expressas em uma forma de linguagem natural. Por exemplo, “o erro tem valor negativo grande” pode ser representado simbolicamente por “E é NB”, (veja Figura 12.2), onde E representa uma variável lingüística e NB uma propriedade. “E é NB” é uma proposição fuzzy atômica por envolver apenas uma variável e um valor. Já “E é NB ou é NM” é uma proposição complexa.

Cada variável lingüística possui o seguinte framework associado:  $\langle X, LX, DX, M_X \rangle$  onde X é o nome simbólico da variável lingüística, LX é o conjunto dos valores lingüísticos de X, DX é o seu domínio físico real e  $M_X$  é a função semântica de X (i.é as funções de pertinência de LX). Essas variáveis estão associadas ao cálculo de incertezas em conjuntos fuzzy, sendo LX define a granularidade da incerteza. Normalmente trabalha-se com granularidade 5 ou 7 (Figura 12.2) e com o domínio  $[-6,6]$  através de uma normalização do domínio DX para esse intervalo (pressupõe a que a distribuição de DX é uniforme).



**Figura 12.2** - Normalização com Granularidade 7

Um número fuzzy N, representado por  $\tilde{N}$  ou  $N_i$ , é um conjunto fuzzy que expressa a noção lingüística “aproximadamente”. Assim “o erro é aproximadamente 4” pode ser expresso como “E é  $\tilde{4}$ ” ou “E é  $4_i$ ”. A função de pertinência de um número fuzzy tem a forma triangular. A notação  $N_i = (a_i, b_i, \alpha_i, \beta_i)$  é derivada da função trapézio. Com ela podemos representar um número x como  $(x, x, 0, 0)$ ; um intervalo  $[x, y]$  como  $(x, y, 0, 0)$ ; um número fuzzy  $a_i$  como  $(a_i, a_i, \alpha_i, \beta_i)$  ou um intervalo (conjunto) fuzzy como  $(a_i, b_i, \alpha_i, \beta_i)$ .

Uma declaração condicional fuzzy, *se X é A então Y é B*, onde A e B são proposições fuzzy pertencentes aos universos de discurso U e V, respectivamente, pode ser vista

como uma regra  $X \text{ é } A \rightarrow Y \text{ é } B$ . Essa declaração representa implicitamente uma relação fuzzy  $R$  dada por  $\mu_R(x,y) = \mu_A(x) \text{ I } \mu_B(y)$ , onde  $\text{I}$  é um operador fuzzy de implicação. Os principais estão ilustrados na Tabela 12.3. No caso mais comum  $\text{I}$  é o produto cartesiano (e representa a função mínimo), logo  $R = A \times B$ ; e se a declaração condicional for do tipo *se  $X$  é  $A$  então  $Y$  é  $B$  senão  $V$* , a relação é dada por  $R = A \times B + (\neg A \times V)$ , onde  $+$  é o operador “ou” (normalmente a função máximo). A *avaliação do conseqüente* (disparo da regra) se-então pode ser do tipo *modus ponens* generalizado (MPG) ou por regra de inferência composicional.

*MPG* tem a forma: dado que  $S_1 \text{ é } Q_1$  e *se  $S_1 \text{ é } P_1$  então  $S_2 \text{ é } P_2$* , então conclua  $S_2 \text{ é } Q_2$ , onde  $S_i$  são variáveis lingüísticas (nomes simbólicos para objetos) e  $P_i$  e  $Q_i$  são propriedades de objetos (valores que assumem). Observe que para podermos usar MPG, a propriedade  $P_i$  deve ser relacionada com a  $Q_i$ , p. ex. “o tomate é muito vermelho”, “se o tomate é vermelho então o tomate está maduro”, conclua “o tomate é muito maduro”.  $S_1 \text{ é } Q_1$  pode ser interpretado como o conjunto fuzzy  $A^*$ , se  $S_1 \text{ é } P_1$  então  $S_2 \text{ é } P_2$  como a regra condicional  $A \rightarrow B$ , e  $S_2 \text{ é } Q_2$  como o conjunto fuzzy  $B^*$ . O valor da inferência ( $B^*$ ) é obtido através da composição de  $A^*$  com a relação fuzzy  $R$ , i.é  $B^* = A^* \circ R$ . A relação  $R$  está representada implicitamente na regra se-então e pode ser calculada através da equação  $\mu_R(x,y) = \mu_A(x) \text{ I } \mu_B(y)$ .

A *regra de inferência composicional* tem a forma: dado que  $S_1 \text{ é } Q_1$  e  $S_1 R S_2$ , então conclua  $S_2 \text{ é } Q_2$ , onde  $S_1 R S_2$  expressa “ $S_1$  está na relação  $R$  para  $S_2$ ”, sendo seu significado representado por uma relação fuzzy  $\mu_R$ . Como no caso do MPG,  $S_1 \text{ é } Q_1$  é interpretado como  $A^*$  e  $S_2 \text{ é } Q_2$  como  $B^*$ , sendo o valor da inferência dado por  $B^* = A^* \circ R$ . Nesse caso a relação  $R$  representa explicitamente a conexão entre duas proposições fuzzy ( $S_1$  e  $S_2$ ).

Uma outra regra de inferência envolvendo a função de implicação fuzzy é o *modus tollens* generalizado (MTG), o qual tem a forma: dado que  $S_2 \text{ é } Q_2$  e *se  $S_1 \text{ é } P_1$  então  $S_2 \text{ é } P_2$* , então conclua  $S_1 \text{ é } Q_1$ .  $S_2 \text{ é } Q_2$  pode ser interpretado como o conjunto fuzzy  $B^*$ , se  $S_1 \text{ é } P_1$  então  $S_2 \text{ é } P_2$  como a regra condicional  $A \rightarrow B$ , e  $S_1 \text{ é } Q_1$  como o conjunto fuzzy  $A^*$ . De forma análoga ao MPG o valor da inferência é  $A^* = B^* \circ R$ , onde a relação  $R$  é dada por  $\mu_R(x,y) = \mu_A(x) \text{ I } \mu_B(y)$ .

A Tabela 12.3 apresenta os principais operadores de implicação usados na lógica fuzzy, para  $X \text{ é } A \rightarrow Y \text{ é } B$ , com respectivos universos de discurso  $U$  e  $V$ .

**Tabela 12.3** - Principais Operadores de Implicação Fuzzy

Significado	Função de Pertinência	Nome
$R_b = (\neg A \times V) \cup (U \times B)$	$\mu_{Rb} = \max(1 - \mu_{ce(A)}(x,y), \mu_{ce(B)}(x,y))$	Kleene-Dienes
$R_a = (\neg A \times V) \oplus (U \times B)$	$\mu_{Ra} = \min(1, 1 - \mu_{ce(A)}(x,y) + \mu_{ce(B)}(x,y))$	Regra aritmética de Zadeh
$R_m = (\neg A \times V) \cup (A \times B)$	$\mu_{Rm} = \max(1 - \mu_{ce(A)}(x,y), \min(\mu_{ce(B)}(x,y), \mu_{ce(A)}(x,y)))$	Regra max-min de Zadeh
$p \rightarrow_g q = 1$ , se $p \leq q$ , q, se $p > q$	$\mu_{Rg} = (\mu_{ce(A)}(x) \rightarrow_g \mu_{ce(B)}(y))$	Implicação Gödel
$R_s = A \times V \rightarrow U \times B$	$\mu_{Rs} = 1$ , se $\mu_{ce(A)}(x) \leq \mu_{ce(B)}(y)$ , senão 0	Implicação Sharp

$R_c = A \times B$	$\mu_{R_c} = \min(\mu_{ce(A)}(x), \mu_{ce(B)}(y))$	Mamdani
$R_p = A \times B$	$\mu_{R_p} = \mu_{ce(A)}(x) \cdot \mu_{ce(B)}(y)$	Implicação Larsen

Fonte: Hengel, Paulo. Notas de aula do curso Sistemas Fuzzy, CPGCC/II, 1995

O operador  $\oplus$  é a soma limitada (ao valor 1) e  $\neg$  é complemento ou negação. As seguintes relações se mantêm:  $R_a \supseteq R_g \supseteq R_s$  (i.é  $\mu_{R_a}(x,y) \geq \mu_{R_g}(x,y) \geq \mu_{R_s}(x,y)$ );  $R_a \supseteq R_g \supseteq R_c$ ; e  $R_a \supseteq R_b \supseteq R_m \supseteq R_c$ .

**Representação de um conjunto de regras fuzzy.** Uma base de regras fuzzy pode ser vista como um conjunto de regras tipo MIMO (Multiplus Input Multiplus Output), com antecedente com  $p$  proposições e conseqüente com  $q$  proposições, da forma  $R_{MIMO}^i$ : se  $(x_1 \text{ é } A_{1i} \text{ e } .. \text{ e } x_p \text{ é } A_{pi})$  então  $(z_1 \text{ é } C_{1i}, .., z_q \text{ é } C_{qi})$ . O seu antecedente forma um conjunto fuzzy em  $A_{1i} \times .. \times A_{pi}$  e o seu conseqüente é a união de  $q$  ações de controle independentes. Portanto a base de  $n$  regras pode ser representada pela união  $\{\bigcup_{i=1}^n R_{MIMO}^i\}$ , mas levando em conta a independência das ações de controle, a base pode ser representada como:  $\{\bigcup_{k=1}^q \bigcup_{i=1}^n [(A_{1i} \times .. \times A_{pi}) \rightarrow z_k]\}$ , ou seja  $\{\bigcup_{k=1}^q R_{MISO}^k\}$ , i.é um conjunto de  $q$  regras MISO (Multiplus Input Single Output). Podemos então utilizar uma T-norma para implementar o conectivo “e” na premissa das regras e um operador de implicação para avaliar a implicação. Os valores das variáveis associadas aos disparos das regras MISO podem ser agregados utilizando uma T-norma ou uma T-conorma. É bastante comum ser utilizada a composição max-min, onde min representa o operador (de implicação fuzzy) mínimo da regra de Mamdani e o max o operador de união.

### 12.1.3 Vantagens e Desvantagens

A grande vantagem de sistema fuzzy é permitir o uso de informação não precisa, representada por expressões lingüísticas que são do conhecimento e uso do especialista no domínio da aplicação. O especialista normalmente as utiliza para expressar (em forma de regras) o conhecimento que possui.

A utilização mais significativa da teoria dos conjuntos fuzzy em sistemas baseados em conhecimento são os controladores FKBC [LEE90, DHR93 Apud BIT 96, p.183]. As regras de controle utilizadas são do tipo MISO. Em cada ciclo do processo são coletados dados sobre o estado do sistema, os quais são utilizados durante o disparo das regras. Em geral os resultados dos disparos são agregados, gerando uma única ação de controle.

Esse enfoque é adequado para especificar o nível em que um objeto satisfaz um descrição lingüística vaga, não constituindo um método geral para raciocínio incerto e tem sido visto por alguns pesquisadores mais como um meio de manipular variáveis contínuas do que incerteza. Lógica fuzzy é funcionalmente-verdade, pois permite determinar o grau de pertinência de um objeto descrito por uma expressão complexa em termos dos graus de pertinências das suas proposições átomos. Como todos os sistemas funcionalmente-verdade, apresenta problemas com raciocínio misto ou intercausal (veja Seção 10.1) [RUS 95]. Mas apresenta o problema de ser inconsistente com a lógica de primeira ordem, pois a equivalência padrão  $A \vee \neg A \Leftrightarrow \text{True}$  não se mantêm, pois é normalmente avaliada como  $\mu_A(x) \vee (1 - \mu_A(x)) = \max(\mu_A(x), 1 - \mu_A(x))$  é diferente de 1.

A lógica fuzzy tem sido utilizada com sucesso em aplicações comerciais em sistemas de controle para transmissão automática de veículos, trens, câmaras de vídeo e chuveiros elétricos, etc [RUS 95]. Elkan, em artigo de 1993, “The paradoxical success of fuzzy logic” [Apud RUS 95], afirma que essas aplicações obtiveram sucesso porque possuem bases de conhecimentos pequenas, não possuem encadeamento de inferências e possuem parâmetros que podem ser ajustados para melhorar a performance do sistema (com treinamento). Elas poderiam ter sido implementadas com sucesso em outros formalismos. Ele prediz que quando aplicações de maior porte forem desenvolvidas, elas irão incorrer na mesma espécie de problemas associada ao uso de fatores de certeza (Capítulo 9).

## 12.2 Teoria de Possibilidades

A teoria de possibilidades foi proposta por Lotfi Zadeh [ZAD 78] para tratar incerteza em sistemas fuzzy e enfocava o conceito de medida de possibilidade apenas. Já Didier Dubois e Henri Prade [DUB 88] enfocam os conceitos de medidas de possibilidade e de necessidade. Uma proposição ‘X é F’ pode ser expressa como  $R(A(X)) = F$ , significando uma restrição sobre os valores de um atributo implícito de X. Por simplicidade, consideraremos  $A(X)=X$ . Para Zadeh, uma distribuição de *possibilidades* é definida como “uma restrição fuzzy  $R(X)$  que atua como uma restrição elástica sobre os valores que a variável X pode assumir. ... Se F é um conjunto fuzzy (do universo de discurso U), caracterizado pela sua função de pertinência  $\mu_F$ , então a proposição ‘X é F’, onde X é uma variável com valores em U, induz a distribuição de possibilidades  $\Pi_X$ , definida por  $\Pi_X = R(X)$ , a qual iguala a possibilidade de  $X=x$  a  $\mu_F(x)$ , i.é  $\pi_x(x) = \mu_F(x)$ .” A função  $\pi_x$  é denotada função de distribuição de possibilidades e  $\pi_x(x)$  representa o grau de possibilidade da variável X ter o valor x. Por exemplo [ZAD 78], considere a proposição “João é jovem”, na qual ‘jovem’ é um conjunto fuzzy de  $U=[0,100]$  anos, caracterizado pela função de pertinência  $\mu_{jovem}(x)$ , onde x representa a idade de João. Seja  $\mu_{jovem}(28) = 0,7$ . Pela lógica fuzzy 0,7 pode ser interpretado como o grau de compatibilidade de 28 anos com o conceito ‘jovem’. Pela teoria de possibilidades 0,7 transforma-se no grau de possibilidade de que a idade de João seja 28 anos, dado que João é jovem.

Em seu artigo Zadeh advoga que a imprecisão intrínseca ao significado de proposições em linguagem natural é principalmente de natureza possibilística ao invés de probabilística e que uma proposição p pode ser traduzida em um procedimento que calcula a distribuição de possibilidades de um conjunto de atributos implicados por p. Também afirma que grande parte da informação com base na qual as pessoas tomam decisões, é de natureza possibilística e não probabilística.

Para realçar a diferença entre uma distribuição de possibilidades e uma distribuição de probabilidades, Zadeh considera a proposição “Hans come X ovos no café da manhã”, com X assumindo valores em  $U = \{1, 2, 3, \dots\}$ . Então podemos associar uma distribuição de possibilidades a X, interpretando  $\pi_x(x)$  como o grau de facilidade com que Hans come x ovos, e uma distribuição de probabilidades a X, interpretando  $P_x(x)$  como a probabilidade de Hans comer x ovos no desjejum (Tabela 12.4).

**Tabela 12.4** - Exemplo de Distribuições de Possibilidade e Probabilidades

x	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

$\pi_x(x)$	1	1	1	1	0,8	0,6	0,4	0,2
$P_x(x)$	0,1	0,8	0,1	0	0	0	0	0

Fonte: Zadeh, 1978. Fuzzy sets as a basis for a theory of possibility, p.8

Observe que um alto grau de possibilidade não implica uma alta probabilidade (p. ex. para  $x=3$ ) e que uma baixa probabilidade também não implica um baixo grau de possibilidade (p. ex.  $x=4$ ). Mas se um evento for impossível então deverá ser improvável.

No exemplo acima, a possibilidade de  $X$  assumir um valor  $x$  é interpretado como o grau de *facilidade* com o qual  $x$  pode ser atribuído a  $X$ , podendo ter ou não significado físico real. No caso em questão, é o grau de facilidade com que Hans pode comer  $x$  ovos (portanto tem significado físico). Já no exemplo “João é jovem”, 0,7 representa o grau de facilidade com que 28 pode ser atribuído a idade de João e é sem significado físico real.

A partir da função de distribuição de possibilidades se pode calcular as medidas de possibilidades  $\Pi$  e de necessidade  $N$  de um evento  $A$ , relacionadas por  $N(A)=1-\Pi(\bar{A})$ . Seja  $X$  uma variável que toma valores no universo de discurso  $U$ ,  $\pi_x$  a função de distribuição de possibilidade associada a  $X$  e  $A$  um conjunto:

a) *não fuzzy* em  $U$ , então:

$$\Pi(A) = \sup_{x \in A} \pi_x(x) \text{ e } N(A) = \inf_{x \in \bar{A}} (1 - \pi_x(x)), \text{ onde}$$

$\Pi(A)$  é interpretado como a possibilidade de um valor de  $X$  pertencer a  $A$ .

b) *fuzzy* em  $U$ , então:

$$\Pi(A) = \sup_{x \in U} \min(\mu_A(x), \pi_x(x)) = \text{alt}(A \cap \Pi_X), \text{ onde alt é o operador altura, e}$$

$$N(A) = \inf_{x \in U} \max(\mu_A(x), (1 - \pi_x(x))), \text{ onde } \mu_A(x) \text{ é a função de pertinência de } A.$$

$\Pi(A)$  é interpretado como a possibilidade de “ $X$  é  $A$ ”.

Seja [ZAD 78]  $\Pi_X = 1/1 + 1/2 + 0,8/3 + 0,6/4 + 0,4/5 + 0,2/6$  a distribuição de possibilidades associada com a proposição “ $X$  é um inteiro pequeno”. Seja  $A = \{3,4,5\}$ , então a possibilidade de um inteiro pequeno  $x$  estar contido em  $A$  é  $\Pi(A) = \max(0,8, 0,6, 0,4) = 0,8$ . Mas se agora consideramos  $A$  como o conjunto fuzzy dos inteiros que não são pequenos, i.é  $A = 0,2/3 + 0,4/4 + 0,6/5 + 0,8/6 + 1/7 + \dots$ , então a possibilidade de  $X$  não ser um inteiro pequeno é  $\Pi(A) = \text{alt}(A \cap \Pi_X) = \text{alt}(0,2/3 + 0,4/4 + 0,4/5 + 0,2/6) = 0,4$ .

### 12.2.1 Lógica Possibilista

Dubois e Prade [DUB 88] possuem outra interpretação para as medidas de possibilidade e de necessidade. Eles atribuem a elas uma interpretação lógica relacionada à ausência de precisão da informação disponível. Assim, na lógica possibilista, a cada proposição  $p_i$  é atribuído um grau de possibilidade  $\Pi(p_i)$  e um grau de necessidade  $N(p_i) = 1 - \Pi(\neg p_i)$  de que  $p_i$  possa ser considerado verdade, com base nos fatos armazenados em uma base de conhecimentos. A lógica possibilista é uma lógica de incerteza baseada no uso de medidas de possibilidade, voltada para a realização de raciocínio incerto (i.é raciocínio preciso sobre uma situação descrita de forma incompleta em um banco de conhecimento) onde a base de conhecimentos é uma descrição fuzzy de uma dada situação (mundo). Os seus axiomas são:  $\Pi(I)=1$ ,  $\Pi(O)=0$  e  $\forall(p,q) \Pi(p \vee q)=\max(\Pi(p), \Pi(q))$ , onde  $I$  e  $O$  são, respectivamente, as proposições *true* e *false*. Se são utilizadas proposições da *lógica clássica* (i.é que assume valores falso ou verdadeiro), então a relação  $\max(\Pi(p), \Pi(\neg p)) = 1$  deve ser satisfeita e portanto: a)  $N(p)=1$  implica  $p$  verdade; b)  $\Pi(p)=0$  implica  $p$  falso; c)  $N(p)=0$  e  $\Pi(p)=1$  representa ignorância total sobre a verdade ou falsidade.

de de p. Em geral a ignorância é representada pelo par  $N(p)=N(\neg p)=0$  e a certeza pelo par  $N(p)=1, N(\neg p)=0$ .

A medida de possibilidade da união e da interseção de dois conjuntos fuzzy A e B são dadas por:  $\Pi(A \cup B) = \Pi(A) \vee \Pi(B)$  e  $\Pi(A \cap B) \leq \Pi(A) \wedge \Pi(B)$ , onde  $\vee$  é o operador máximo e  $\wedge$  é o operador mínimo.

Retornando as idéias de Zadeh, temos trabalhado com proposições  $p \equiv X \text{ é } F$ , onde  $\equiv$  representa “é definido como”, que traduzimos como:  $X \text{ é } F \rightarrow R(A(X)) = F$  o que induz uma função de distribuição de possibilidades, ou seja:  $X \text{ é } F \rightarrow \Pi_{A(X)} = F$ . Portanto temos assumido que p contém apenas um único atributo implícito de X, i.é A(X), cuja função de distribuição de possibilidades é dada pela função de pertinência de F.

No caso mais geral p pode conter n atributos implícitos,  $A_i(X)$ , cada um deles com valores em  $U_i$ ,  $i = 1, n$ . Logo a tradução de p se torna  $X \text{ é } F \rightarrow R(A_1(X), \dots, A_n(X)) = F$ , onde F é uma relação fuzzy definida no produto cartesiano  $U = U_1 \times \dots \times U_n$  e analogamente  $X \text{ é } F \rightarrow \Pi_{(A_1(X), \dots, A_n(X))} = F$ , onde R é uma restrição fuzzy n-ária e  $\Pi_{(A_1(X), \dots, A_n(X))}$  é a distribuição de possibilidades n-ária induzida por p, com função de distribuição de possibilidades dada por:  $\pi_{(A_1(X), \dots, A_n(X))}(x_1, \dots, x_n) = \mu_F(x_1, \dots, x_n)$ , para  $(x_1, \dots, x_n) \in U$ , onde  $\mu_F$  é a função de pertinência de F. No caso específico de F ser o produto cartesiano de n relações unárias  $F_1, \dots, F_n$ , então  $X \text{ é } F$  pode ser traduzido por n relações unárias  $R(A_i(X))$  sobre n conjuntos fuzzy  $F_i$ , implicando as equações abaixo:

$$\begin{array}{ll} X \text{ é } F \rightarrow R(A_1(X)) = F_1 & \Pi_{(A_1(X), \dots, A_n(X))} = \Pi_{A_1(X)} \times \dots \times \Pi_{A_n(X)} \\ & \pi_{(A_1(X), \dots, A_n(X))}(x_1, \dots, x_n) = \mu_{A_1(X)}(x_1) \wedge \dots \wedge \mu_{A_n(X)}(x_n) \\ & \dots \\ R(A_n(X)) = F_n & \pi_{(A_i(X))}(x_i) = \mu_{F_i}(x_i), \text{ para } x_i \in U_i, i = 1, n \end{array}$$

Os conceitos de união, interseção e produto cartesiano são facilmente estendidos para distribuições de possibilidades n-dimensionais. Podemos expressar a distribuição de possibilidades n-dimensional  $\Pi_X = \sum_i \pi_i / r_1^i \dots r_n^i$ , para  $i=1, N$ , onde N é o número de tuplas n-dimensionais existentes, sendo o símbolo  $\sum$  interpretado como o operador união. Então seja ‘r’ e ‘s’ duas tuplas n-dimensionais e  $\alpha$  e  $\beta$  as suas respectivas possibilidades:

- a)  $\alpha/r + \beta/s = (\alpha \vee \beta)/r$ , onde ‘+’ é a união e ‘ $\vee$ ’ é o máximo;
- b)  $\alpha/r \cap \beta/s = (\alpha \wedge \beta)/r$ , onde ‘ $\cap$ ’ é a interseção e ‘ $\wedge$ ’ é o mínimo;
- c)  $\alpha/r \times \beta/s = (\alpha \wedge \beta)/rs$ , onde ‘ $\times$ ’ é o produto cartesiano.

As variáveis fuzzy  $X_{(q)} \equiv (X_1, \dots, X_m)$  e  $X_{(q')} \equiv (X_m, \dots, X_n)$  são *não interativas* sse a distribuição de possibilidades associada com  $X = (X_1, \dots, X_n)$  é o produto cartesiano das distribuições de possibilidades associadas com  $X_{(q)}$  e  $X_{(q')}$ , i.é:  $\Pi_X = \Pi_{X_{(q)}} \times \Pi_{X_{(q'')}}$ .

### 12.2.2 Distribuição de Possibilidades de Proposições Complexas

A tradução de uma proposição em linguagem natural p, visando capturar seu significado, produz uma distribuição de possibilidade de um conjunto de variáveis ou relações fuzzy que estão explícitas ou implícitas em p. Através de regras de *tradução condicional* podemos relacionar o significado de uma proposição ao significado de suas modificações ou combinações com outras proposições. As regras a serem vistas permitem a ob-

tenção de distribuições de possibilidades a partir da modificação de distribuições existentes.

**Regras do Tipo I.** Seja  $p$  uma proposição da forma “ $X$  é  $F$ ” e  $m$  um modificador tal como “muito”, “mais ou menos”, “não”, etc. Se  $X$  é  $F \rightarrow \prod_{A(X)} = F$  então  $X$  é  $mF \rightarrow \prod_{A(X)} = F^+$ , onde  $A(X)$  é um atributo implícito de  $X$  e  $F^+$  é a modificação de  $F$  definida por  $m$ . Por exemplo se  $m$  for “muito”,  $F^+ = F^2$  (i.é  $\pi_X^2$ ); se for “mais ou menos”,  $F^+ = F^{1/2}$ ; e se for “não”,  $F^+ = F'$  (complementar de  $F$ ).

**Regras do Tipo II.** Se  $p$  e  $q$  são proposições, então  $r=p*q$  é uma proposição denotada composição de  $p$  e  $q$ . As composições mais comuns são: conjunção (operador ‘e’), disjunção (operador ‘ou’) e condicional (operador ‘se .. então’). Seja  $p \equiv “X$  é  $F”$  e  $q \equiv “Y$  é  $G”$  tal que a tradução de  $p$  induz  $A \prod_{A(X)} = F$  e a tradução de  $q$  induz  $A \prod_{B(Y)} = G$ . Então as respectivas distribuições de possibilidades associadas a composição (*não interativas*) destas proposições são dadas por:

- a) conjunção (não interativa):  $X$  é  $F$  e  $Y$  é  $G \rightarrow \prod_{(A(X),B(Y))} = F \times G$ ,
- b) disjunção (não interativa):  $X$  é  $F$  ou  $Y$  é  $G \rightarrow \prod_{(A(X),B(Y))} = ce(F) + ce(G)$ , onde  $ce(.)$  é a extensão cilíndrica (veja seção 12.1.1),
- c) condicional (não interativo): Se  $X$  é  $F$  então  $Y$  é  $G \rightarrow \prod_{(A(X),B(Y))} = ce(F') \oplus ce(G)$  onde  $F'$  é o complemento de  $F$  e  $\oplus$  é o operador de soma limitada. Portanto a função de distribuição de possibilidades é:  $\mu_{ce(F') \oplus ce(G)} = \min(1, 1 - \mu_F + \mu_G)$ .

### 12.2.3 Vantagens e Desvantagens

Lógica possibilística permite a representação de ignorância sobre o verdadeiro valor de uma variável. Na teoria de probabilidades só se consegue modelar a ignorância de forma aproximada, através da aleatoriedade. Se nenhum dos elementos de uma distribuição de possibilidades tem valor de possibilidade igual a um, então a informação disponível é incompleta.

Pela interpretação de Dubois e Prade, o grau de possibilidade e o grau de necessidade atribuído a uma proposição  $p$  podem ser considerados como uma medida da extensão em que é possível ou necessário que  $p$  seja verdadeira. Observe que a proposição não envolve predicados vagos (i.é assume apenas valores verdadeiro ou falso) e que é avaliada com base nos fatos armazenados em uma base de conhecimentos. Devido a falta de precisão da informação disponível (ela é incerteza), podemos obter apenas uma resposta aproximada da veracidade de  $p$ . Com essa interpretação, lógica possibilística pode ser considerada uma extensão da lógica clássica, quando se considera que o par  $(N(p), N(\neg p))$  somente seja  $(0,1)$  ou  $(1,0)$ .

Esses autores consideram “a lógica possibilística como um quasi-cálculo qualitativo onde números são comparados mas não somados ou multiplicados. São utilizados apenas para permitir comparações, não necessitando grande precisão.” [DUB 88, p.760]. Ao contrário do enfoque probabilístico, onde pode haver um aumento de erros em função da soma de probabilidades no cálculo da união de eventos, os erros permanecem constantes com a aplicação da lógica possibilística.

Teoria de possibilidades é mais apropriada para representação da imprecisão inerente ao uso de conceitos lingüísticos vagos do que para representar incerteza em um domínio.

Permite raciocinar com predicados precisos em uma base de conhecimento imprecisa. Assim se sabemos que “João é alto” então “a altura de João é maior do que 1,80m?” é respondida em termos de um grau de possibilidade. Lógica fuzzy é apropriada para raciocinar com predicados vagos (fuzzy) em uma base de conhecimentos precisa. Assim se sabemos que “a altura de João é 1,70 m” então a declaração “João é alto” é dita ser verdadeira, com um grau de verdade associado [Marie-Odile Cordier In: DUB 88].

Dubois e Prade [DUB 88] apresentam os seguintes exemplos de aplicações que utilizam a teoria de possibilidades: a) SPII-2 - máquina de inferência capaz de combinar raciocínio simbólico com cálculo numérico e tratar predicados vagos (lógica fuzzy) ou incertos (distribuição de possibilidades) - testada no domínio de geologia de petróleo; b) DIA-BETO III - ferramenta de apoio a decisão no tratamento de diabetes; c) TAIGER - máquina de inferência capaz de tratar regras e informação vaga ou incerta sobre o valor de variáveis lógicas ou numéricas.

### 12.3 Normas Triangulares

Bonissone considera os enfoques simbólico e numérico (não baseado em lógica fuzzy) inadequados para tratar incerteza e propõe uma abordagem baseada em três camadas: representação, inferência e controle. Ela opera com probabilidades lingüísticas e não exige estimativa precisa e consistente da incerteza de dados atômicos e de suas relações.

#### 12.3.1 Camada de Representação

A camada de representação utiliza um formalismo baseado em *frames* (para representar informação simbólica) com *slots* numéricos preenchidos com valores representando  $L(A)$ , limite inferior de confirmação da evidência  $A$  e  $L(\neg A)$ , limite inferior de refutação de  $A$ . Esses limites são usados pela camada de inferência. Outros *slots* numéricos contêm a) uma avaliação dessa medida de incerteza (similar ao conceito de variância estatística); b) uma avaliação de uma função de entropia (para medir a qualidade da informação disponível) e c) uma medida do grau (potencial) de conflito. Eles são utilizados pela camada de controle para identificar a presença de ignorância e conflito. São utilizados *slots* não numéricos contendo informação contextual sobre a fonte de evidência, sua credibilidade em prover tal tipo de evidência e as condições (ambientais ou operacionais) sob as quais a fonte obteve a informação. Com base nesses dados, a camada de controle seleciona o cálculo de incerteza a utilizar, caracterizado pela tripla (T-norma, T-conorma, negação).

Bonissone utiliza o resultado obtido por Szolovits e Pauker de que “embora as pessoas pareçam totalmente preparadas para fornecer estimativas qualitativas de verossimilhança, elas com freqüência, notoriamente relutam para fornecer estimativas numéricas precisas” para afirmar que é razoável esperar que o usuário seja capaz de prover estimativas lingüísticas (consistentes). Em seus experimentos, o autor utilizou três conjuntos de probabilidades lingüísticas, contendo 5, 9 e 13 termos. A seleção de um desses conjuntos define uma granularidade de incerteza. A semântica de cada elemento é dada por uma número fuzzy no intervalo  $[0,1]$ , cuja função de pertinência é interpretada como o significado de um termo descrevendo o grau de certeza de uma forma lingüística. A seleção apropriada da granularidade da incerteza ainda está sujeita a uma certa subjetividade, mas aparentemente é improvável que um especialista ou usuário possa quantifi-



car incerteza de forma consistente usando mais do que 9 valores diferentes [BON 86]. A Tabela 12.5 apresenta o conjunto de 9 probabilidades lingüísticas utilizadas por Bonissone, descrito com a notação  $N_i=(a_i,b_i,\alpha_i,\beta_i)$ , apresentada na seção 12.1.2.

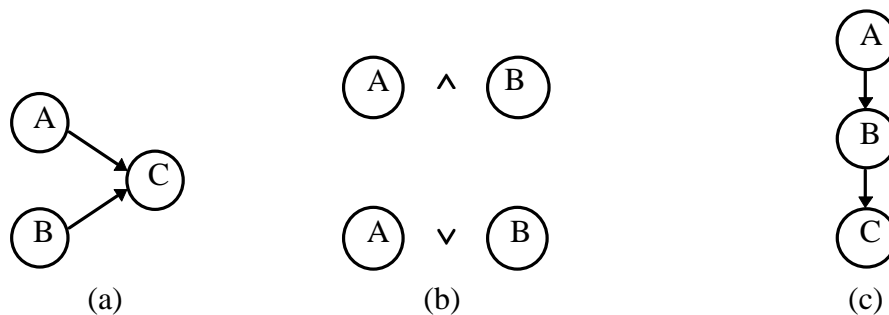
**Tabela 12.5** - Probabilidades Lingüísticas

Símbolo	Significado
Impossível	(0, 0, 0, 0)
Extremamente improvável	(0,01, 0,02, 0,01, 0,05)
Chance muito baixa	(0,1, 0,18, 0,06, 0,05)
Pequena chance	(0,22, 0,36, 0,05, 0,06)
É provável	(0,41, 0,58, 0,09, 0,07)
Chance significativa	(0,63, 0,80, 0,05, 0,06)
Muito provável	(0,78, 0,92, 0,06, 0,05)
Extremamente provável	(0,98, 0,99, 0,05, 0,01)
Certo	(1, 1, 0, 0)

Fonte: Bonissone, 1987. Summarizing and Propagating Uncertain Information with Triangular norms, p.242

### 12.3.2 Camada de Inferência

A camada de inferência utiliza normas triangulares para combinar incertezas. A Figura 9.2, reproduzida aqui como Figura 12.3, ilustra as seguintes formas: consolidar uma conclusão derivada de diferentes regras (combinação paralela - Figura 12.3a), avaliar a satisfação de premissas (combinação serial - Figura 12.3b) e propagar incerteza através de encadeamento de regras (combinação de regras - Figura 12.3c).



**Figura 12.3** - Combinando Regras Incertas

A incerteza associada a evidência A é representada através dos limites inferiores  $L(A)$  e  $L(\neg A)$ . A seguinte identidade se mantém:  $U(A) = 1 - L(\neg A)$ , onde  $L(\neg A)$  representa o limite inferior de refutação de A e  $U(A)$  representa o limite superior de certeza em A. Assim o intervalo de certeza associado a A é dado por  $[L(A), U(A)]$ .

Nesse cálculo, geralmente é utilizado o operador de negação  $N(x) = 1-x$  e normas triangulares *duais* T-normas  $T(.,.)$  e T-conormas  $S(.,.)$ , definidas na seção 12.1.1, conforme descrito a seguir:

a) combinação paralela

Seja  $[c_j, C_j]$  o intervalo formado pelo limite inferior e pelo limite superior de certeza associado a uma mesma conclusão, derivada por m regras. Então o intervalo  $[d, D]$  consolidado da conclusão é dado por  $[S(c_1, c_2, \dots, c_m), S(C_1, C_2, \dots, C_m)]$ ;

## b) combinação serial

Seja  $[b_i, B_i]$  o intervalo formado pelo limite inferior e pelo limite superior de certeza da  $i$ -ésima condição de uma premissa de uma dada regra. Então o intervalo  $[b, B]$  da premissa é definido por  $[T(b_1, b_2, \dots, b_m), T(B_1, B_2, \dots, B_m)]$ ;

## c) combinação de regra

Sejam os limites inferiores dos graus de suficiência e necessidade de uma regra do tipo  $P \leftrightarrow Q$ , dados respectivamente por  $s$  e  $n$ , e  $[b, B]$  o intervalo de certeza associado à sua premissa. O intervalo de certeza associado à conclusão da regra é dado por  $[c, C] = [T(s, b), 1 - T(n, 1 - B)]$ , onde  $s = L(P \rightarrow Q)$  e  $n = L(\neg P \rightarrow \neg Q)$ .

Se existir mais de uma fonte de evidência, podemos utilizar a operação de *consenso* de fontes a qual reflete a fusão das medidas de certeza da mesma evidência  $A$ , providas pelas diversas fontes. Essa operação reduz a ignorância sobre a incerteza de  $A$ , produzindo um intervalo de certeza  $[L_{\text{total}}(A), U_{\text{total}}(A)]$  que é sempre menor ou igual ao menor intervalo disponível. Se existir inconsistência entre as fontes de informação, o intervalo resultante será disjunto, i.é  $L_{\text{total}}(A) > U_{\text{total}}(A)$ . Seja  $[L_i(A), U_i(A)]$ ,  $i=1, n$  os intervalos de certeza de uma mesma conclusão associada com  $n$  fontes de informação. Então o intervalo agregado é dado pela interseção de todos os intervalos, ou seja:  $[L_{\text{total}}(A), U_{\text{total}}(A)] = [\max_i L_i(A), \min_i U_i(A)]$ .

Ocorre uma inconsistência se  $L(A) > U(A)$ . Caso contrário, a medida  $L(A) - U(A)$  pode ser utilizada como uma estimativa da ignorância associada à certeza de  $A$ .

## 12.3.3 Camada de Controle

A camada de controle identifica a presença de ignorância e conflito com base em *slots* numéricos contendo a) avaliação da medida de incerteza (área sob a curva definida pelo intervalo fuzzy  $[L(A), U(A)]$  ou simplesmente a diferença  $U(A) - L(A)$  se não são fuzzy); b) avaliação da função de entropia  $f(x) = -K(x \log(x) + (1-x) \log(1-x))$ , onde a constante de normalização  $K = 1/\log(2)$ , normaliza os valores de  $f(x)$  para o intervalo  $[0,1]$  e c) medida do grau de inconsistência,  $L(A) - U(A)$ , se  $L(A) > U(A)$ .

Bonissone fez diversos experimentos com 11 T-normas (e suas T-conormas duais), considerando conjuntos de 5, 9 e 13 termos representando probabilidades lingüísticas e conseguiu identificar algumas classes de equivalências (i.é subconjuntos de normas que produzem o mesmo resultado). Para detalhes consulte [BON 86 ou 87]. Para facilitar, trabalhou com a família parametrizada de T-normas proposta por Schweizer e Sklar, denotada  $T_{Sc}(a, b, p)$ , onde  $a$  e  $b$  são argumentos da T-norma e  $-\infty < p < +\infty$  é um parâmetro, permitindo representar o espaço de T-normas variando de  $T_0$  a  $T_3$  (Tabela 12.6).

**Tabela 12.6** - Família de T-normas de Schweizer e Sklar

$T_{Sc}(a, b, p)$	$p$	T-norma
$T_{Sc}(a, b, p) = (a^{-p} + b^{-p} - 1)^{-1/p}$ se $a^{-p} + b^{-p} \geq 1$ e $p < 0$	$-\infty$	$T_0$
$T_{Sc}(a, b, p) = 0$ se $a^{-p} + b^{-p} < 1$ e $p < 0$	$-1$	$T_1$
$T_{Sc}(a, b, 0) = ab$ $p = 0$	$0$	$T_2$
$T_{Sc}(a, b, p) = (a^{-p} + b^{-p} - 1)^{-1/p}$ $p > 0$	$+\infty$	$T_3$

Fonte: Bonissone, 1987. Summarizing and Propagating Uncertain Inf. with Triangular norms.

Para o conjunto de 5 termos lingüísticos foi constatado que os resultados associados às 11 T-normas testadas poderiam ser obtidos apenas com  $T_1$ ,  $T_2$  e  $T_3$ . Para os conjuntos de 9 e 13 termos lingüísticos, os resultados associados a todas as 11 T-normas podem ser obtidos utilizando-se apenas  $T_1$ ,  $T_{Sc}(a,b,-0,5)$ ,  $T_2$ ,  $T_{Sc}(a,b,1)$  e  $T_3$ .

Para cada regra devemos indicar qual T-norma será utilizada na combinação serial e na combinação de regra; e para cada conjunto de regras com mesma conclusão, qual T-conorma utilizar na combinação paralela. A camada de controle usa tais informações para selecionar um cálculo de incerteza. Elas são codificadas em *slots* não numéricos contendo informação sobre a fonte de evidência, sua credibilidade em prover tal tipo de evidência e as condições (ambientais ou operacionais) sob as quais obteve a informação.

A indicação da T-norma reflete a posição do decisor perante risco, variando de uma atitude mais pessimista ( $p = -1$  ou seja  $T_1$ ) a uma mais otimista ( $p \rightarrow \infty$  ou seja  $T_3$ ).  $T_1$  captura a noção de pior caso, sendo os seus dois argumentos considerados tão mutuamente exclusivos quanto possível;  $T_2$  captura a noção de independência dos argumentos e  $T_3$  a noção de melhor caso, onde um argumento subordina (engloba) o outro (i.é geometricamente constituem dimensões colineares). As outras duas T-normas representam casos em que a informação possui correção negativa ( $p = -0,5$ ) ou positiva ( $p = 1$ ) suave. A indicação da T-conorma reflete a presença ou ausência de correlação positiva/negativa entre as regras no conjunto de regras com a mesma conclusão, variando de correlação negativa extrema de exclusão mútua ( $T_1$ ) a uma positiva extrema de subordinação ( $T_3$ ).

#### 12.3.4 Vantagens e Desvantagens

Essa abordagem pode ser vista como um formalismo híbrido visto que utiliza informação numérica e simbólica e apresenta as seguintes vantagens:

- a) representação explícita da evidência de suporte e de refutação de uma hipótese, do meta-conhecimento (informação sobre evidência), da consistência e de uma medida de consistência, e da medida de ignorância;
- b) não considera qualquer restrição sobre a estrutura da informação (p.ex. independência condicional das evidências e/ou exclusão mútua de hipóteses exaustivas) e nem exige que o usuário ou o especialista forneça uma estimativa precisa e consistente da incerteza dos dados atômicos e de suas relações, sendo a incerteza representada através de probabilidades lingüísticas;
- c) mantém a semântica da representação da incerteza através de um processo de transformação das probabilidades lingüísticas em conjuntos fuzzy parametrizados e da operação (com T-normas) sobre esses conjuntos, sendo que ao final do processo de raciocínio, a forma paramétrica é expressa novamente em termos lingüísticos, através da aplicação de um processo de desfuzzificação;
- d) distingue claramente inconsistência (conflito) e ignorância;
- e) permite a explicitação de mecanismo declarativo na camada de controle (regras dependentes do contexto) para especificar a política de combinação de incerteza;
- f) a funcionalidade-verdade do cálculo de incerteza usado na camada de controle garante baixa complexidade computacional; e
- g) para uma granularidade de incerteza fixa (p.ex. 9 termos lingüísticos) o número infinito de T-normas disponíveis (i.é cálculos de incerteza) pode ser reduzido a um máximo de cinco classes de equivalência distintas.

Como desvantagem podemos citar que essa abordagem dificulta a construção de aplicações porque elas devem ser estruturadas em uma arquitetura de três camadas (representação, inferência e controle), o que é complexo. Para testar as suas idéias, Bonissone construiu a arquitetura RUM - Reasoning with Uncertainty Module e conduziu alguns experimentos, cujos resultados foram aqui citados.

## Parte III    Redes de Decisão

Nesta parte são apresentados os princípios da teoria da decisão e da teoria de utilidade e como podem ser combinados com redes bayesianas para produzir o formalismo de redes de decisão.

As redes bayesianas são representadas por grafos orientados acíclicos cujos nós são variáveis aleatórias conectados por arcos que indicam uma relação de causalidade direta entre as variáveis. Cada nó tem uma tabela de probabilidades condicionais que quantificam os efeitos que os nós pais (aqueles diretamente conectados) têm sobre ele.

As redes bayesianas podem ser expandidas com uma função utilidade e incorporação de ações, produzindo um formalismo denominado *redes de decisões*. A função utilidade permite representar e raciocinar com preferências. Nós de decisão representam ações que podem ser tomadas. Esse formalismo é utilizado para a modelagem de agentes de decisão teórica, de forma tal que um agente, ao decidir por uma ação, possa mensurar a utilidade esperada (desejabilidade) dos estados possíveis, dada uma certa evidência. Cada estado (que possa ser atingindo quando um agente toma uma decisão) tem um grau de utilidade, sendo preferido os estados com maior utilidade.

Esses agentes são capazes de tomarem decisões racionais, baseados em suas crenças e na meta que desejam alcançar, mesmo atuando em um ambiente com incerteza e metas conflitantes.

Sistemas multi-agentes de decisão teórica podem ser utilizados onde aparece o problema de escolha entre um conjunto de estratégias, com base em diversas opiniões individuais, com credibilidade diversas.

## 13. Teoria da Decisão

Esse capítulo apresenta os conceitos básicos da teoria da decisão e da teoria de utilidade. A teoria da decisão provê o arcabouço racional para escolher entre cursos de ação alternativos (*decisões*) quando as conseqüências dessa escolha não são perfeitamente conhecidas (i.e os resultados são incertos). Nesse arcabouço todas informações disponíveis são utilizadas para deduzir qual das decisões alternativas é a “melhor decisão lógica possível” de acordo com as *preferências* do decisor. Obviamente não se pode garantir que essa decisão produza resultados ótimos, apenas se espera que ela minimize a conseqüência de se obter um resultado desfavorável [NOR 68]. Esse formalismo se baseia na teoria de probabilidades (para atribuir crenças a informações incompletas) e na teoria de utilidade (para manter consistência entre preferências do decisor e as decisões tomadas). Neste contexto tomar decisões significa escolher as decisões que maximizam a utilidade esperada dos resultados em um conjunto de decisões com resultados incertos.

### 13.1 Decisão Racional

Agir racionalmente significa atuar de forma a atingir os seus objetivos, face às suas crenças. Um *agente* é algo que percebe o ambiente onde está inserido, através de sensores e atua sobre o ambiente através de seus atuadores. Um agente pode ser um ser humano, um artefato de hardware (agente robótico) ou um artefato de software. Nesse caso as percepções e atuações do agente são codificadas em seqüências de bits. Um agente *racional* (ideal) é aquele que para cada possível seqüência de percepção executa a ação que espera maximizar a sua medida de performance, baseado nas evidências providas pela seqüência de percepção e no conhecimento embutido que possui. Um seqüência de percepções é tudo o que o agente percebe através de seus sensores, até o momento. A medida de performance é um critério que determina o grau de sucesso do agente em atingir as metas estabelecidas.

Agentes *reativos* atuam em resposta ao estado corrente do ambiente, o qual corresponde às condições necessárias e suficientes para disparar uma certa ação. As ações que podem executar estão previamente programadas de forma tal que cada uma delas tenha um conjunto de condições associadas. Essa associação pode se dar na forma de regras *condição-ação*. Quando os sensores captam sinais compatíveis com um conjunto de condições, a ação correlata é disparada. Ela é desencadeada com base apenas na condição-ação disparada, sem uma análise explícita dos seus desdobramentos em relação ao alcance ou não das metas e do grau de sucesso em atingi-las.

Agentes *baseados em metas* utilizam o estado corrente do ambiente e informações sobre as metas a atingir, em combinação com os resultados de ações possíveis, para escolher ações que conduzem às metas desejadas. Esse processo de decisão pode necessitar usar alguma forma de busca ou planejamento, visto que envolve a consideração do futuro. A ação é desencadeada sem uma análise explícita do grau de sucesso em atingir as metas. Idealmente podem existir diversas alternativas para se atingir uma meta sendo que algumas são mais rápidas, confiáveis, seguras ou mais baratas do que outras. Agentes baseados em metas são mais flexíveis do que os reativos mas são mais complexos, necessitando mais sensores e/ou atuadores e mais capacidade de memória e processamento.

Agentes *baseados em utilidade* tentam maximizar seus graus de satisfações (sucessos). As ações a tomar são escolhidas com base no estado corrente do ambiente; informações sobre as metas a atingir; os resultados das ações possíveis e utilidade dos estados que possam ser atingidos ao escolher determinada ação. Utilidade é uma função que mapeia um estado (ou sequência de estados) em um número real que descreve o grau de preferência do agente por aquele estado. Os agentes preferem estados com maior utilidade.

Uma especificação completa de uma função utilidade permite decisões racionais onde agentes baseados em metas não apresentam bom desempenho, quando:

- há metas conflitantes e somente algumas delas podem ser atingidas. A função utilidade estabelece o nível de compromisso adequado entre elas (p. ex. minimizar o risco de perder um voo e minimizar o tempo de espera no salão de embarque);
- existem diversas metas que o agente pode perseguir, mas sem a certeza de que vai alcançá-las (o resultado das suas ações pode ser aleatório!). A função utilidade prevê uma forma de ponderar a probabilidade de sucesso vs. a importância das metas.

A presença de incerteza altera a forma do agente racional tomar decisões. Em um ambiente sem incerteza um agente possui metas e executa um plano que garantidamente permitirá a satisfação das suas metas. Uma ação pode ser selecionada com base apenas se ela permitirá ou não atingir a meta, independentemente de outras ações que também apresentam sucesso. Com incerteza o agente deve analisar a utilidade dos diferentes resultados que podem ocorrer, em decorrência das ações que tomar.

Considere que você mora a 20 quilômetros do aeroporto, vai se deslocar até lá dirigindo o seu carro e não deseja perder o voo mas também não deseja esperar muito no salão de embarque. Lembrando que a lista de espera abre 30 minutos antes do voo, você está considerando a possibilidade (plano  $A_{90}$ ) de sair com 90 minutos de antecedência. Como pode haver congestionamento, o carro pode quebrar, furar um pneu, etc., em 95% das vezes anteriores, seguindo esse plano, você conseguiu embarcar. É racional adotar o plano  $A_{90}$ ? Não necessariamente! Em especial se for época de carnaval e o seu voo for para o Nordeste e só existirem lugares vagos nos voos após a quarta-feira de cinzas ... Assim faz sentido considerar o plano  $A_{120}$  e talvez o  $A_{150}$  mas, com certeza não o plano  $A_{24h}$ .

Nessa situação um agente é dito *racional* se e somente se ele escolhe a ação que produzirá a Máxima Utilidade Esperada (MEU), considerada sobre todos os resultados possíveis decorrentes da ação selecionada. O agente que atuar segundo a heurística MEU é denotado agente de *decisão teórica*; ele toma decisões racionais com base nas suas crenças, mesmo atuando em um ambiente com incerteza e/ou metas conflitantes.

MEU é definida por:  $\max_j UE(d_j | ev) = \max_j \{ \sum_i P(r_i(d_j) | ev, f(d_j)) \cdot U(r_i(d_j)) \}$ , onde:

- $UE(d_j | ev)$  — utilidade esperada da decisão (ação)  $d_j$  dado a evidência  $ev$ ;  
 $r_i$  —  $i$ -ésimo resultado (aleatório) possível ao tomar a decisão  $d_j$ ;  
 $f(d_j)$  — proposição indicando tomar a decisão  $d_j$  no estado corrente;  
 $P(r_i(d_j) | ev, f(d_j))$  — probabilidade de ocorrer  $r_i$ , dado  $ev$  e a decisão  $d_j$ ; e  
 $U(r_i(d_j))$  — utilidade do estado  $r_i$ , resultante da decisão  $d_j$ .

$P(r_i(d_j) \mid \text{ev}, f(d_j))$  ou de forma mais simples  $P(\theta \mid d)$  é denotada função de probabilidade de massa, sendo definida como:  $\forall(\theta \in \Theta), P(\theta \mid d) \geq 0$  e  $\sum_{\theta \in \Theta} P(\theta \mid d) = 1$ , onde  $\theta$  é um resultado possível e  $\Theta$  é o conjunto de todos os resultados possíveis, para a decisão  $d$ .

O princípio MEU aponta para a direção certa a tomar mas os cálculos envolvidos podem ser proibitivos ou ser difícil formular o problema completamente, pois:

- conhecer o *estado inicial* do mundo requer percepção, aprendizagem, representação de conhecimento e inferência;
- calcular a probabilidade do resultado de uma ação, dadas algumas evidências requer um *modelo causal* (rede bayesiana) do mundo e, em alguns casos, a atualização das crenças constitui um problema de complexidade NP completo;
- calcular a utilidade de cada estado geralmente requer um procedimento de busca ou planejamento porque o agente não sabe o quão bom é um estado até que ele conheça onde se pode chegar, partindo do mesmo. Cada estado possível é considerado ser uma fotografia completa do mundo onde o agente atua.

Teoria da decisão trata situações onde devem ser tomadas decisões cujas conseqüências não podem ser previstas com exatidão, mas podem ser estimadas e sua desejabilidade (utilidade) avaliada. Para Castiñeira [CAS 96, p.14], “o ponto chave é escolher uma ação antes de se ter a oportunidade de determinar as incertezas, como por exemplo escolher uma aplicação financeira antes de saber se ela terá o maior rendimento”. Essa teoria é *normativa* pois fornece um conjunto de critérios para garantir a consistência entre crenças, preferências e decisões, o qual *deve* ser utilizado por um agente racional.

Se a desejabilidade for medida em termos econômicos então o valor monetário associado a uma determinada conseqüência é denotado *retorno* (do inglês payoff) e o objetivo da tomada de decisão pode ser formulado como encontrar a decisão  $d^*$  que maximizar o valor esperado do retorno (critério do valor monetário esperado - EMV):

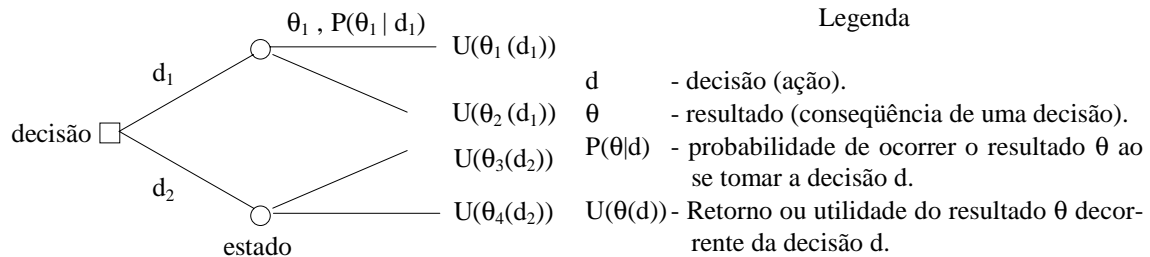
$$R(d^*) = \max_j \{ \sum_i P(r_i(d_j) \mid \text{ev}, f(d_j)) \cdot R(r_i(d_j)) \}, \text{ onde}$$

$R(r_i(d_j))$  é a função de retorno que especifica o ganho (geralmente em unidades monetárias) correspondente ao resultado  $r_i$  ao se tomar a decisão  $d_j$ .

Em muitas situações o critério do EMV não reflete as preferências do agente, pois o valor do dinheiro pode diferir de agente para agente, depende da quantia de dinheiro que já se possui e da motivação para aceitar correr riscos ou ter uma postura mais conservadora. Se um agente já tem muito dinheiro, p.ex. US\$ 500.000.00, e acabou de ganhar mais US\$ 1.000.000 em um programa de TV, ele geralmente estará mais disposto a arriscar ganhar US\$ 3.000.000, se obter cara ao lançar uma moeda, mesmo que possa perder o milhão que já ganhou (se sair coroa), do que um outro agente pobre [RUS 95]. O primeiro age de acordo com o princípio EMV: o retorno esperado de não jogar é US\$ 1.000.000 e o de jogar é US\$ 1.500.000; já o segundo não age de acordo com EMV, pois sua opção de não jogar não maximiza o retorno esperado. Portanto “em determinadas situações o critério de EMV não é o mais apropriado, faltando considerar as preferências individuais de cada agente com respeito às conseqüências do problema. A teoria de utilidades oferece um outro critério que cobre essas deficiências: o critério da utilidade. A utilidade serve para medir as preferências do agente, i.é o valor relativo frente às conseqüências ou retornos do determinado problema.” [CAS 96, p.17]. Árvores de decisão (Figura 13.1) e redes de decisão (Capítulo 14) são métodos gráficos geralmente utilizados na teoria de decisão. Uma árvore de decisão pode ser utilizada para calcular a



utilidade esperada associada com cada seqüência de ações e para determinar a utilizada esperada máxima. A seqüência  $d^*$  associada à utilidade (ou retorno) esperada máxima é denotada *estratégia de decisão*. Se o número de variáveis do problema for grande, as árvores de decisão geradas podem tornar-se confusas e complexas.



**Figura 13.1** - Árvore de Decisão

### 13.2 Teoria da Utilidade

A teoria da utilidade impõe restrições sobre as preferências do agente para que dadas algumas preferências sobre estados atômicos (básicos) individuais, seja possível derivar preferências para cenários complexos de tomada de decisão. Esse princípio é similar ao princípio da funcionalidade-verdade ou seja a partir dos valores de utilidade de estados atômicos é possível calcular a utilidade de cenários complexos.

Para facilitar o entendimento dessas restrições vamos introduzir o conceito de *loteria* (L) e a notação de preferência entre loterias ou entre estados.

Um cenário complexo é denominado *loteria* para enfatizar a idéia de que os diferentes resultados possíveis são como diferentes prêmios a serem pagos, sendo o resultado determinado pela sorte (i.é aleatoriamente). Uma loteria pode ter qualquer número de resultados possíveis e cada um deles pode ser um estado atômico ou uma outra loteria. Por exemplo,  $L = [p, A; (1-p), B]$ , representa a loteria na qual existem apenas dois estados atômicos possíveis; um com probabilidade  $p$  e o outro com probabilidade  $(1-p)$ .

As preferências entre os prêmios pagos são utilizadas para determinar preferências entre loterias ou estados. Essa preferência (ou falta de) é expressa da seguinte maneira:

- $A \succ B$        $A$  é preferível a  $B$ .
- $A \sim B$       o agente é indiferente entre  $A$  e  $B$ .
- $A \succ \sim B$   $A$  é preferível a  $B$  ou o agente é indiferente entre eles.

Os axiomas da utilidade (que estabelecem restrições sobre as preferências do agente) são os seguintes [RUS 95][NOR 68]:

- **Ordenabilidade:** dados dois estados quaisquer, um agente deve saber qual deles ele prefere ou então prefere a ambos com a mesma intensidade.

$$(A \succ B) \vee (B \succ A) \vee (A \sim B)$$

- **Transitividade:** dados três estados quaisquer, se o agente prefere  $A$  a  $B$  e prefere  $B$  a  $C$ , então ele deve preferir  $A$  a  $C$ . De forma análoga para a indiferença.

$$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C) \text{ ou } (A \sim B) \wedge (B \sim C) \Rightarrow (A \sim C)$$

- **Continuidade:** se a preferência por algum estado B está entre as preferências pelos estados A e C, então existe uma probabilidade  $0 < p < 1$  para a qual o agente ficará indiferente entre o estado B e uma loteria que forneça A com probabilidade p e C com probabilidade (1-p).

$$A \succ B \succ C \Rightarrow \exists p \mid [p, A; (1-p), C] \sim B$$

- **Substitutabilidade:** se um agente é indiferente entre duas loterias, A e B, então também será indiferente entre duas loterias mais complexas, idênticas exceto pela substituição de A em uma delas por B.

$$A \sim B \Rightarrow [p, A; (1-p), C] \sim [p, B; (1-p), C]$$

- **Monotonicidade:** suponha que existam duas loterias que forneçam os mesmos dois resultados, A e B, mas com probabilidades diferentes. Se prefere A a B, então também prefere a loteria com a maior probabilidade para A (e vice-versa).

$$A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A; (1-p), B] \succ \sim [q, A; (1-q), B])$$

- **Decomponabilidade:** loterias compostas podem ser reduzidas a outras mais simples, através das leis da probabilidade.

$$[p, A; (1-p), [q, B; (1-q), C]] \sim [p, A; (1-p)q, B; (1-p)(1-q), C]$$

Esses axiomas expressam o sistema de preferências de um agente racional, ou seja como uma agente racional *deve* atuar, sendo portanto normativos e não descritivos. Se as preferências do agente obedecem aos axiomas da utilidade, então existe uma função utilidade U que associa valores de prêmios ou loterias a valores da reta real, tal que:

$$U(A) > U(B) \Leftrightarrow A \succ B \text{ e } U(A) = U(B) \Leftrightarrow A \sim B$$

Com os axiomas e a função definida acima, a utilidade de uma loteria é completamente determinada pela utilidade e pela probabilidade de cada um dos seus resultados; assim:

$$U([p_1, S_1; \dots; p_n, S_n]) = \sum p_i \cdot U(S_i)$$

Portanto podemos calcular a utilidade de uma cenário complexo a partir da utilidade de seus estados atômicos. Esses axiomas não especificam uma função de utilidade única para um agente, dadas as suas preferências. A rigor podemos trabalhar com uma função normalizada, visto que o comportamento do agente independe da escala de utilidade utilizada, mantido o mesmo conjunto de crença. Assim a função utilidade  $U'(\cdot)$  dada por:

$$U'(S) = k_1 + k_2 \cdot U(S),$$

onde  $k_1$  é uma constante qualquer e  $k_2$  é uma constante positiva, mantêm a mesma estrutura de preferências expressa pelo agente, ao utilizar a função de utilidade  $U(\cdot)$ ; ou seja se  $U(S_1) \geq U(S_2)$ , então  $U'(S_1) \geq U'(S_2)$ . A normalização é feita considerando uma escala de 0 (pior resultado catastrófico possível) e 1 (melhor resultado possível).

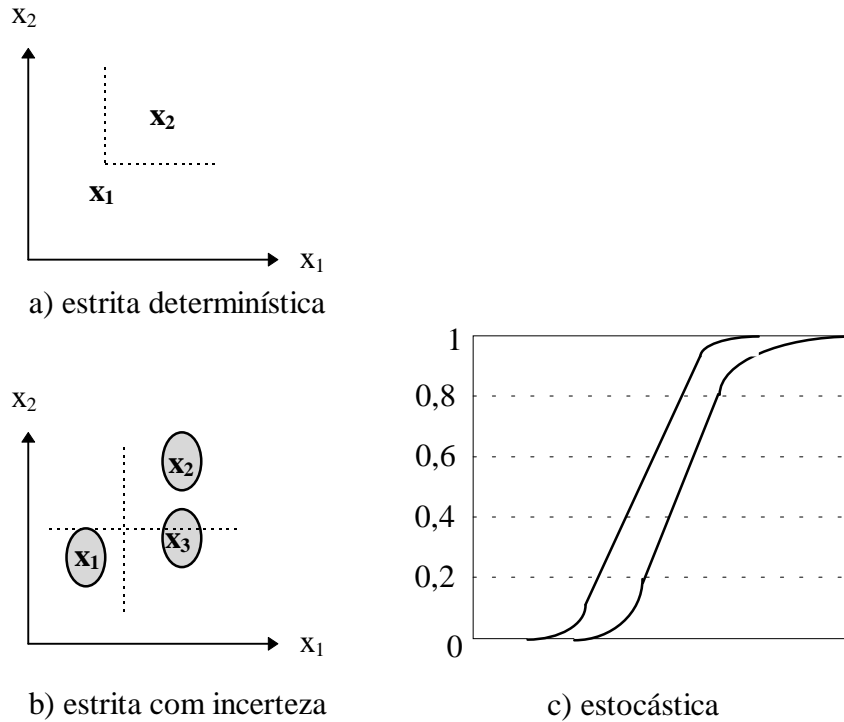
Quando a tomada de decisão envolver a análise da utilidade de mais de uma variável, temos uma função utilidade *multivariada*. Para simplificar o raciocínio adota-se a hipótese de que cada variável x é definida de forma que, mantendo-se os valores de todas as outras variáveis constantes, maiores valores para x correspondem a maiores utilidades.

Com funções multivariadas não é tão simples decidir que um estado  $\mathbf{x} = \langle x_1, \dots, x_n \rangle$  é preferível a outro. Podemos definir os seguintes casos especiais (Figura 13.2):

- **dominância estrita:** se um estado  $\mathbf{x}_1$  apresenta todas as suas n variáveis com valores menores (menores utilidades) do que o estado  $\mathbf{x}_2$ , então  $\mathbf{x}_2$  domina estritamente  $\mathbf{x}_1$  e  $\mathbf{x}_1$  pode ser descartado. No caso de estados aleatórios essa dominância só ocor-

rerá quando todos os resultados possíveis de todas as  $n$  variáveis são menores do que os resultados possíveis das variáveis associadas ao outro estado.

- **dominância estocástica:** se a distribuição de probabilidades acumulada para o estado  $\mathbf{x}_2$  está sempre à direita da distribuição de probabilidades acumulada para o estado  $\mathbf{x}_1$ , então  $\mathbf{x}_2$  domina estocasticamente a  $\mathbf{x}_1$ . Se um estado é estocasticamente dominado por outro (em todas as suas variáveis), ele pode ser descartado.



**Figura 13.2** - Tipos de Dominância

Observe que na Figura 13.2(b)  $\mathbf{x}_2$  domina estritamente  $\mathbf{x}_1$ , mas não a  $\mathbf{x}_3$  e que a Figura 13.2(c) ilustra a distribuição de probabilidades acumulada de funções de uma variável apenas (obviamente para permitir a visualização). Para simplificar procedimentos, normalmente se trabalha com agentes que apresentam *independência* preferencial mútua em estados  $n$ -dimensionais, na *ausência* de incerteza. Nesse caso duas variáveis  $x_1$  e  $x_2$  são preferencialmente independentes das outras quando a preferência entre os estados  $\langle x_1, x_2, x_3, \dots, x_n \rangle$  e  $\langle x_1', x_2', x_3, \dots, x_n \rangle$  não depende de valores particulares de  $x_3$  a  $x_n$ . Nesse caso o comportamento das preferências do agente pode ser imaginado como se maximizasse a função:  $V(\mathbf{x}) = \sum_i V_i(x_i)$ , onde  $V_i$  é uma função de valor (similar a uma função de utilidade) que depende apenas da variável  $x_i$  do estado  $\mathbf{x}$ . Normalmente  $V(\mathbf{x})$  é uma função de valor aditiva, do tipo:  $V(\mathbf{x}) = \sum_i k_i \cdot x_i$ .

Quando o domínio *envolve incerteza*, para a tomada de decisão por parte do agente, é necessário considerar a estrutura das suas preferências entre loterias. Nesse caso trabalhamos com funções utilidade e não com funções de valor. Normalmente se considera que o agente apresenta *independência* de utilidade mutuamente entre quaisquer os conjuntos  $\mathbf{y}$  e  $\mathbf{z}$  de variáveis de um estado  $\mathbf{x} = \langle \mathbf{y}, \mathbf{z} \rangle$ , i.é as preferências entre loterias nas variáveis  $\mathbf{y}$  são independentes de valores particulares das variáveis  $\mathbf{z}$  (as quais podem, por exemplo, serem mantidas constantes).

Nesse caso o comportamento das preferências do agente pode ser modelado por uma função de utilidade *multiplicativa* da forma:

$$U = k_1 U_1 + k_2 U_2 + k_3 U_3 + k_1 k_2 U_1 U_2 + k_2 k_3 U_2 U_3 + k_3 k_1 U_3 U_1 + k_1 k_2 k_3 U_1 U_2 U_3,$$

onde  $U_i = U_i(x_i)$  representa uma função utilidade da  $i$ -ésima variável do estado  $\mathbf{x}$  e  $k_i$  são constantes. Portanto para um estado  $n$ -dimensional, a função utilidade depende de  $n$  funções utilidade de um único argumento e de  $n$  constantes.

## 14 Redes de Decisões

No processo de tomada de decisão o agente pode descartar algumas alternativas quando existe dominância (estrita ou estocástica). Caso contrário deverá utilizar uma função de valor (se o domínio for determinístico) ou uma função utilidade (se o domínio for aleatório) para avaliar as suas preferências. Rede de decisão combina rede bayesiana com nós adicionais para representar *ação* e *utilidade*, compondo um formalismo útil no processo de tomada de decisão com incerteza. Foram introduzidas por Howard e Matheson, em 1984, com o nome de *diagramas de influência*, na publicação *Readings on the Principles and Applications of Decision Analysis*, editada pelos dois, no artigo *Influence Diagrams* (p.721-762), publicado pelo Strategic Decisions Group, Menlo Park, 1984.

Uma rede de decisão representa informação sobre o estado corrente, as ações possíveis de tomar, os estados resultantes e a utilidade de cada um deles, estruturada na forma de um grafo  $G=(N,A)$ , sendo  $N = P \cup D \cup V$  o conjunto dos nós e  $A$  o conjunto dos arcos:

- **nós de probabilidade** (P): variáveis aleatórias (como na rede bayesiana) representadas por ovais. Cada nó tem associado uma tabela de *probabilidades condicionais* (CPT) a qual é indexada pelo estados dos seus nós pais, como no caso da rede bayesiana. Os nós pais podem ser outros nós de probabilidade ou nós de decisão. No caso do nó não possuir nó pai, então a CPT é substituída pelas probabilidades a priori da variável em questão, sendo ela considerada uma variável de evidência.
- **nós de decisão** (D): pontos de escolha de ações (retângulos). Seus nós pais podem ser outros nós de decisão ou nós de probabilidade.
- **nós de utilidade**<sup>15</sup>(V): funções de utilidade (losangos). Cada nó possui uma tabela contendo uma descrição da utilidade do agente como função das variáveis associadas aos seus nós pais. Seus pais podem ser nós de decisão ou nós de probabilidade.

Os arcos incidindo em nós probabilísticos ou em nós de utilidade representam dependência probabilística e são denominados *arcos condicionais*. Arcos entre nós de probabilidade representam dependência entre as variáveis aleatórias associadas aos nós, podendo ou não representar uma relação de causalidade. Como a sua topologia corresponde a uma distribuição de probabilidades conjunta, o significado do diagrama não altera se invertermos a direção dos arcos. Obviamente devemos alterar as CPTs, aplicando o teorema de Bayes, para refletir essas inversões. Assim se duas variáveis aleatórias são independentes então não existirá nenhum arco entre seus nós. Nenhuma decomposição de uma função de distribuição de probabilidades conjunta pode levar a um ciclo no grafo. Se isso acontecer existe um erro na formulação do problema. Um nó de decisão é pai de um nó de probabilidade se o seu valor influencia a distribuição da variável aleatória.

Arcos incidindo em nós de decisão representam precedência temporal e indicam disponibilidade da informação na hora de tomar a decisão. Eles são denotados *arcos de informação*. Um nó de probabilidade é pai de um nó de decisão se o valor da variável aleatória for conhecido na hora da decisão e puder influenciá-la. Um nó de decisão é pai de um outro nó de decisão se a primeira decisão for realizada antes da segunda e o seu valor a influenciar.

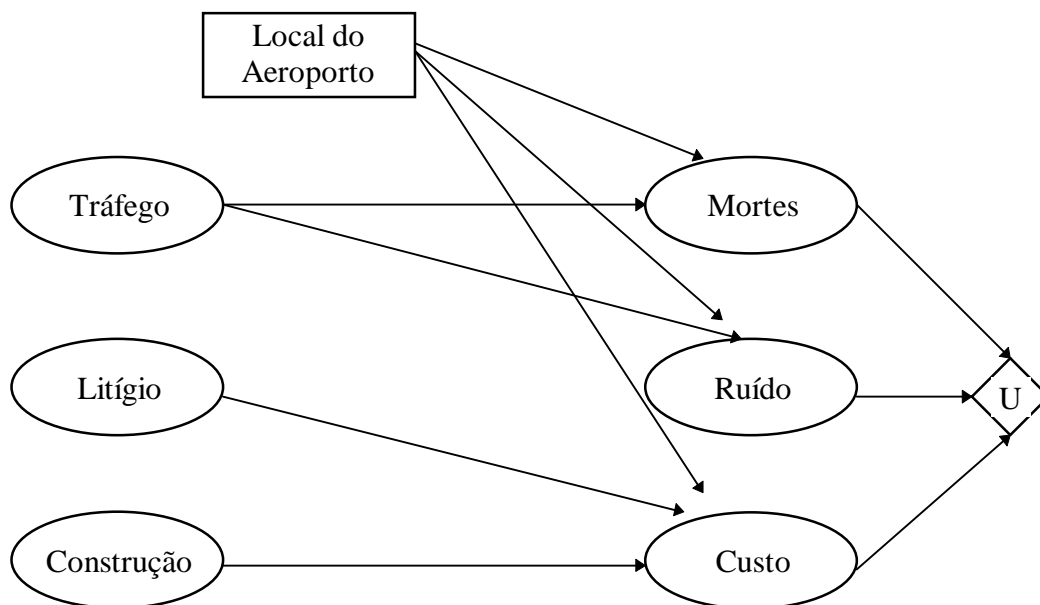
<sup>15</sup> Ou nós de valor. É preferível chama-los nós de utilidade para enfatizar que os seus valores podem representar um loteria, pois nem sempre estaremos utilizando o critério EMV (veja Capítulo 13).

Um nó de utilidade representa o valor esperado da utilidade dados os valores de seus nós pais e corresponde a um objetivo (em um problema de otimização) cujo valor esperado deve ser maximizado. Sem perda de generalidade “é possível assumir que os problemas (de decisão) podem ser formulados como uma rede de decisão com um **único** nó de utilidade, o qual não possui sucessores” [CAS 96]. Seus nós pais são todas as variáveis descrevendo as decisões e variáveis aleatórias que afetam diretamente o cálculo da utilidade.

O algoritmo de avaliação da rede de decisão com apenas um nó de decisão é:

- atribua valores para as variáveis de evidência para o estado corrente,
- para cada valor possível do nó de decisão:
  - atribua esse valor para o nó de decisão,
  - calcule as probabilidades a posteriores para os pais do nó de utilidade,
  - calcule a utilidade resultante da ação,
- retorne a ação com a mais alta utilidade.

A Figura 14.1 apresenta a rede de decisão para o problema de decidir onde construir um novo aeroporto. São considerados os seguintes aspectos: incerteza associada ao custo de construção (custo da terra, distância do centro populacional mais próximo, etc.), nível de tráfego aéreo (correlação com número esperado de morte por acidente por ano, nível de ruído, etc.) e potencial de litígio com a população vizinha ou com ecologistas (correlacionado com o custo de construção). Esses fatores são representados através de uma rede causal (como no caso da construção de uma rede bayesiana). A única variável decisão considerada é o local de construção do aeroporto. A função utilidade depende apenas das variáveis morte (esperada por milhões de passageiros-milha), custo e ruído.



**Figura 14.1** - Rede de Decisão Simples

Existem redes de decisões e algoritmos específicos para problemas mais complexos onde a utilidade de cada ação do agente depende de uma seqüência de decisões e não apenas de uma ação. Se a estrutura do grafo da rede de decisão atender a certas condições, a solução pode ser obtida através de transformações que mudam a topologia da rede (reduzindo o número de nós) e os dados armazenados em cada nó, mas preservam o

valor da solução. As condições que o DAG deve atender são [Shachter-86 Apud CAS 96]:

- c1) os nós de decisão  $D_1, D_2, \dots, D_m$  podem ser ordenados no tempo de forma que  $D_n$  seja o pai de  $D_{n+1}$ , para  $1 \leq n < m$ ;
- c2) a rede não contém ciclos;
- c3) o nó de utilidade é o único nó que não possui sucessor;
- c4) se o nó de probabilidade  $P_i$  não é pai de um nó de decisão  $D_j$  então não deve existir um caminho orientado de  $P_i$  a  $D_j$ , passando só por nós de probabilidade.

Observe que se existir algum nó sem sucessor, exceto o de utilidade, tal nó não influi em nenhum dos outros e pode ser removido do DAG [Shachter-86 Apud CAS 96].

Se o DAG não atender a c4, aplicamos a transformação [Shachter-86 Apud CAS 96]:

- t1) inverter a direção do arco entre dois nós probabilísticos, usando o teorema de Bayes e c4 passa a ser atendida. Ambos herdam os nós pais do outro nó.

Caso atenda às condições acima então existe (pelo menos) um nó cujo único sucessor direto é o nó de utilidade. Podemos iniciar a avaliação da rede através da absorção desse nó, aplicando uma das seguintes transformações [Shachter-86 Apud CAS 96]:

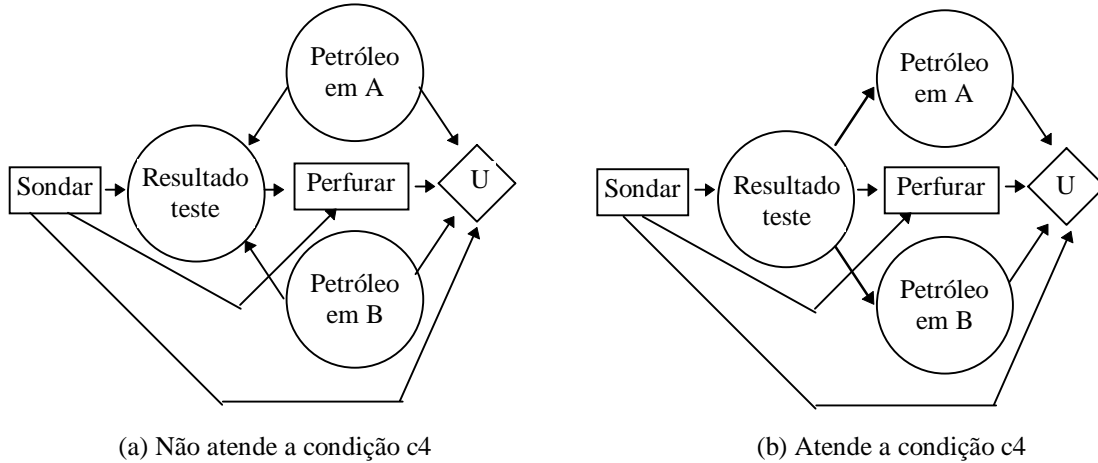
- t2) remover um nó probabilístico após calcular o seu valor esperado e atribuí-lo ao nó de utilidade, seu sucessor. O nó probabilístico deve ser pai somente do nó de utilidade, o qual herda todos os arcos incidentes ao nó removido. Nesse processo são afetados somente o nó absorvido e o nó de utilidade, sendo que as informações e relações entre os nós restantes permanecem inalteradas.
- t3) remover um nó de decisão após maximizar o valor esperado da utilidade do nó de utilidade, condicionada aos valores de seus pais. A decisão  $d^*$  que maximiza o valor esperado dessa utilidade condicional, deve ser registrada como estratégia ótima. O nó de decisão **não** herda novos pais nessa transformação. Esse nó somente poderá ser absorvido se for o pai de um nó de utilidade e todos os outros predecessores do nó de utilidade também forem predecessores do nó de decisão.

Esse processo de redução do DAG continua até obtermos um diagrama com um único nó, o nó de utilidade, representando o valor esperado da utilidade associada à solução do problema se a sequência (estratégia) ótima de decisões for seguida.

A Figura 14.2 ilustra a rede de decisão associada ao seguinte exemplo, extraído de Smith [Smith-88 Apud CAS 96, p.22]. O Governo oferece a uma companhia de petróleo a opção de perfurar o campo A (com 0,4 de probabilidade de encontrar petróleo) ou o campo B (com probabilidade 0,2), sendo a existência de petróleo em um campo independente da existência no outro. As decisões possíveis da companhia são: a) investigar apenas um dos campos, ao custo de US\$ 6 milhões; b) perfurar o campo A ou o B ao custo de US\$ 31 milhões; ou c) não perfurar nenhum dos campos.

Observe que o resultado da investigação do campo não é totalmente preciso. Se ele aconselhar perfurar, a probabilidade de encontrar óleo será de 0,8 e a probabilidade de não encontrar óleo será de 0,4 para qualquer um dos dois campos que tiver sido investigado. Mesmo no caso da perfuração ser desaconselhável existe uma probabilidade baixa de existir óleo e a companhia deve decidir correr ou não o risco de arcar com os custos de perfuração no campo investigado ou no outro campo que não foi testado.

Se existir petróleo, o ganho esperado (não deduzidos os custos de perfuração ou teste) é US\$ 77 milhões para o campo A e de US\$ 195 milhões para o campo B. Se a função de utilidade do dinheiro da companhia for linear, os critérios MEU e EMV são equivalentes.



**Figura 14.2** - Exemplo de Rede de Decisão

Observe que os nós de probabilidade da Figura 14.2(a) representam a distribuição de probabilidades conjunta das variáveis aleatórias T (resultado do teste), A (existência de petróleo no campo A) e B (existência de petróleo no campo B) como:

$$P(T,A,B) = P(T | A,B) \cdot P(A,B) = P(T | A,B) \cdot P(A) \cdot P(B),$$

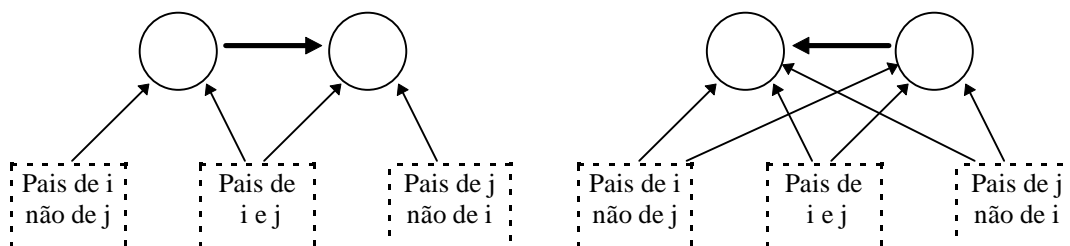
porque A e B são independentes. Com essa representação esses três nós de probabilidade armazenam CPTs que representam respectivamente, as seguintes distribuições de probabilidades condicionais:  $P(T | A,B)$ ,  $P(A)$  e  $P(B)$ .

Com essa configuração existe um caminho orientado quer do nó A quer do nó B para o nó de decisão 'perfurar' embora nenhum deles seja pai deste nó de decisão. Portanto a condição c4 não se mantém. c4 pode ser restabelecida se representarmos a distribuição de probabilidades conjunta da seguinte forma:

$$P(T,A,B) = P(A,B | T) \cdot P(T) = P(A | T) \cdot P(B | T) \cdot P(T).$$

Esta representação implicar inverte os arcos entre os nós A e B e o nó T. Com essa inversão as CPTs armazenadas nos nós de probabilidade T, A e B devem ser alteradas e passam a representar, respectivamente as seguintes distribuições de probabilidades condicionais:  $P(T)$ ,  $P(A | T)$  e  $P(B | T)$ . Esta situação está ilustrada na Figura 14.2(b).

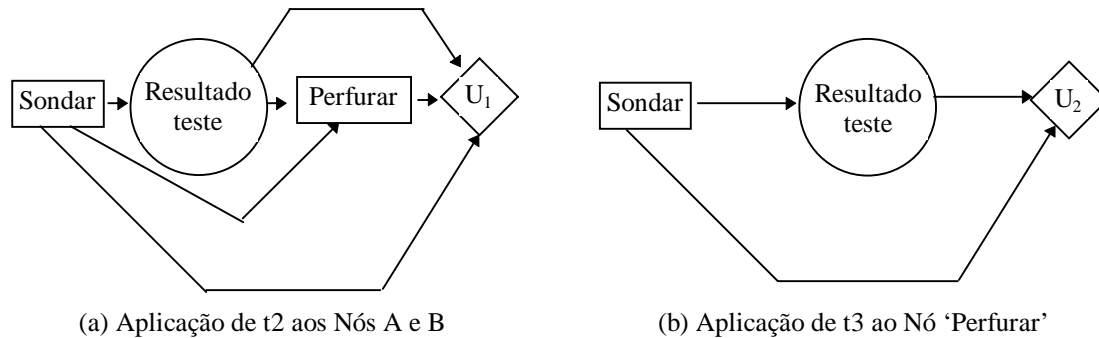
A aplicação da transformação t1, em um caso geral está ilustrado na Figura 14.3 a qual representa a inversão do sentido de um arco e a atualização das respectivas CPTs através da aplicação do teorema de Bayes. Note que cada um dos nós herda os pais do outro nó.



**Figura 14.3** - Transformação t1: Inversão de Arco em Rede de Decisão



Agora podemos aplicar a transformação t2 aos nós de probabilidade A e B visto que ambos somente são pais do nó de utilidade. Para tanto é calculado o valor esperado da utilidade dada as distribuições de probabilidades de A e de B. Esse valor é armazenado no nó de utilidade (que passa a ser denotado por  $U_1$ ), o qual também recebe todos os arcos incidentes aos nós A e de B, sendo estes nós removidos a seguir (Figura 14.4(a)).



**Figura 14.4** - Aplicações das Transformações t2 e t3: Absorção de Nós

Depois da aplicação de t2, como ilustra a Figura 14.2(a), o nó de decisão 'Perfurar' tem como único sucessor direto o nó de utilidade  $U_1$ . Como o conjunto  $C = \{ \text{'Sondar'}, \text{'Resultado do teste'} \}$  dos outros pais de  $U_1$  também são pais de 'Perfurar', a condição de aplicação de t3 é satisfeita. Assim podemos escolher a decisão  $d^*$  (i.e perfurar ou não perfurar) de forma a maximizar o valor esperado da função de utilidade condicionada a C. A essa função utilidade condicional máxima, denotamos  $U_2$ .

Como ilustra a Figura 14.4(b), o nó de probabilidade 'Resultado do teste' atende a condição de aplicação de t2, podendo então ser absorvido. Após esse transformação, o nó de decisão 'Sondar', no diagrama resultante, atende a condição para aplicar t3 e também pode ser removido, resultando em uma rede de decisão com um único nó, o nó de utilidade, o qual armazena a estratégia de decisão ótima. No caso exemplo essa estratégia é investigar o campo B e realizar a perfuração somente se o resultado da sondagem for positivo. Se for achado petróleo, o ganho líquido será de US\$ 158 milhões; caso contrário o prejuízo será de US\$ 37 milhões se o poço tiver sido perfurado e de US\$ 6 milhões se apenas a sondagem tiver sido feita.

### 14.1 Redes de Crenças

As principais abordagens vistas para o raciocínio probabilístico baseado em redes foram redes bayesianas e redes de decisão. Cadeias de grafos, modelos recursivos e redes de Markov [CAS 96] constituem outros formalismos. Normalmente se usa a expressão *redes de crenças* para designar todas elas. Redes de crenças utilizam a teoria de probabilidade para representar incerteza e a vizinhança dos nós em um grafo para expressar a relevância entre variáveis. Desta forma é necessário somente consultar os nós vizinhos do nó associado a uma certa variável para se obter todas as informações que podem ser relevantes; os outros nós podem ser ignorados. Todas as dependências entre as variáveis do domínio de interesse são representadas no grafo por meio de arcos, sendo assumida independência em caso da ausência do arco.

Na apresentação de redes bayesianas e de redes de decisão admitimos que as distribuições de probabilidades permaneciam constantes ao longo do tempo, portanto admitimos

que eram *estacionárias*. No caso geral podemos ter um processo estocástico: o valor de um vetor de variáveis aleatórias no instante  $t$  pode depender dos valores assumidos nos instantes anteriores ou seja  $P[\mathbf{X}_t | \mathbf{X}_{t-1}, \mathbf{X}_{t-2}, \dots, \mathbf{X}_1]$ . Quando essa probabilidade não é influenciada pela história passado (i.é a sequência dos valores prévios) mas apenas pelo instante imediatamente anterior, temos um *processo de Markov* que é caracterizado por  $P[\mathbf{X}_t | \mathbf{X}_{t-1}, \mathbf{X}_{t-2}, \dots, \mathbf{X}_1] = P[\mathbf{X}_t | \mathbf{X}_{t-1}]$ . Para visualizar essa situação podemos imaginar um jogo em um tabuleiro onde o agente inicia em uma certa casa (caracterizada por uma posição  $x, y$ ) e deve atingir uma casa objetivo. Assim  $\mathbf{X}_t$  representa a posição do agente no instante  $t$ , a qual só depende da sua posição no instante anterior. Neste contexto uma ação (decisão) representa um movimento aleatório (para cima, para baixo, para a esquerda, para a direita) a partir da casa em que se está. Uma sequência de decisões é denominada uma *política*. O *estado* pode ser visto como a casa na qual o agente se encontra. No caso geral a decisão a ser tomada pelo agente também depende do estado do ambiente no qual ele atua. Assim ele deve possuir algum mecanismo (sensor) de *percepção* do ambiente. O problema de calcular uma política ótima (p.ex. aquela que minimiza o número de movimentos ou no caso geral maximiza uma certa função utilidade) em um ambiente estocástico *acessível* (i.é aquele no qual o estado pode ser identificado através da percepção do agente no instante atual apenas) é denominado um *MDP Problema de Decisão de Markov* (Markov Decision Problem). Observe que em um processo de Markov a *probabilidade de transição* de um dado estado para outro depende apenas do estado atual e não do histórico prévio.

Em ambientes *não acessíveis* (i.é aqueles nos quais a percepção - informações obtidas através de sensores - do agente num certo instante **não** é suficiente para determinar o estado atual ou as probabilidades de transição associadas) o problema de calcular uma política ótima é denotado POMDP - *MDP Parcialmente Observável* (Partially Observable Markov Decision Problem). Para resolver um MDP ou um POMDP são necessários algoritmos específicos que não serão abordados nesse trabalho. O leitor interessado poderá consultar [Cassandra 1994 Apud RUS 95].

Os algoritmos apresentados para redes bayesianas admitem que o ambiente é imutável (ou pelo menos não influencia as probabilidades condicionais). A evolução do ambiente ao longo do tempo pode ser representada através de DBN - *redes bayesianas dinâmicas* (Dynamic Belief Network). Uma DBN é uma rede bayesiana expandida de forma a conter um nó para cada instante de tempo  $t$ , para cada variável que represente estado do ambiente e analogamente outro nó para cada sensor (de percepção de evidências) do estado do ambiente. É admitido que o processo estocástico é markoviano e estacionário, portanto a evolução do ambiente pode ser modela através de  $P[\mathbf{X}_t | \mathbf{X}_{t-1}]$  e a percepção (observações da variável aleatória  $\mathbf{E}$ ) é assumida ser ocasionada pelo estado atual ou seja  $P[\mathbf{E}_t | \mathbf{X}_t, \mathbf{X}_{t-1}, \dots, \mathbf{X}_1] = P[\mathbf{E}_t | \mathbf{X}_t]$ . O objetivo da manipulação da DBN é calcular a distribuição de probabilidades para o estado no instante  $t$  ou em instantes futuros. Para tanto podem ser utilizados os algoritmos tradicionais para avaliação de redes bayesianas, embora seja altamente ineficiente devido ao grande tamanho de uma DBN. Um algoritmo mais especializado é o seguinte [RUS 95]:

- **Fase de Predição** da distribuição de probabilidades do estado corrente, dado o estado anterior,  $\text{Bel}(\mathbf{X}_t)$ , calculada como a soma das probabilidades de chegar em um dado estado no instante  $t$ , partindo do estado conhecido no instante  $t-1$ :

$$\text{Bel}(\mathbf{X}_t) = \sum_{\mathbf{X}_{t-1}} P(\mathbf{X}_t | \mathbf{X}_{t-1} = \mathbf{x}_{t-1}) \cdot \text{Bel}(\mathbf{X}_{t-1} = \mathbf{x}_{t-1});$$

onde  $\mathbf{x}_{t-1}$  varia ao longo de todos os valores possíveis do vetor de variáveis de estado  $\mathbf{X}_{t-1}$  e  $\text{Bel}(\mathbf{X}_{t-1})$  é a distribuição de probabilidades do vetor de estado em  $t-1$ .

- **Fase de Estimação** da distribuição de probabilidades do vetor de variáveis de estado atual, levando em conta a mais recente observação (percepção) do estado:

$$\text{Bel}(\mathbf{X}_t) = \alpha \cdot P(\mathbf{E}_t | \mathbf{X}_t) \cdot \text{Bel}(\mathbf{X}_t),$$

onde  $\alpha$  é uma constante de normalização. Observe que tal equação reproduz uma atualização bayesiana padrão.

DBN podem ser vistas como um processo de Markov esparsos (i.e. com poucos nós) e têm sido aplicadas a análise de tráfego em rodovias usando visão por computador, em robôs móveis [Huang et al. 1994 e Dean and Wellman 1991 Apud RUS], etc.

Se acrescentarmos nós de utilidade e nós de decisão a uma DBN obtemos uma DDN - *Rede de Decisão Dinâmica* (Dynamic Decision Network). Os algoritmos de avaliação de DDN são basicamente os mesmos vistos para uma rede de decisão convencional. DDN também foram utilizadas por Dean e Wellmann em aplicações com robôs móveis.

As chamadas QPN, *redes probabilísticas qualitativas* (Qualitative Probabilistic Network), propostas por Wellman [WEL 90], constituem uma abstração puramente qualitativa de redes bayesianas que utiliza a noção de influência positiva ou negativa entre variáveis. Os valores numéricos de probabilidades são substituídos por relações qualitativas entre as variáveis. Essas relações podem ser consideradas como limitantes dos valores das probabilidades condicionais, expressando inequações que devem ser satisfeitas pelos diversos elementos da rede [CAS 96]. Uma rede de decisão convencional representa uma particular distribuição de probabilidades conjunta para as variáveis consideradas enquanto uma QPN representa uma família de distribuições conjuntas que verifica as restrições impostas pelas relações qualitativas.  $N^\delta(a,c)$  onde  $\delta \in \{+, -, 0, ?\}$  denota influência qualitativa da variável aleatória  $a$  sobre  $c$ , no sentido de dominância estocástica, significando que para  $a$  crescente a distribuição de probabilidades de  $c$  dado  $a$  é:

$N^+(a,c)$  - não decrescente,

$N^-(a,c)$  - não crescente,

$N^0(a,c)$  - condicionalmente independente,

$N^?(a,c)$  - indeterminada.

Os arcos em uma QPN são rotulados com +, - ou ? sendo que a influência 0 não é representada (pois como no caso de uma rede de decisão padrão a ausência de um arco indica independência condicional). De forma análoga dizemos que a variável  $a$  influencia positivamente o nó de valor  $v$ ,  $N^+(a,v)$ , sse a função de utilidade  $U$  satisfaz  $U(a_1, x) \geq U(a_2, x)$ , onde  $a_1 > a_2$  são valores de  $a$  e  $x$  é o conjunto dos pais de  $v$ , exceto o nó  $a$ . Para a influência negativa ( $\delta = -$ ) a relação entre esses valores de utilidade é  $\leq$ .

Uma outra relação entre dois nós de uma QPN é a *sinergia qualitativa* que descreve a influência de incrementos em duas variáveis  $a$  e  $b$  sobre uma terceira variável  $c$ , sendo denotada  $S^\delta(\{a,b\},c)$ . A sinergia positiva mostra que o efeito combinado de incrementar as variáveis  $a$  e  $b$  sobre a distribuição de probabilidades condicional acumulada de  $c$  é maior do que o efeito obtido ao se incrementar as duas separadamente. Esse efeito é mensurado em termos de incrementos na distribuição de probabilidades condicional acumulada. Assim a sinergia qualitativa  $S^\delta(\{a,b\},c)$  se verifica sse para todo valor  $c_0$  de

$c$ , sendo  $x$  o conjunto dos pais de  $c$  (exceto os nós  $a$  e  $b$ ), para  $a_1 > a_2$  valores de  $a$  e para  $b_1 > b_2$  valores de  $b$ , satisfaz a relação:

$$[F_c(c_0 | a_1 b_1 x) - F_c(c_0 | a_2 b_1 x)] R_\delta [F_c(c_0 | a_1 b_2 x) - F_c(c_0 | a_2 b_2 x)],$$

onde  $R_\delta$  é  $\leq$ ,  $\geq$  ou  $=$  se  $\delta$  for respectivamente,  $+$ ,  $-$  ou  $0$ .

Em muitas situações tal informação é suficiente para tomar a decisão ótima sem necessidade de especificação precisa de valores de probabilidade como exemplificado em [WEL 90] e [CAS 96].

A tese de doutorado da Dra. Maria Inês Castiñeira [CAS 96] focaliza a explicação em sistemas baseados em conhecimentos para apoio a decisão. Assim frente a uma solução ótima de um problema de decisão o usuário poderia formular as questões: “o que aconteceria se ... ?” ou “porque esta é a estratégia ótima?”, sendo que ele geralmente deseja uma explicação no contexto do problema e não uma explicação geral do tipo é ótima porque maximiza a utilidade esperada. Ela descreve o mecanismo de *análise de sensibilidade* (observar o que acontece com a estratégia ótima quando os valores das variáveis do sistema são alteradas) para responder a primeira questão e utiliza o formalismo de QPN para tratar a segunda.

Maria Castiñeira conclui que “a possibilidade de observar graficamente o que acontece com o modelo quando os valores das variáveis são modificadas (em decorrência de uma análise de sensibilidade) permite compreender melhor o problema descobrindo quais as variáveis que influenciam as decisões. Este processo também auxilia a refinar os valores das variáveis envolvidas. Entretanto as QPNs permitem realizar abstrações e simplificações apropriadas do modelo, i.é permitem obter as relações qualitativas do modelo a partir do seu nível quantitativo.” [CAS 96, p.61].

A análise de sensibilidade permite avaliar a importância de cada variável do modelo e eliminar aquelas que não são significativas assim como identificar as principais variáveis para a decisão ótima. A explicação baseada em QPN evidencia a estrutura qualitativa do problema, destacando as propriedades e relações entre as suas variáveis, sendo de mais fácil entendimento para as pessoas. Por outro lado, o uso de QPN possui as seguintes limitações: a) permite identificar apenas as relações monótonas; b) não especificam completamente a distribuição conjunta das variáveis o que pode implicar em perda de informação; c) algumas situações ficam indeterminadas, por exemplo a combinação de influências paralelas de sinal contrário resulta em uma influência indeterminada ( $N^?$ ).

O uso de QPN é inadequado para justificar uma escolha de decisão particular mas é indicado para fornecer restrições úteis sobre um estratégia ótima como função da informação disponível [CAS 96].

## 15 Conclusões

Os esquemas de representação de conhecimento que não consideram o tratamento de incerteza foram apresentados na Parte I, partindo dos formalismos mais genéricos e de aplicação mais ampla (como representação lógica e regras de produção) para aqueles mais específicos e com maior orientação semântica (como scripts e dependência conceitual). De forma geral esses formalismos são apropriados para representar conhecimento quando se dispõe de um modelo completo, consistente e imutável do domínio em questão. As informações disponíveis são completas com relação ao domínio e a adição de novos fatos consistentes à base de conhecimentos não implica a revisão da crença nos fatos existentes (os quais são verdadeiros). Como não existe dúvida com relação a esse tipo de conhecimento, ele é dito **conhecimento certo**.

Em aplicações reais, muitas vezes é necessário raciocinar com falta de informação e atribuir medidas de certeza às crenças em algumas hipóteses para as quais existem certas evidências de apoio. Para raciocínio em certos domínios - diagnósticos médico, detecção de causas de falhas, etc. - é necessário fazer simplificações, tais como deixar fatos desconhecidos (ignorância teórica sobre o domínio), não ditos (ignorância prática devido a falta de evidências sugestivas) ou grosseiramente resumidos (devido ao esforço excessivo necessário para detalhá-los). Portanto lidamos com conhecimento que não é indiscutível mas ao qual está associada alguma medida de incerteza que descreve crenças para as quais existem certas evidências de apoio. Esse tipo de conhecimento é dito **conhecimento incerto**. As fontes de incerteza são: confiabilidade parcial da informação, imprecisão da linguagem de representação do conhecimento utilizado, informação parcial, e agregação ou resumo das informações provenientes de múltiplas fontes. A informação parcial pode ser originada por relações causa-efeito desconhecidas ou não determinísticas e devido a ignorância teórica sobre o domínio.

Os esquemas de representação e tratamento de conhecimento incerto foram apresentados na Parte II, partindo dos formalismos que representam a incerteza de forma simbólica (p.ex. lógicas não-monotônicas) para os que utilizam abordagens numéricas, incluindo representações na forma de quantidade escalares ou intervalares precisas (p.ex. fatores de crença ou teoria da evidência de Dempster-Shafer) até os baseados em conjuntos fuzzy.

Segundo Hecherman, Mamdani e Wellman “nos primórdios da programação baseada em regras, os métodos predominantes para tratamento da incerteza usavam variantes do cálculo de probabilidades para combinar fatores de certeza associados com as regras aplicáveis. Embora fosse reconhecido que o cálculo de fatores de certeza não concordava com a teoria de probabilidade, esses métodos eram não obstante favorecidos porque as técnicas probabilísticas disponíveis naquela época requeriam a especificação de um número intratável de parâmetros (funções de distribuições conjuntas) ou assumir um conjunto irrealístico de relações de independência. Atualmente métodos baseados na teoria de probabilidade novamente começam a ganhar aceitação nas comunidades de ciência da computação e de raciocínio incerto. O rompedor de barreiras foi uma linguagem de modelagem gráfica para representar relacionamentos incertos: as redes bayesianas.” [HEC 95a, p.25-26].

As redes bayesianas são representadas por grafos orientados acíclicos cujos nós são variáveis aleatórias conectados por arcos que indicam uma relação de causalidade direta entre as variáveis. Cada nó tem uma tabela de probabilidades condicionais que quantificam os efeitos que os nós pais (aqueles diretamente conectados) têm sobre ele.

Embora os algoritmos apresentados para redes bayesianas tenham explorado a independência condicional, tornando o problema mais tratável, a inferência probabilística em uma rede bayesiana arbitrária é um problema NP-completo quer se realizada com métodos exatos ou com aproximados (estocásticos). No entanto para muitas aplicações as redes são suficientemente pequenas (ou podem ser simplificadas) de forma tal que a inferência possa ser realizada com um esforço computacional aceitável [HEC 95]. Para os casos nos quais os métodos usuais de inferência são impraticáveis, estão sendo desenvolvidos novos algoritmos apropriados para determinadas topologias de rede [5 e 15 Apud HEC 95] ou para questionamentos específicos [7, 12 ou 14 Apud HEC 95].

A Parte III apresenta o formalismo de rede de decisão, o qual combina ação (descrita pela teoria da decisão) e utilidade (descrita pela teoria de utilidade) com redes bayesianas. A função utilidade permite representar e raciocinar com preferências. Nós de decisão representam ações que podem ser tomadas. Cada estado tem um grau de utilidade, sendo preferido os estados com maior utilidade.

Esse formalismo é utilizado em problemas de decisão, de forma tal que ao se decidir por uma ação, se possa mensurar a utilidade esperada dos estados viáveis, dada uma certa evidência. Também pode ser utilizado para a modelagem de agente de decisão teórica, o qual é capaz de tomar decisões racionais baseado nas suas crenças e na meta (estado) que deseja alcançar, mesmo atuando em um ambiente com incerteza e metas conflitantes.

Como vimos redes de decisão e árvores de decisão são utilizadas em problemas de decisão. No entanto uma árvore representa todas as possíveis combinações de decisões e eventos enquanto a rede representa a relação entre os componentes da decisão e as variáveis aleatórias do problema, constituindo uma representação mais compacta, concisa e intuitiva, proporcionando uma mesma linguagem para quem desenvolve, analisa e utiliza o modelo [CAS 96].

A estrutura gráfica de uma rede de decisão contém suficiente informação, mesmo sem considerar as tabelas associadas a cada nó, facilitando os procedimentos computacionais de resolução do problema. Essa estrutura representa qualquer independência condicional entre variáveis, a sequência temporal das diversas decisões possíveis, e a disponibilidade de informação antes da tomada de uma decisão, permitindo facilmente calcular o valor esperado da utilidade associada a um nó de utilidade.

Assim como no caso de redes bayesianas, a possibilidade de representar as relações entre as variáveis na direção causa-efeito facilita o processo de aquisição de conhecimento (durante a construção da rede), permitindo o envolvimento do usuário na modelagem do problema e na análise da solução.

Por outro lado redes de decisão, além de permitir o tratamento de incerteza, permitem explicitar as diferenças nas predileções e posicionamentos das pessoas frente a uma determinada situação de risco, em problemas de decisão.

Alguns pesquisadores têm questionado a praticidade de se obter probabilidades numéricas com suficiente precisão para criar redes (bayesianas ou de decisão) para grandes aplicações. Obter as probabilidades para problema reais pode ser uma tarefa difícil. As vezes é possível obter dados experimentais a partir dos quais as probabilidades podem ser calculadas. Em geral tais probabilidades devem ser obtidas a partir da opinião do especialista.

Pradhan et al. analisou a sensibilidade de redes bayesianas a probabilidades imprecisas no domínio médico (diagnóstico de doenças do fígado e biliar) concluindo “que diferenças extremas no mapeamento de probabilidades e (a introdução de) grandes quantidades de ruído (aleatório) conduzem somente a modestas reduções na performance do diagnóstico” [PRA 96, p.363].

Uma área atual de pesquisa é a de técnicas de aprendizagem em redes [HEC 95b] [BUN 94] [Spiegelhalter-92, 93 e Gebhardt-95 Apud CAS 96].

Outras áreas de pesquisa são combinação de redes de decisão com outros formalismos (p.ex regras com abdução por cláusulas de Horn [POO 93]), modelagem de agentes de decisão teórica ([CHA 93], [EKE 94] e [ETZ 91]), além do desenvolvimento de novos algoritmos de avaliação de redes, como já citado.

## Bibliografia

- [ABE 94] ABEL, Mara. **Introdução aos Sistemas Especialistas**. Porto Alegre: II/UFRGS, 1994. 89p.
- [BAR 86] BARR, A.; FEIGENBAUM, E. A. **The Handbook of Artificial Intelligence**. v.I. Los Altos: Morgan Kaufmann, 1986. 423p.
- [BHA 86] BHATNAGAR, Raj K.; KANAL, Laveen N. Handling Uncertain Information: A Review of Numeric and Non-numeric Methods. In: KANAL, L.N.; LEMMER, J.F. (Editors). **Uncertainty in Artificial Intelligence**. Amsterdam: Elsevier Science, p.3-26, 1986.
- [BIT 96] BITTENCOURT, Guilherme. **Inteligência Artificial: Ferramentas e Teorias**. Campinas: Instituto de Computação, UNICAMP, 1996.
- [BON 86] BONISSONE, Piero P.; DECKER, Keith S. Selecting Uncertainty Calculi and Granularity: an Experiment in Trading-off Precision and Complexity. In: KANAL, L.N.; LEMMER, J.F. (Editors). **Uncertainty in Artificial Intelligence**. Amsterdam: Elsevier Science, p.217-247, 1986
- [BON 87] BONISSONE, Piero P. Summarizing and Propagating Uncertain Information with Triangular Norms. In: SHAFER, Glenn; PEARL, Judea (Eds.). **Readings in Uncertain Reasoning**. San Mateo: Morgan Kaufmann, 1990, p.239-253.
- [BON 90] BONISSONE, Piero P. Plausible Reasoning. In: SHAPIRO, Stuart C. (Ed.) **Encyclopedia of Artificial Intelligence**. New York: John Wiley, 1990.
- [BRO 91] BROOKS, Rodney A. Intelligence without representation. **Artificial Intelligence (Special Volume Foundations of Artificial Intelligence)**. Amsterdam: Elsevier Science, v.47, January 1991, p.139-159.
- [BUN 94] BUNTINE, Wray L. Operations for Learning with Grapical Models. **Journal of Artificial Intelligence Research**. Morgan Kaufmann, v.2, 1994, p.159-225.
- [CAS 87] CASANOVA, M. A.; GIORNO, F. A. C.; FURTADO, A. L. **Programação em Lógica e a Linguagem Prolog**. São Paulo: E. Blücher, 1987. 461p.
- [CAS 96] CASTIÑEIRA, Maria Inés. **Proposta de Explicação em SAD Utilizando Diagramas de Influências**. São Carlos: Instituto de Física de São Carlos - USP, 1996, 98p (Tese de doutorado).
- [CHA 85] CHARNIAK, Eugene; McDERMOTT, Drew **Introduction to Artificial Intelligence**. Reading: Addison-Wesley, 1985. 701p.
- [CHA 93] CHARNIAK, Eugene. A Bayesian Model of Plan Recognition. Amsterdam: North-Holland, v.64, 1993, p.53-79.



- [COE 88] COELHO, Helder; COTTA, José C. **Prolog by Examples: How to Learn, Teach, and Use It**. Berlin: Springer Verlag, 1988, 392p.
- [DAV 77] DAVIS, Randall; BUCHANAN, Bruce; SHORTLIFFE, E. Production Rules as a Representation for a Knowledge-Based Consultation Program. **Artificial Intelligence**. Amsterdam: North-Holland, v.8, 1977, p.15-45.
- [DOY 79] DOYLE, Jon. A Truth Maintenance System. **Artificial Intelligence**. Amsterdam: North-Holland Publishing, v.12, 1979, p.231-272.
- [DOY 83] DOYLE, Jon. Methodological Simplicity in Expert System Construction: The Case of Judgments and Reasoning Assumptions. In: SHAFER, Glenn; PEARL, Judea (Eds.). **Readings in Uncertain Reasoning**. San Mateo: Morgan Kaufmann, 1990, p.689-693.
- [DUB 88] DUBOIS, Didier; PRADE, Henri. An Introduction to Possibilistic and Fuzzy Logics. In: SHAFER, Glenn; PEARL, Judea (Eds.). **Readings in Uncertain Reasoning**. San Mateo: Morgan Kaufmann, 1990, p.742-761.
- [EKE 94] EKENBERG, Love; DANIELSON, Mats; BOMAN, Magnus. A Tool for Handling Uncertain Information in Multi-Agent Systems. In: DEMAZEAU, Yves; JEAN-PIERRE, Müller; PERRAM, John (Eds.). **Proceedings of MAAMAW'94**. Lindo Center for Applied Mathematics. Odense University, 3-5 August, 1994.
- [ETZ 91] ETZIONI, Oren. Embedding decision-analytic control in a learning architecture. **Artificial Intelligence**. Amsterdam: v.49, 1991, p.129-159.
- [FER 75] FERREIRA, Aurélio Buarque de Holanda. **Novo Dicionário da Língua Portuguesa**. Rio de Janeiro: Nova Fronteira, 1975.
- [FOR 82] FORGY, C. L. RETE: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. **Artificial intelligence**. Amsterdam: North-Holland Publishing, v.19, n.1, 1982, p.17-37.
- [FRI 85] FRIELAND, Peter. Special Section on Architecture for Knowledge-Based Systems. **Communications of the ACM**. New York: ACM Press, v.28, n.9, Sept. 1985. (Guest Editor).
- [FUN 86] FUNG, Robert M; CHONG, Chee Yee. Metaprobability and Dempster-Shafer in Evidential Reasoning. In: KANAL, L.N.; LEMMER, J.F. (Editors). **Uncertainty in Artificial Intelligence**. Amsterdam: Elsevier Science, p.295-302, 1986
- [GRO 86] GROSOFF, Benjamin N. Evidential Confirmation as Transformed Probability In: In: KANAL, L.N.; LEMMER, J.F. (Editors). **Uncertainty in Artificial Intelligence**. Amsterdam: Elsevier Science, p.153-166, 1986
- [HAY 85] HAYES-ROTH, Frederick. Rule-Based Systems. **Communications of the ACM**. New York: ACM Press, v. 28, n. 9, Sept. 1985, p.921-932.

- [HEC 86] HECKERMAN, David. Probabilistic Interpretation for MYCIN's Certainty Factors. In: KANAL, L.N.; LEMMER, J.F. (Eds). **Uncertainty in Artificial Intelligence**. Amsterdam: Elsevier Science, p.167-196, 1986.
- [HEC 95] HECKERMAN, David; WELLMAN, Michael P. Bayesian Networks. In: **Communications of the ACM**. New York: ACM Press, v.38, n.3, March 1995, p.27-30.
- [HEC 95a] HECKERMAN, David; MAMDANI, E.H "Abe", WELLMAN, Michel P. Real-World Applications of Bayesian Networks. In: **Communications of the ACM**. New York: ACM Press, v.38, n.3, March 1995, p.24-26.
- [HEC 95b] HECKERMAN, David. **A Tutorial on Learning Bayesian Networks**. Technical Report MSR-TR-95-06. Advanced Technology Division, Microsoft Research. Redmond: Microsoft, 1995. 40p.
- [KIR 91] KIRSH, David. Foundations of AI: the big issues. **Artificial Intelligence (Special Volume Foundations of Artificial Intelligence)**. Amsterdam: Elsevier Science, v.47, January 1991, p.3-30.
- [LAK 94] LAKEMEYER, Gerhard; NEBEL, Bernhard. Foundations of Knowledge Representation and Reasoning. In: LEKEMEYER, G.; NEBEL, B. (Editors). **Foundations of Knowledge Representation and Reasoning**. Berlin: Springer-Verlag, 1994, p.1-12.
- [MAT 91] MATTOS, Nelson Mendonça. **An Approach to Knowledge Base Management** (Lecture Notes in Artificial Intelligence, n.513). Berlin: Springer-Verlag, 1991. 259p.
- [McC 69] McCARTHY, John; HAYES, P. J. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: MICHIE, D.; MELTZER, B. (Eds). **Machine Intelligence**. Edimburgh: Edimburgh University Press, v.4, 1969, p.463-502.
- [McC 77] McCARTHY, John. Epistemological Problems of Artificial Intelligence. In: **Proceedings of The Fifth International Joint Conference on Artificial Intelligence - IJCAI**, p.1038-1044, 1977.
- [McC 80] McCARTHY, John. Circumscription - a Form of Non-Monotonic Reasoning. **Artificial Intelligence**. Amsterdam: North-Holland, v.13, n.1,2, April 1980, p.27-39.
- [McC 86] McCARTHY, John. Applications of Circumscription to Formalizing Commonsense Knowledge. **Artificial Intelligence**. Amsterdam: North-Holland, v.28, 1986, p.89-116.
- [McD 80] McDERMOTT, D.; DOYLE, J. Non-monotonic Logic I. **Artificial Intelligence**. Amsterdam: North-Holland, v.13, n.1,2, April 1980, p.41-72.
- [NEW 76] NEWELL, Allen; SIMON, Herbert. A. Computer Science as Empirical Inquiry: Symbols and Search. **Communications of the ACM**. New York: ACM Press, v.19, n.3, 1976, p.113-126.

- [NOR 68] NORTH, D. Warner. A Tutorial Introduction to Decision Theory. In: SHAFER, Glenn; PEARL, Judea (Eds). **Readings in Uncertain Reasoning**. San Mateo: Morgan Kaufmann, 1990. p.68-78.
- [PAL 89] PALAZZO, Luiz Antônio M. **RHESUS: Um Modelo Experimental para Representação do Conhecimento**. Porto Alegre: CPGCC da UFRGS, 1989. (Trabalho Individual I). 115p.
- [PEA 88] PEARL, Judea. **Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference**. San Mateo: Morgan Kaufmann, 1988; revised second printing, 1991. 571p.
- [POO 93] POOLE, David. Probabilistic Horn Abduction and Bayesian Networks. **Artificial Intelligence**. Amsterdam: North-Holland, v.64, 1993, p.81-126.
- [PRA 96] PRADHAN, Malcolm; HERION, Max; PROVAN, Gregory; Del FAVERO, Brendan; HUANG, Kurt. The Sensitivity of Belief Networks to Imprecise Probabilities: an Experimental Investigation. **Artificial Intelligence**. Amsterdam: North-Holland, v.85, 1996, p.363-397.
- [REÁ 92] REÁTEGUI, Eliseo B. **Representação de Conhecimento e Aprendizado em Sistemas Especialistas: a Arquitetura de um Sistema Híbrido**. Porto Alegre: CPGCC da UFRGS, 1991. (Trabalho Individual I). 66p.
- [REI 80] REITER, Raymond. A Logic for Default Reasoning. **Artificial Intelligence**. Amsterdam: North-Holland Publishing, v.13, n.1,2, April 1980, p.81-132.
- [REI 87] REITER, Raymond. Nonmonotonic Reasoning. In: SHAFER, Glenn; PEARL, Judea (Eds). **Readings in Uncertain Reasoning**. San Mateo: Morgan Kaufmann, 1990. p.637-656.
- [RIC 94] RICH, Elaine; KNIGHT, Kevin. **Inteligência Artificial**. 2. ed. São Paulo: Makron Books, 1994. 722p.
- [RUS 95] RUSSEL, Stuart; NORVIG, Peter. **Artificial Intelligence: a Modern Approach**. New Jersey: Prentice-Hall, 1995. 932p.
- [SAN 96] SANDRI, Sandra. **Numerical Models for the Treatment of Imperfect Information in Knowledge-Based Systems**. Tutorial presented in the XIII Brazilian Symposium on Artificial Intelligence, held in Curitiba, from 23-25 October, 1996.
- [SHA 90] SHAFER, Glenn; PEARL, Judea. Non-numerical Approaches to Plausible Inference. In: SHAFER, G.; PEARL, Judea (Eds.). **Readings in Uncertain Reasoning**. San Mateo: Morgan Kaufmann, 1990, p.625-628.
- [SHO 75] SHORTLIFFE, Edward H.; BUCHANAN, Bruce G. A Model of Inexact Reasoning in Medicine. In: BUCHANAN, B.G.; SHORTLIFFE, E.H. **Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project**. Reading: Addison-Wesley, 1984, p.233-262.

- [SHO 94] SHOHAM, Yoav. **Artificial Intelligence Techniques in Prolog**. San Francisco: Morgan Kaufmann, 1994. 337p.
- [SIL 94] SILVA, Wagner Teixeira; MILIDIÚ, Ruy Luiz. Decision Making with Belief Function. **Proceedings of the Eleventh Brazilian Symposium on Artificial Intelligence**. Fortaleza: LIA/UFCE, p.483-491, 1994. 581p.
- [SUL 85] SULLIVAN, Michael; COHEN, Paul R. An Endorsement-Based Plan Recognition Program. In: SHAFER, G.; PEARL, Judea (Eds.). **Readings in Uncertain Reasoning**. San Mateo: Morgan Kaufmann, 1990, p.672-676.
- [TOU 84] TOURETZKY, David S. Implicit Ordering of Defaults in Inheritance Systems. In: SHAFER, Glenn; PEARL, Judea (Eds.). **Readings in Uncertain Reasoning**. San Mateo: Morgan Kaufmann, 1990. p.668-671.
- [TUR 84] TURNER, R. **Logics for Artificial Intelligence**. Great Britain: Ellis Horwood, 1984.
- [VIC 90] VICCARI, Rosa Maria. **Inteligência Artificial: Representação do Conhecimento**. Porto Alegre: II/UFRGS, 1990, 71p. (Folheto)
- [WAL 96] WALLEY, Peter. Measures of Uncertainty in Expert Systems. **Artificial Intelligence**. Amsterdam: Elsevier, v.83, 1996, p. 1-58.
- [WEL 90] WELLMAN, Michel P. Fundamental Concepts of Qualitative Probabilistic Networks. **Artificial Intelligence**. Amsterdam: Elsevier, v.44, 1990, p. 257-303.
- [WIN 84] WINSTON, Patrick. **Artificial Intelligence**. Reading: Addison-Wesley, 1984. (2nd Edition).
- [ZAD 78] ZADEH, Lotfi A. Fuzzy Sets as a Basis for a Theory of Possibility. In: **Fuzzy Sets and Systems**. Amsterdam: North-Holland Publishing, v.1, 1978, p.3-28.