MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO DE PESQUISAS ESPACIAIS

AUTORIZAÇÃO PARA PUBLICAÇÃO
AUTHORIZATION FOR PUBLICATION

PALAVRAS CHAVES/KEY WORDS (AUTORES/AUTHORS)

MOTOR DE INFERÊNCIA
SISTEMAS ESPECIALISTAS
INTELIGÊNCIA ARTIFICIAL

AUTORIZADA POR/AUTHORIZED BY

Marco Antonio Raupp
Diretor Geral

AUTOR RESPONSÁVEL / RESPONSIBLE AUTHOR

Eason Luiz F. Senne

DISTRIBUIÇÃO/DISTRIBUTION

- [ ] INTERNA / INTERNAL
- [x] EXTERNA / EXTERNAL
- [ ] RESTRITA / RESTRICTED

REVISADA POR / REVISED BY

Celso Renna e Souza

CDU/UDC

681.3.019

DATA/DATE

Janeiro, 1988

PUBLICAÇÃO Nº / PUBLICATION NO

INPE-4465-TDL/321

ORIGEM / ORIGIN

PG/LAC

PROJETO / PROJECT

PHR/CAP

Nº DE PAG. / NO OF PAGES

176

ULTIMA PAG. / LAST PAGE

C.3

VERSÃO / VERSION

-

Nº DE MAPAS / NO OF MAPS

-

TÍTULO/TITLE

MOTOR DE INFERÊNCIA PARA SISTEMAS ESPECIALISTAS

AUTORES/AUTHORSHIP

Eason Luiz França Senne

RESUMO - NOTAS / ABSTRACT - NOTES

Neste trabalho propõe-se o núcleo de uma ferramenta para a construção de sistemas especialistas. Essencialmente, propõe-se uma linguagem para descrever as regras de decisão que irão compor o banco de conhecimento de um sistema especialista e um mecanismo de inferência controlado por meta-conhecimento e que pode ser adaptado às características específicas de um domínio de aplicação. Propõe-se também o emprego de um esquema de rede semântica para representar o conhecimento estruturado e bem estabelecido do domínio e para fornecer as informações necessárias para o estabelecimento de contextos adequados à utilização de regras de decisão. Neste trabalho propõe-se também uma abordagem para o tratamento de incerteza em sistemas especialistas, com base na teoria de Dempster-Shafer.

OBSERVAÇÕES / REMARKS

Tese de Doutorado em Computação Aplicada, aprovada por Banca Examinadora em 01 de dezembro de 1987, no Instituto de Pesquisas Espaciais.

Aprovada pela Banca Examinadora
em cumprimento a requisito exigido
para a obtenção do Título de Doutor
em Computação Aplicada

Dr. Flávio Roberto Dias Velasco — Presidente

Dr. Celso de Renna e Souza — Orientador

Drª Maria Carolina Monard — Membro da Banca -convidada-

Dr. Eleri Cardoso — Membro da Banca -convidado-

Dr. Paulo Ouvera Simoni — Membro da Banca

Dr. Valter Rodrigues — Membro da Banca

Candidato: Edson Luiz França Senne

São José dos Campos, 01 de dezembro de 1987

Ao amor de

Lígia,
Luís Francisco e
Thaís Helena

e às amizades de

Acioli,
Alcir,
Guilherme,
Tuca e
Vijay.

ACRADECIMENTOS

Sou muito grato,

ao Dr. Celso de Renna e Souza, pela orientação e apoio,

aos meus companheiros do grupo de Inteligência Artificial, especialmente a Sandra Aparecida Sandri e Pedro Paulo Balbi de Oliveira, pelas críticas e sugestões,

a Eunice Kinuyo Odashima, pela inestimável ajuda na depuração dos programas,

ao Instituto de Pesquisas Espaciais, por tornar possível a realização deste trabalho.

RESUMO

Neste trabalho propõe-se o núcleo de uma ferramenta para a construção de sistemas especialistas. Essencialmente, propõe-se uma linguagem para descrever as regras de decisão que irão compor o banco de conhecimento de um sistema especialista e um mecanismo de inferência controlado por meta-conhecimento e que pode ser adaptado às características específicas de um domínio de aplicação. Propõe-se também o emprego de um esquema de rede semântica para representar o conhecimento estruturado e bem estabelecido do domínio e para fornecer as informações necessárias para o estabelecimento de contextos adequados à utilização de regras de decisão. Neste trabalho propõe-se também uma abordagem para o tratamento de incerteza em sistemas especialistas, com base na teoria de Dempster-Shafer.

ABSTRACT

In this work, a kernel of a tool for building expert systems is proposed. In essence, it is proposed a language to express decision rules that will be stored in the system's knowledge base and a meta-knowledge-controlled inference mechanism that uses forward as well as backward chaining and can be adapted to the characteristics of specific domains. It is also proposed the use of a semantic network to store structured, well-established knowledge of a domain and to provide informations in order to determine suitable contexts for decision rules application. In addition, this work proposes an approach to inexact reasoning based on the Dempster-Shafer theory.

SUMÁRIO

Pág.

LISTA DE FIGURAS xi

CAPÍTULO 1 - INTRODUÇÃO

1.1 - O problema em estudo 1
1.2 - Engenharia de conhecimento 4
1.2.1 - Representação de conhecimento 5
1.2.2 - Resolução de problemas 7
1.3 - Arquitetura de sistemas especialistas 10
1.4 - O sistema DIAGNÓSTICO 13
1.5 - Abordagem proposta: o sistema MOTIN 15
1.6 - Organização do documento 24

CAPÍTULO 2 - REPRESENTAÇÃO DE CONHECIMENTO NO SISTEMA MOTIN

2.1 - Introdução 27
2.2 - Armazenagem de conhecimento em regras de decisão 28
2.3 - Atributos, predicados e evidências 32
2.4 - Armazenagem de conhecimento em rede semântica 34
2.5 - Aquisição de conhecimento no sistema MOTIN 38
2.6 - Resumo 52

CAPÍTULO 3 - CONTEXTOS DE APLICAÇÃO DE REGRAS DE DECISÃO

3.1 - O problema do contexto de aplicação 55
3.2 - Grafos de contexto de uma regra 56
3.3 - Determinação de grafos de contexto 57
3.4 - Estabelecimento de contextos de aplicação 63
3.5 - Resumo 69

CAPÍTULO 4 - TRATAMENTO DA INCERTEZA E COMBINAÇÃO DE EVIDÊNCIAS

4.1 - Introdução 71
4.2 - Fundamentos da teoria de Dempster-Shafer 73
4.3 - Regras de decisão como funções de atribuição de crença 79
4.4 - Propagação de crença no sistema MOTIN 81
4.5 - Evidências de atributos numéricos 85
4.6 - Resumo 89

CAPÍTULO 5 - MECANISMO DE INFERÊNCIA E CONTROLE DE RACIOCÍNIO

5.1 - Introdução 91
5.2 - Estruturas para utilização de regras de decisão 93
5.3 - Emprego de meta-regras para controle do raciocínio 101
5.4 - Mecanismo de inferência do sistema MOTIN 105
5.5 - Sessão de consulta ao sistema MOTIN 110
5.6 - Resumo 118

CAPÍTULO 6 - CONCLUSÃO

6.1 - Comparação com outras abordagens 121
6.2 - Estágio atual e extensões futuras 129

REFERÊNCIAS BIBLIOGRÁFICAS 133

APÊNDICE A - ALGUNS RESULTADOS ACERCA DO MODELO DE VERDADE
APÊNDICE B - REGRAS DE DECISÃO PARA O DOMÍNIO DE METEOROLOGIA
APÊNDICE C - IMPLEMENTAÇÃO DO SISTEMA MOTIN

LISTA DE FIGURAS

Pág.

1.1 - Estrutura básica de um sistema especialista 2
1.2 - Arquitetura do sistema DIAGNÓSTICO 13
1.3 - Representação estruturada de conceito 19
1.4 - Àrvore de contextos 21
1.5 - Hierarquia de objetos 22
1.6 - Relacionamento de objetos 23
2.1 - Rede semântica para o domínio de Meteorologia 35
2.2 - Representação de evidências 37
3.1 - Estruturação de objetos na rede semântica 57
3.2 - Grafo de objetos 58
3.3 - Grafo de objetos modificado 58
3.4 - Grafos de contexto 59
3.5 - Grafo de instâncias 65
3.6 - Grafo de contexto 66
3.7 - Exemplo de pergunta para instanciar objeto 66
3.8 - Grafo de instâncias atualizado 67
4.1 - Estruturação de valores 73
4.2 - Classes de valores 83
5.1 - Paradigma de resolução de problema 92
5.2 - Arquitetura do sistema MOTIN 94
6.1 - Um sofisticado sistema baseado em regras 127
C.1 - Organização do sistema MOTIN C.1

CAPÍTULO 1

INTRODUÇÃO

1.1 – O PROBLEMA EM ESTUDO

O paradigma atual para a construção de programas de **Inteligência Artificial** é o de **sistemas baseados em conhecimento.** Assim, um sistema de análise de cenas por computador deve ter conhecimento sobre os objetos do domínio de aplicação, para que ele possa orientar o processo de análise; um sistema de entendimento de linguagem natural deve possuir conhecimento sobre o universo de discurso para compreender as sentenças que ele processa; um sistema de diagnóstico médico necessita conhecer as características das doenças para poder diagnosticá-las e sugerir terapias apropriadas.

O desenvolvimento destes sistemas fez emergir na inteligência artificial uma área separada de estudo – **representação de conhecimento** – que se preocupa com duas questões fundamentais: como armazenar e como manipular o conhecimento que tais sistemas necessitam para se comportarem inteligentemente.

Uma classe de sistemas baseados em conhecimento vem, nos últimos anos, ganhando grande destaque: a dos **sistemas especialistas,** que são assim denominados porque resolvem problemas difíceis e requerem, para isto, o conhecimento de pessoas com grande proficiência no domínio destes problemas.

A diferença principal entre sistemas especialistas e programas de aplicação tradicionais é que nesses sistemas o conhecimento específico para a resolução de problemas é estabelecido como uma entidade separada (**base de conhecimento**) ao invés de aparecer implicitamente como parte da codificação do programa, misturado

portanto com as instruções de controle. Nos sistemas especialistas, a base de conhecimento é manipulada por uma estrutura de controle claramente identificável: o **motor de inferência.**

Essa arquitetura básica dos sistemas especialistas, que estabelece uma separação muito nítida entre a base de conhecimento e os procedimentos encarregados de explorá-la, é conseqüência de uma série de razões (Hayes-Roth et alii, 1983), dentre as quais pode-se destacar que para muitos problemas interessantes não se conhece uma solução algorítmica tratável. Para tais problemas é mais adequado um tratamento heurístico (Buchanan, 1985) utilizando o conhecimento disponível para fazer inferências plausíveis. Numa abordagem deste tipo, tal arquitetura é interessante uma vez que, flexível, permite que o conhecimento seja alterado e estendido facilmente, aumentando a capacidade dos sistemas para resolver problemas.

A Figura 1.1 ilustra a estrutura básica de um sistema especialista.

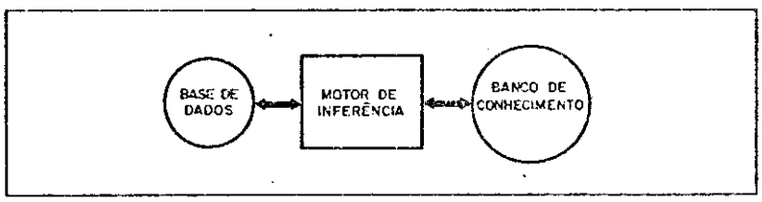

Fig. 1.1 - Estrutura básica de um sistema especialista

As funções dos componentes dessa estrutura são as seguintes:

BASE DE DADOS: armazenar dados para um problema particular, resultados intermediários e histórico do processamento efetuado;
MOTOR DE INFERÊNCIA: decidir sobre como utilizar o conhecimento disponível e aplicá-lo para resolver o problema;
BASE DE CONHECIMENTO: armazenar fatos e heurísticas de resolução de problema associadas ao domínio de aplicação.

Em sistemas práticos é comum acrescentar a essa estrutura básica, outros componentes, como por exemplo, um módulo para facilitar o processo de aquisição de conhecimento e um módulo capaz de fornecer aos usuários do sistema, boas justificativas e explicações para as ações tomadas. As tarefas destes módulos podem envolver, até mesmo, o processamento de linguagem natural (Oliveira, 1986a). Pode-se imaginar também um módulo de aprendizado, capaz de enriquecer e reestruturar os elementos da base de conhecimento do sistema (Silva, 1983; Oliveira, 1986b).

O esforço dispendido na construção de um sistema especialista, particularmente na representação de conhecimento de especialistas humanos, depende da ferramenta adotada para construí-lo. Uma classe de ferramentas - sistemas do tipo esqueleto (Barstow et alii, 1983) - utilizam um mesmo motor de inferência para a criação de vários sistemas especialistas que diferem uns dos outros apenas no conteúdo de suas bases de conhecimento.

No presente trabalho propõe-se o núcleo de uma ferramenta do tipo esqueleto para a construção de sistemas especialistas. Fundamentalmente procura-se estabelecer uma ferramenta com características de generalidade tanto com relação aos esquemas de representação de conhecimento como, na forma de utilizá-los para realizar inferências.

Desta forma, propõe-se uma representação por regras de decisão um pouco mais geral do que a utilizada pelos sistemas especialistas atuais e na qual permite-se facilmente a incorporação de primitivos conceituais. Propõe-se também o emprego de um esquema de rede semântica para representar o conhecimento melhor estruturado do domínio de aplicação e no qual é possível definir a semântica de novas relações. Consegue-se assim o que constitui uma tendência atual no desenvolvimento de sistemas especialistas: a separação entre o conhecimento empírico e o conhecimento bem estabelecido de um domínio de aplicação, com o emprego de esquemas de representação flexíveis e adequados a cada caso.

Com relação aos procedimentos de inferência, propõe-se um mecanismo controlado por meta-conhecimento e que pode utilizar as regras de decisão progressiva, retroativamente ou mesmo de uma forma inovativa, em que estas estratégias se combinam.

Outras características da ferramenta proposta neste trabalho e que também constituem avanços ao estado atual das pesquisas sobre sistemas especialistas são:

- o emprego do conhecimento bem estabelecido do domínio para estabelecer contextos adequados de aplicação do conhecimento de resolução de problema;
- uma abordagem para o tratamento da informação incerta, com a utilização de um mecanismo de combinação de crenças que permite o raciocínio inexato em níveis variados de abstração.

1.2 - ENGENHARIA DE CONHECIMENTO

De uma forma geral (Buchanan et alii, 1983), para desenvolver um sistema especialista é preciso especificar, sobre o domínio de aplicação:

- o **conhecimento estrutural**: identificação de termos, conceitos, relacionamentos e associações entre conceitos e determinação de esquemas de representação; e
- o **conhecimento estratégico**: identificação de técnicas básicas de como usar o conhecimento estrutural para resolver o problema.

Para isto uma nova área de estudos vem ganhando corpo: a **engenharia de conhecimento** (Hayes-Roth, 1984).

Discute-se sucintamente nesta seção, algumas abordagens de representação de conhecimento e de resolução de problemas utilizadas

atualmente em engenharia de conhecimento, para a construção de sistemas especialistas.

1.2.1 – REPRESENTAÇÃO DE CONHECIMENTO

Os primeiros esforços no desenvolvimento de esquemas de representação de conhecimento surgiram no final da década de 1960, devido a uma mudança de orientação da pesquisa em inteligência artificial, quando passou-se dos esquemas de resolução de problemas de **propósito geral** (Newell and Simon, 1963), para os enfoques baseados no conhecimento **específico** de um domínio de aplicação. A partir daí, uma grande variedade de maneiras de representar conhecimento foi proposta e explorada, as quais são largamente repertoriadas na literatura (Brachman and Smith, 1980; Barr and Feigenbaum, 1981; Rich, 1983; Chouraqui et alii, 1985). Apesar de haver interação entre representação de conhecimento e psicologia cognitiva (Norman and Rumelhart, 1975), ainda não é possível mostrar que um esquema represente aspectos da memória humana melhor do que outros.

Os principais esquemas de representação de conhecimento utilizados atualmente são:

LÓGICA DE PREDICADOS. A característica mais importante da lógica de predicados de primeira ordem é que as deduções são garantidamente corretas (desde que os axiomas não sejam conflitantes) e portanto as bases de conhecimento e de dados de um sistema especialista podem ser mantidas logicamente consistentes. A utilidade da lógica como técnica de representação de conhecimento tornou-se evidente devido à exploração do princípio da resolução (Robinson, 1965) em prova automática de teoremas (Nilsson, 1971).

REPRESENTAÇÃO PROCEDIMENTAL. Na representação procedimental o conhecimento está contido em **procedimentos**, pequenos programas que "sabem" como proceder em situações bem especificadas. Este tipo de

técnica de representação surgiu da tentativa de codificar o controle do processo de prova de teoremas, da qual surgiram, dentre outras, as linguagens de programação PLANNER e SAIL (Bobrow and Raphael, 1974) e mais recentemente, a linguagem PROLOG (Kowalski, 1979). Muitas representações procedimentais foram também influenciadas pela linguagem de programação LISP (Winston and Horn, 1981). A grande desvantagem da representação procedimental está na dificuldade de entender e modificar a base de conhecimento de um sistema que a utiliza.

REDE SEMÂNTICA. O esquema de rede semântica foi desenvolvido inicialmente como um modelo da memória associativa humana (Quillian, 1968). Atualmente existem muitos tipos de redes semânticas (Findler, 1979) mas, basicamente, elas consistem em:

- uma estrutura de nós (representando objetos, conceitos, ...) e de ligações entre nós (representando seus relacionamentos);
- um conjunto de procedimentos especializados que operam sobre esta estrutura de dados para realizar inferências.

Uma característica importante da representação de conhecimento por rede semântica é que as associações importantes e os fatos relevantes acerca de um objeto ou conceito podem ser inferidos diretamente dos nós aos quais ele está ligado, sem que haja necessidade de uma grande busca. A utilização de redes semânticas é interessante quando deseja-se representar conhecimento bem estruturado, como por exemplo, em sistemas de análise de cenas (Simoni, 1986).

SISTEMA DE PRODUÇÃO. Os sistemas de produção já foram utilizados como modelos de cognição humana (Newell and Simon, 1972) e têm sido muito empregados em programas de inteligência artificial, notadamente em sistemas especialistas. A idéia básica dos sistemas de produção como técnica de representação (Davis et alii, 1977) é constituir a base de conhecimento como um conjunto de **regras de decisão** na forma de pares condição-ação, que são interpretados como "se a condição ocorre, então

execute a **ação"**. A utilidade deste formalismo vem do fato de que as condições de aplicabilidade de cada regra estão estabelecidas explicitamente (na sua parte de condição) e também porque as interações entre regras são mínimas. Estas características são importantes porque facilitam o entendimento e a modificação de sistemas especialistas com grandes bases de conhecimento, além de permitirem facilmente o desenvolvimento de capacidade explanatória (Wallis and Shortliffe, 1982).

"FRAME". Um "frame" (Minsky, 1975) é, basicamente, uma estrutura de dados que inclui tanto informação declarativa quanto procedimental, ligadas por relações predefinidas, sendo útil para representar conhecimento sobre conceitos ou situações padronizadas. Quando um conceito ou situação é reconhecida, os campos ("slots") do "frame" podem ser preenchidos com valores associados a esta situação particular ou então, podem "herdar" valores preestabelecidos para outros campos.

Existem, evidentemente, outras técnicas de representação de conhecimento. Em sistemas práticos é comum a utilização de vários desses esquemas simultaneamente ou mesmo de esquemas "mistos" nos quais "fundem-se" características de várias técnicas, como por exemplo, em um sistema de produção no qual as regras são compostas por "frames" (que especificam suas partes de condição e de ação) e classificadas numa forma hierárquica (Bittencourt, 1986).

1.2.2 - RESOLUÇÃO DE PROBLEMAS

As aplicações de sistemas especialistas são de tal natureza que, tipicamente, não é conhecida uma seqüência precisa de passos necessários para resolver o problema. Conseqüentemente, é preciso, a todo instante, procurar num espaço de muitos caminhos alternativos, aqueles que levam a uma solução adequada. Por outro lado, a exploração do espaço de busca (em geral, sujeito à explosão combinatória) deve

ser feita eficientemente, isto é, com um volume razoável de recursos (tempo e memória).

O tipo de programa adequado para tratar tais situações é o de uma coleção de **módulos dirigidos por padrões** (Hayes-Roth and Waterman, 1977; Waterman and Hayes-Roth, 1978). Estes módulos não são invocados por outros módulos como é comum na programação tradicional. Ao invés disto, são ativados por uma estrutura de controle sempre que determinadas condições são satisfeitas pelo estado de uma base de dados. Isto ajuda o desenvolvimento do procedimento heurístico de resolução de problema porque a revisão de um módulo não afeta os demais.

Um sistema especialista, na tentativa de resolver um problema, deve executar uma série de ações. Cada uma dessas ações aplica algum módulo dirigido por padrões - ou **fonte de conhecimento** - e, em geral, efetua uma **transição**, isto é, altera o estado de sua base de dados, gerando um novo elemento de solução ou modificando algum elemento já existente.

A cada passo deste processo, muitas dessas ações são possíveis e o sistema deve decidir qual delas executar. Para isto o sistema deve determinar o seu próprio comportamento, ou seja, deve decidir que subproblemas resolver, que conhecimento usar, que métodos de resolução e estratégias aplicar, como avaliar soluções alternativas, como saber quando os subproblemas específicos estão resolvidos e sob que circunstâncias interromper sua atenção para selecionar um novo subproblema.

A estrutura de controle para resolução de problemas utilizada pelos sistemas especialistas é constituída de 3 componentes básicos (Gevarter, 1983): direção de busca, estratégia de controle e transformação do espaço de busca.

DIREÇÃO DE BUSCA. O espaço de soluções pode ser explorado de uma forma progressiva ("forward") ou de uma forma **retroativa** ("backward"). No

primeiro caso tem-se uma busca dirigida por dados e no segundo, uma busca dirigida por objetivos. Outros enfoques para a direção de busca são, por exemplo, a **busca bidirecional**, na qual as duas direções - progressiva e retroativa - se intercalam, ou a **busca dirigida por eventos**, na qual o tipo de uma transição feita determina a ativação da próxima fonte de conhecimento. Podem-se citar alguns exemplos da utilização dessas direções de busca em sistemas especialistas já desenvolvidos: sistema R1 (McDermott, 1982) - busca progressiva; sistema MYCIN (Shortliffe, 1976) - busca retroativa; sistema MELD (Thompson and Wojcik, 1984) - busca bidirecional; sistema CRYSALIS (Engelmore and Terry, 1979) - busca dirigida por eventos.

ESTRATÉGIA DE CONTROLE. Uma estratégia de controle para resolução de problema é a utilização **exaustiva** das fontes de conhecimento. Esta técnica entretanto, só é factível em problemas de complexidade limitada, caracterizados por espaços de busca pequenos. Problemas intrinsicamente difíceis requerem estratégias de controle mais sofisticadas. Algumas dessas estratégias são: **geração e teste**, na qual utiliza-se um modelo para a geração de possíveis soluções que em seguida, são testadas com os dados disponíveis para o problema; **raciocínio plausível**, que consiste basicamente em fazer certas conjecturas sobre estados no espaço de busca, propagar as conseqüências dessas conjecturas e verificar se o resultado obtido é aceitável ou contraditório; **compromisso mínimo**, na qual a decisão de ativar uma fonte de conhecimento para resolver um subproblema é adiada o máximo possível até que se disponha de toda a informação necessária à sua ativação; ou a utilização de **várias linhas de raciocínio**, com a ativação múltipla de fontes de conhecimento durante o processo de resolução de problema. Alguns exempos do emprego dessas estratégias em sistemas baseados em conhecimentos são: sistema AM (Lenat, 1982), desenvolvido para formação de conceitos matemáticos - geração e teste; sistema EL (Stallman and Sussman, 1977), para análise de circuitos de resistores, diodos e transistores - raciocínio plausível; sistema NOAH (Sacerdoti, 1977), de planejamento de ações de um robô - compromisso mínimo; e sistema HEARSAY-II (Erman et a ii, 1980), para entendimento de fala - várias linhas de raciocínio.

TRANSFORMAÇÃO DO ESPAÇO DE BUSCA. A transformação mais comum do espaço de busca é a **redução** (ou **decomposição**) do problema em subproblemas. Neste caso procura-se reduzir um problema a subproblemas cada vez mais simples, até que se atinjam problemas elementares, cujas soluções são então combinadas para obter a solução do problema original. MYCIN é o exemplo clássico de sistema especialista que utiliza esse tipo de transformação do espaço de busca. Outro tipo de transformação é o **refinamento hierárquico**, no qual o espaço de busca é visto como tendo vários níveis de abstração e onde busca-se a solução para o problema partindo-se do nível mais alto (delineamento de solução) até chegar ao nível de detalhe desejado. Este tipo de transformação ocorre, por exemplo, no sistema ABSTRIPS (Sacerdoti, 1977). Uma outra abordagem para a transformação do espaço é restringir a busca pela aplicação de **meta-conhecimento**, ou seja, de conhecimento sobre quando e como o conhecimento de resolução de problema deve ser utilizado. Davis (1977) mostra as vantagens do emprego de **meta-regras** para a transformação do espaço de busca num sistema especialista.

1.3 - ARQUITETURA DE SISTEMAS ESPECIALISTAS

A organização de um sistema especialista deve refletir o tipo e a complexidade do problema a ser atacado, a disponibilidade do conhecimento necessário para resolvê-lo e a estratégia de controle que se pretende adotar para buscar a solução. Cada esquema de controle requer uma organização distinta da base de conhecimento e também um motor de inferência apropriado para aplicar o conhecimento disponível.

Segundo Stefik et alii (1982) e Hayes-Roth (1984), a dificuldade que um especialista encontra para resolver uma determinada tarefa, deve-se principalmente a 4 fatores:

- pode-se não dispor de dados ou de conhecimentos precisos;
- os dados podem variar dinamicamente;

- o espaço de busca pode ser grande;
- os procedimentos usados para descartar possibilidades de solução podem ser muito complexos e onerosos.

Portanto, a arquitetura mais simples para um sistema especialista ocorre quando se tem: um pequeno espaço de busca, dados estáticos e confiáveis, e conhecimento estático e confiável. Neste caso, como o espaço de busca é pequeno não é preciso preocupação com limitações de recursos computacionais, sendo possível utilizar **busca exaustiva** e é suficiente desenvolver uma **única linha de raciocínio.** Além disto, devido à confiabilidade e à estabilidade tanto dos dados como do conhecimento, o raciocínio pode ser **monotônico,** isto é, as conclusões uma vez inferidas, podem permanecer sempre na base de dados do sistema.

Uma dificuldade ocorre quando existem dados ou conhecimento não-confiáveis, o que é típico em tarefas de interpretação e diagnóstico. Muitas abordagens já foram propostas para tratar o problema de raciocínio com incerteza (Prade, 1982). Para problemas desse tipo deve-se portanto enriquecer a arquitetura do sistema com mecanismos para combinar evidências no processo de raciocínio.

Outra dificuldade aparece quando existem dados que variam com o tempo. Nestes casos o sistema deve basear seu raciocínio em expectativas e rever suas conclusões sempre que essas expectativas não forem confirmadas. Esse processo de raciocínio não-monotônico, denominado **revisão de crença** ou manutenção da verdade, pode ser feito por um mecanismo conhecido como **"backtracking" dirigido por dependências** ou "backtracking" não-cronológico (Doyle, 1979). A utilização de registros de dependência tem implicações importantes na arquitetura de um sistema especialista, até mesmo sobre sua capacidade de explicar ações e conclusões (deKleer et alii, 1979).

O grande desafio para a especificação da arquitetura de um sistema especialista surge quando o problema a ser atacado possui um grande espaço de solução, sendo inaplicável a busca exaustiva. Para

ganhar eficiência, é preciso descartar muitas alternativas do espaço de busca por meio de **regras de poda** que, em geral, são incorporadas num **gerador de soluções.** Com um gerador de soluções deste tipo, é possível utilizar geração e teste como estratégia de controle em grandes espaços de busca. A grande dificuldade é especificar as regras de poda, o que envolve muito conhecimento especializado. O interessante é dispor de regras que possam ser aplicadas o quanto antes no processo de geração, ou seja, regras que permitam descartar soluções ainda parcialmente especificadas ou que permitam descartar **classes** de soluções.

Uma outra forma de ganhar eficiência é dividir o problema em subproblemas de forma que, pela combinação de soluções parciais, se consiga, com menos esforço, uma solução aceitável. A divisão em subproblemas pode ser feita estaticamente, como ocorre por exemplo, no sistema R1, onde a tarefa de determinar uma boa configuração para um sistema VAX é vista como uma seqüência ordenada de 6 subtarefas, e no sistema DIPMETER ADVISOR (Smith and Baker, 1983), que divide a tarefa de interpretar medidas geológicas de poços de petróleo em 11 subtarefas sucessivas. Quando não for possível determinar um conjunto fixo de subproblemas para todos os problemas do domínio, pode-se tentar uma partição "ad hoc". Um enfoque é o do **refinamento hierárquico,** em que se buscam soluções para os subproblemas em níveis cada vez mais específicos. No caso dos subproblemas interagirem, este enfoque pode ser combinado com a estratégia de **compromisso mínimo** para guiar o processo de raciocínio - o que é feito, por exemplo, pelo sistema NOAH.

Quando a natureza do problema é muito complexa, a ponto de ser preciso combinar a contribuição de vários modelos ou a utilização de várias fontes de conhecimento, é preciso recorrer a arquiteturas que implementam **várias linhas de raciocínio** e são capazes de controlar a abrangência e a direção de busca. O sistema HEARSAY-II, por exemplo, emprega uma arquitetura deste tipo, na qual todos os elementos de solução gerados durante a resolução do problema são armazenados numa base de dados com estrutura hierárquica e acessível globalmente

("blackboard"). Pode-se sofisticar esta arquitetura com o emprego de 2 "blackboards", um para armazenar elementos de solução do problema e outro exclusivamente para controlar a ativação das fontes de conhecimento (Hayes-Roth, 1985a).

1.4 - O SISTEMA DIAGNÓSTICO

O sistema DIAGNÓSTICO (Sandri, 1985) é uma ferramenta para construção de sistemas especialistas desenvolvido à semelhança do sistema EMYCIN (van Melle, 1979). A Figura 1.2 ilustra a arquitetura do sistema.

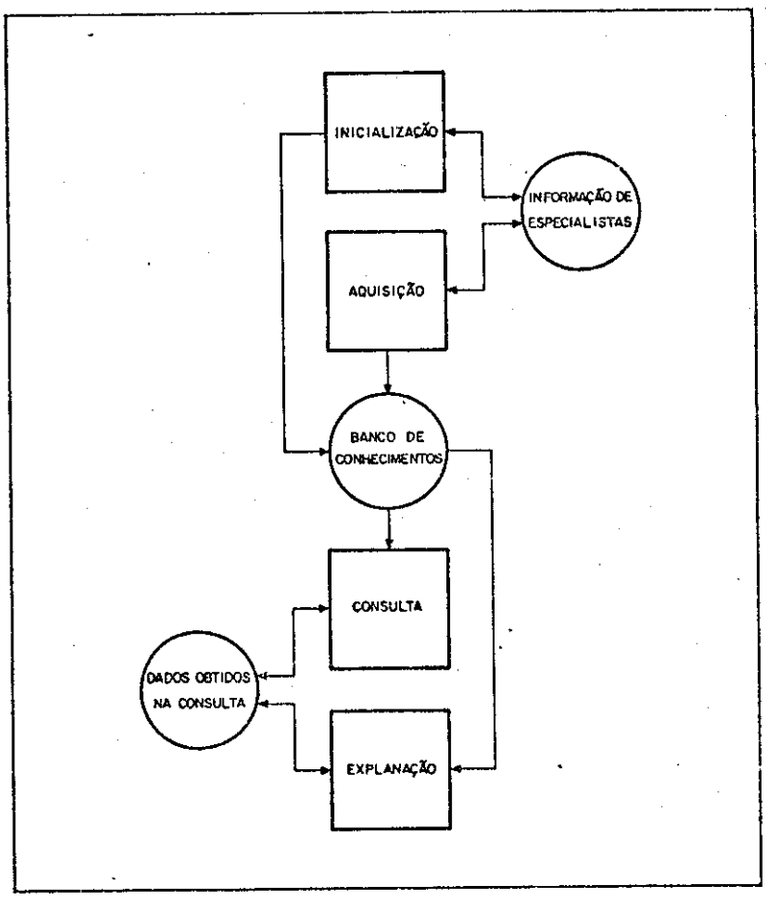

Fig. 1.2 - Arquitetura do sistema DIAGNÓSTICO

O sistema constitui-se de 4 módulos: INICIALIZAÇÃO, AQUISIÇÃO, CONSULTA e EXPLANAÇÃO, sendo os dois primeiros responsáveis pela aquisição do conhecimento de um dado domínio de aplicação e o último, pelo fornecimento de informações sobre o "status" de uma dada sessão de consulta. O interesse nesta seção é discutir, sucintamente, o módulo CONSULTA e o BANCO DE CONHECIMENTOS deste sistema.

O banco de conhecimentos do sistema DIAGNÓSTICO compõe-se de **regras de decisão** do tipo SE **premissa** ENTÃO (x) **conclusão**, onde x atesta a crença do especialista na validade da conclusão, caso a premissa seja absolutamente verdadeira. A premissa das regras é composta por cláusulas escritas na forma normal conjuntiva. A conclusão é sempre uma conjunção de cláusulas.

Uma cláusula é, basicamente, uma **quádrupla:**

(atributo, objeto, função, valor).

O **objeto** de uma cláusula é uma das entidades do domínio de conhecimento do sistema e o **atributo** representa uma das características deste objeto. A **função** relaciona o **valor** especificado ao atributo do objeto.

Durante uma aplicação, os objetos das cláusulas são instanciados e organizados numa estrutura conhecida como **árvore de contextos**, que é utilizada para dirigir a aplicação de regras de decisão.

O módulo CONSULTA é o **motor de inferência** do sistema, o qual, interagindo diretamente com o usuário, fornece as respostas a um dado problema. Este módulo utiliza como entrada o banco de conhecimentos (que contém as regras relativas ao domínio de aplicação) e os dados específicos do problema. O encadeamento de regras é feito de modo **retroativo** ("backward chaining") e a estratégia de controle é a utilização **exaustiva** das regras de decisão.

O sistema DIAGNÓSTICO emprega **fatores de certeza** (Shortliffe and Buchanan, 1975) para modelar o raciocínio inexato. Neste modelo a crença que se tem numa determinada informação é representada por um número que varia de -1 (absolutamente falso) até 1 (absolutamente verdadeiro).

1.5 - ABORDAGEM PROPOSTA: O SISTEMA MOTIN

O sistema **MOTIN** é uma proposta de **MOT**or de **IN**ferência para sistemas especialistas cujo paradigma arquitetural é o de sistemas de produção (Davis and King, 1977). Nesta proposta são atacados alguns dos problemas que normalmente se encontram em sistemas especialistas construídos segundo esse paradigma, tais como:

- o poder, muitas vezes, extremamente limitado de representação de conhecimento por meio de regras de decisão;
- a falta de um mecanismo para representar conhecimento estruturado ou de bom senso;
- a capacidade bastante limitada para estabelecer contextos de aplicação de regras, o que resulta numa fragilidade do mecanismo de inferência;
- a dificuldade em raciocinar de forma inexata em níveis variados de abstração;
- a falta de flexibilidade do procedimento de encadeamento de regras, o que limita seriamente a aplicabilidade dos sistemas quando se consideram domínios diferenciados.

As soluções propostas para esses problemas, motivadas pela idéia de que o sistema MOTIN deva ser o núcleo de uma ferramenta para a construção de sistemas especialistas e que portanto o requisito de generalidade é indispensável, envolvem:

- o estabelecimento de um esquema de representação por regras de decisão capaz de admitir o emprego de variáveis quantificadas para expressar conhecimento em cláusulas generalizadas;
- o emprego de um esquema de rede semântica para representar o conhecimento bem estabelecido ou de bom senso sobre o domínio de aplicação e também para estruturar primitivos conceituais, especialmente objetos e valores;
- a utilização de uma estrutura de grafo, resultante de instanciações de objetos e de relações entre elas e em conformidade com a rede semântica que estrutura o conhecimento do domínio, para estabelecer contextos generalizados de aplicação de regras, ou seja, contextos de aplicação para regras que envolvem objetos relacionados de maneira não trivial;
- uma aproximação da teoria de Dempster-Shafer (Shafer and Logan, 1985; Gordon and Shortliffe, 1985; Zadeh, 1986) para o tratamento da incerteza no raciocínio evidencial, na qual é possível estabelecer crença em classes de valores de interesse semântico e desta maneira permitir o raciocínio inexato num nível de abstração que o especialista ou a aplicação exigir;
- a busca de eficiência e de flexibilidade nos mecanismos de encadeamento de regras de decisão, seja pela utilização de meta-regras para selecionar as regras mais promissoras dependendo de características do problema a resolver, seja pelo uso de estratégia que evite a rigidez dos encadeamentos puramente progressivo ou retroativo.

A motivação maior para o desenvolvimento do sistema MOTIN foram as dificuldades encontradas com a utilização do sistema DIAGNÓSTICO na construção de um sistema especialista para o domínio de Meteorologia, além de uma série de pontos passíveis de melhoria que foram levantados já no desenvolvimento da própria ferramenta.

Uma das grandes dificuldades foi o emprego de um mecanismo de inferência por encadeamento retroativo de regras numa aplicação de previsão (de precipitação pluviométrica em São José dos Campos), para a qual um procedimento progressivo parece ser mais adequado.

Numa aplicação de previsão tem-se um certo volume de dados e deseja-se saber suas conseqüências. Isto encaixa-se perfeitamente numa estratégia progressiva de raciocínio. Ao utilizar a estratégia retroativa numa aplicação de previsão é preciso partir de objetivos ("vai chover" e "não vai chover", no caso específico do sistema desenvolvido), persegui-los independentemente e ao final compor uma resposta com base em valores de certeza.

Este tipo de solução, restrita e forçada, será inevitável sempre que as tarefas a serem resolvidas não se adequarem perfeitamente à rigidez do procedimento de inferência implementado. Pior do que isto, como o planejamento da solução dependerá do especialista que utiliza o sistema e como esse tipo de raciocínio não lhe será natural, é provável que uma solução obtida não seja satisfatória. Este cenário mostra a necessidade de se ter mecanismos de inferência flexíveis em ferramentas para a construção de sistemas especialistas.

As limitações do esquema de representação de conhecimento dão origem a outras dificuldades, como por exemplo, quando uma dada parcela do conhecimento necessita de outros primitivos conceituais, além de

atributo (objeto) função valor

para ser representada. Considere-se por exemplo:

"a velocidade da zona frontal às 13 horas é de 10 km/h"

que poderia ser representada por:

VELOCIDADE (ZONA-FRONTAL-ÀS-13-HORAS) = 10

ou por:

VELOCIDADE-ÀS-13-HORAS (ZONA-FRONTAL) = 10

o que cria dificuldades intransponíveis quando é preciso representar este conhecimento em outros instantes de tempo, como por exemplo:

"a velocidade da zona frontal às 12 horas é de 15 km/h",
pois num caso seria preciso explicitar que os eventos se referem à mesma zona frontal e no outro ter-se-ia uma infinidade de atributos.

Estas dificuldades são superadas ao se dispor de um esquema "aberto" de representação, ou seja, um esquema que não restrinja a quatro os primitivos conceituais, podendo incorporar novos primitivos de acordo com a necessidade do domínio de conhecimento e da aplicação desejada. No caso acima, poder-se-ia introduzir tempo como novo primitivo, ou seja:

VELOCIDADE (ZONA-FRONTAL , 12) = 15
VELOCIDADE (ZONA-FRONTAL , 13) = 10

O emprego de um esquema aberto, no entanto, não supera todas as dificuldades. Considere-se por exemplo que uma vez introduzido tempo como novo primitivo conceitual, deseja-se raciocinar temporalmente, ou seja, deseja-se representar fatos tais como:

"a velocidade da zona frontal está diminuindo"
o que pode ser traduzido em:

"para quaisquer instantes de tempo T1 e T2,
tais que T1 > T2,
VELOCIDADE(ZONA-FRONTAL,T1) < VELOCIDADE(ZONA-FRONTAL,T2)"

Isto mostra a necessidade de, além de cláusulas nas quais representam-se valores de atributos, poder-se também representar relações entre valores.

Uma outra dificuldade, também relacionada ao problema de representação, aparece em domínios nos quais o conhecimento é mais estruturado, sendo necessário, por exemplo, representar taxionomias ou mesmo relações de natureza não hierárquica entre conceitos. Representar este tipo de conhecimento através de regras de decisão tem o sério inconveniente de se perder a estrutura global do domínio, uma

vez que as regras devem ser encaradas como módulos independentes de conhecimento.

Além disso, é conveniente separar o conhecimento empírico do conhecimento bem estabelecido, de natureza profunda ou de bom senso. Isto porque o conhecimento empírico é muitas vezes controverso e subjetivo pois expressa a opinião de um ou mais especialistas, o que é interessante representar por regras de decisão, dadas as características de separabilidade e alterabilidade deste esquema. Por outro lado o conhecimento bem estabelecido dependerá apenas do domínio de aplicação em questão.

Uma representação mais conveniente para este conhecimento melhor estruturado é obtida através de um esquema de rede semântica. Desta forma, ao invés de escrever:

```
SE:    LOCAL (SISTEMA-SINÓTICO) = PARANÁ OU
       LOCAL (SISTEMA-SINÓTICO) = SANTA-CATARINA OU
       LOCAL (SISTEMA-SINÓTICO) = RIO-GRANDE-SUL
ENTÃO: LOCAL (SISTEMA-SINÓTICO) = SUL-DO-BRASIL
```

como foi preciso para o sistema especialista em previsão pluviométrica, poder-se-ia representar o conceito mostrado na Figura 1.3 o qual então, poderia ser encarado como LOCAL para outros objetos do domínio.

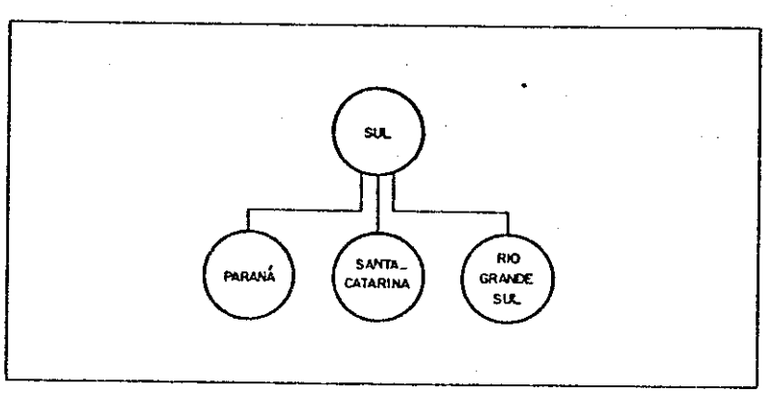

Fig. 1.3 - Representação estruturada de conceito

Além das vantagens inerentes a esse tipo de esquema como por exemplo a possibilidade de um conceito herdar propriedades de outro ou a utilização de regras sobre um conceito (por exemplo, SANTA-CATARINA) para deduzir propriedades de outro (por exemplo, SUL), a representação explícita da estrutura do domínio, para o caso dos valores, irá permitir também que se implemente mecanismos para raciocínio em níveis diferenciados de abstração. Por exemplo, para o caso da Figura 1.3 será possível associar crença a SUL sem que haja a necessidade de saber se o local exato é PARANÁ, SANTA-CATARINA ou RIO-GRANDE-SUL, ou mesmo associar crença ao conceito "o local NÃO é Paraná" o que corresponde a associar crença à classe { SANTA-CATARINA , RIO-GRANDE-SUL}. Em outras palavras, será possível raciocinar em termos de **grupos de valores**, o que não ocorre quando se adota um modelo de fatores de certeza.

Para o caso dos objetos, a representação explícita da estrutura do domínio irá permitir superar uma das grandes dificuldades dos sistemas especialistas desenvolvidos segundo o paradigma de sistemas de produção, qual seja o do **estabelecimento de contextos para aplicação de regras**. Em sistemas que trabalham com domínios hierarquizados nos quais é possível existir várias instâncias de um mesmo objeto, já desenvolvidos, a aplicação de regras é sempre orientada por uma **árvore de contexto.** Esta árvore surge em conseqüência de se instanciar objetos que são considerados como obedecendo - implicitamente - a uma hierarquia linear estrita. Este tipo de estrutura tem origem no sistema MYCIN para o qual é muito comum ter-se uma árvore de contextos como a da Figura 1.4.

Para uma árvore desse tipo, os contextos são estabelecidos pelas instâncias que se apresentam numa mesma linha de ancestralidade em relação à instância-raiz (deve-se notar que no sistema MYCIN os objetos mantêm estritamente a hierarquia PACIENTE - INFECÇÃO - CULTURA - ORGANISMO). Para o caso da Figura 1.4, por exemplo,

PACIENTE-1, INFECÇÃO-1, CULTURA-1, ORGANISMO-1

é um contexto válido, mas

PACIENTE-1, INFECÇÃO-2, CULTURA-2, ORGANISMO-4

não é.

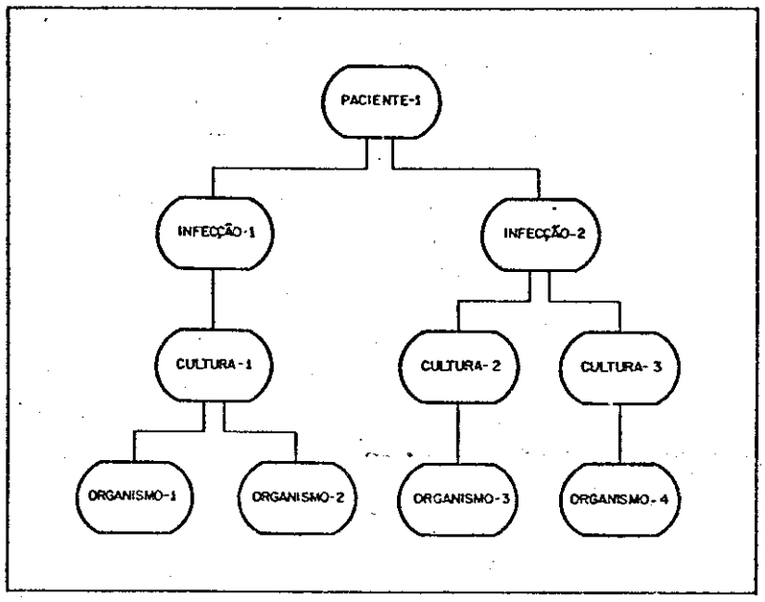

Fig. 1.4 - Árvore de contextos

Este tipo de estrutura - muito conveniente para o domínio de infecções bacterianas para o qual o sistema MYCIN foi desenvolvido - vem sendo utilizado na construção de muitos sistemas especialistas devido à sua simplicidade. Entretanto, seu pressuposto básico, isto é, que os objetos do domínio mantêm entre si uma hierarquia estritamente linear, limita seriamente sua aplicabilidade. Considere-se, por exemplo, os objetos SJC, SISTEMA-SINÓTICO, JATO e BLOQUEIO que, dentre outros, ocorrem no sistema especialista em precipitação pluviométrica construído por meio do sistema DIAGNÓSTICO. Estes objetos mantêm entre si a estrutura mostrada na Figura 1.5.

Este relacionamento, apesar de muito simples, invalida o requisito de ancestralidade necessário para se estabelecer contextos através de uma árvore de instâncias desses objetos. Basta, neste caso,

que uma regra referencie pelo menos duas folhas da árvore para se ter objetos (e conseqüentemente, instâncias) que não participam de uma mesma linha de ancestralidade em relação a SJC. No sistema especialista desenvolvido foi preciso, por exemplo, escrever regras que referenciam SISTEMA-SINÓTICO e JATO e regras que referenciam SISTEMA-SINÓTICO e BLOQUEIO o que, devido ao mecanismo de árvore de contexto, forçou ter que se considerar JATO e BLOQUEIO não como objetos mas como **atributos** de SJC.

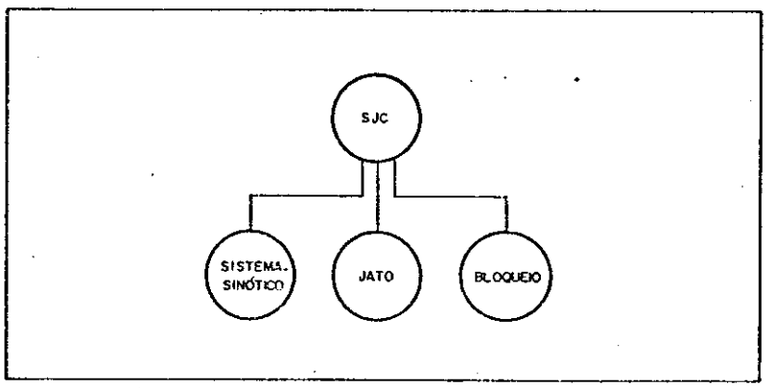

Fig. 1.5 - Hierarquia de objetos

A transformação de objetos do domínio em atributos para resolver problemas no estabelecimento de contextos, apresenta o sério inconveniente de não ser mais possível caracterizar aqueles objetos, ou seja, atribuir-lhes qualidades, uma vez que não existem atributos de atributos. Dessa maneira, para o sistema especialista desenvolvido, com a transformação de JATO e BLOQUEIO em atributos fica impossível falar sobre a VELOCIDADE do JATO ou o LOCAL em que ocorre o BLOQUEIO.

Outro sério problema em relação aos objetos ocorre quando é preciso considerar a semântica de seus relacionamentos. Por exemplo, os objetos ZONA-FRONTAL e SISTEMA-SINÓTICO obedecem a uma relação de **particularização**, ou seja:

ZONA-FRONTAL **particularização-de** SISTEMA-SINÓTICO

Havendo a possibilidade de representar este tipo de relacionamento, pode-se por exemplo, utilizar regras que concluem sobre SISTEMA-SINÓTICO quando for preciso concluir sobre ZONA-FRONTAL e não existirem, ou não for possível aplicar, regras para isso. De uma maneira geral, a possibilidade de representar a semântica de relações aumenta o poder de raciocínio dos sistemas.

Para fazer frente a este problema, dada a ausência de mecanismos para representação de relações entre objetos no sistema DIAGNÓSTICO, foi preciso novamente lançar mão da artificialidade, considerando a estrutura da Figura 1.6, onde SS-ZF representa todos os sistemas sinóticos exceto os relativos a zonas frontais, que por necessidade da aplicação são consideradas separadamente.

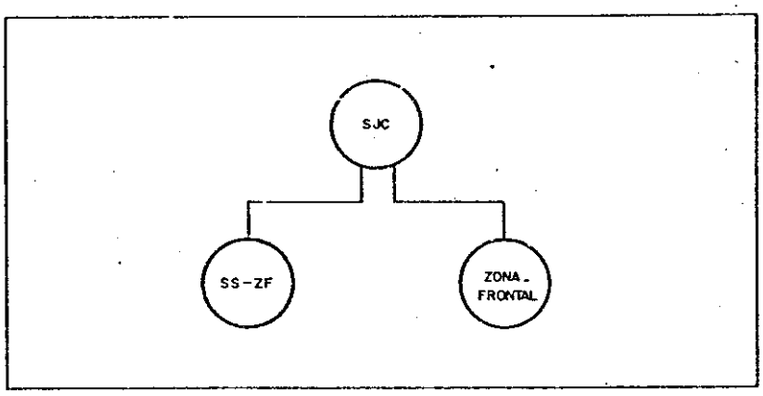

Fig. 1.6 - Relacionamento de objetos

Todos esses problemas de contexto são superados ao se adotar um esquema de rede semântica para representar a estrutura dos objetos do domínio de conhecimento. No que diz respeito à aplicabilidade de regras, substitui-se o mecanismo de estabelecimento de contextos baseado em árvore por um mecanismo baseado em **grafo**, tornando-o completamente geral, qualquer que seja a estrutura construída pelos objetos. Evidentemente existe um preço a se pagar por tal generalidade: a complexidade do mecanismo é muito maior.

Enfim, para resumir e apontar os ganhos em relação ao sistema DIAGNÓSTICO e principalmente, para ressaltar os avanços

teóricos à construção de sistemas especialistas em geral, tem-se com o sistema MOTIN:

- o conhecimento empírico e incerto de um domínio de aplicação **separado** do conhecimento bem estabelecido e de bom senso. Dispõe-se no primeiro caso de uma representação por regras de decisão um pouco mais geral do que a utilizada pelos sistemas atuais e na qual a incorporação de primitivos conceituais é permitida e facilitada, e no segundo caso de um esquema de rede semântica para estruturar principalmente os objetos e valores do domínio de conhecimento;
- um mecanismo geral de estabelecimento de contextos de aplicação de regras e que leva em conta a estrutura definida pelos objetos do domínio;
- um mecanismo de combinação e propagação de crenças que, fazendo uso da estrutura dos valores do domínio, permite o raciocínio inexato em níveis variados de abstração;
- incorporado num motor de inferência, mecanismos flexíveis para estabelecer a linha de raciocínio de um sistema especialista, afastando-se das estratégias puramente progressivas ou retroativas e ampliando as suas fronteiras de aplicação.

1.6 - ORGANIZAÇÃO DO DOCUMENTO

Neste capítulo discutiu-se, com base no sistema DIAGNÓSTICO, as limitações e os problemas que enfrentam os sistemas especialistas atualmente desenvolvidos à maneira de sistemas de produção e mostrou-se, em linhas gerais, o que se dispõe com o sistema MOTIN.

No Capítulo 2 discutem-se as questões relativas à **representação de conhecimento**, considerando os esquemas de regras de decisão e de rede semântica disponíveis no sistema MOTIN. Mostra-se

também neste capítulo, trechos de sessão de aquisição de conhecimentos na qual estes esquemas de representação são atualizados.

O Capítulo 3 aborda o problema de contextos de aplicação de regras de decisão e apresenta os algoritmos utilizados pelo **mecanismo de estabelecimento de contextos** do sistema MOTIN.

No quarto capítulo apresenta-se brevemente a teoria de Dempster-Shafer para **combinação de evidências**, propõe-se uma aproximação a esta teoria e discute-se por quê e como esta aproximação está implementada no sistema MOTIN.

No Capítulo 5 discutem-se os **mecanismos de inferência** e de **controle de raciocínio** do sistema MOTIN e suas relações com os esquemas de armazenagem de conhecimento. Apresenta-se também neste capítulo, trechos de sessão de consulta ao sistema nos quais ilustram-se capacidades dos procedimentos de raciocínio.

Por fim, no Capítulo 6 conclui-se apresentando um balanço geral deste trabalho comparando a abordagem proposta com trabalhos relatados na literatura. Apresentam-se também algumas direções para futuras extensões ao trabalho realizado.

No Apêndice A demonstram-se alguns resultados acerca do modelo de verdade adotado no sistema MOTIN. O Apêndice B contém o banco de regras do sistema de precipitação pluviométrica citado neste capítulo e utilizado nos exemplos deste trabalho. No Apêndice C tem-se a organização e a descrição dos módulos constituintes do programa que implementa o sistema MOTIN.

CAPÍTULO 2

REPRESENTAÇÃO DE CONHECIMENTO NO SISTEMA MOTIN

2.1 - INTRODUÇÃO

A representação de conhecimento no sistema MOTIN se dá com a utilização de dois esquemas: **regra de decisão** e **rede semântica.** O primeiro é utilizado para codificar o conhecimento empírico e incerto, e o segundo para estruturar o conhecimento melhor estabelecido, de um determinado domínio de aplicação.

Uma característica interessante dos esquemas propostos é a sua **flexibilidade.** Para as regras de decisão é possível representar, além de cláusulas que estabelecem atributos de determinados objetos do domínio de conhecimento, relações entre objetos ou valores e também chamadas a procedimentos. Além disso, permite-se a introdução de novos primitivos conceituais em cima dos quais as regras de decisão são escritas. Para o esquema de rede permite-se a definição da semântica de novas relações. Essas características de flexibilidade revestem-se de grande importância para ferramentas de construção de sistemas especialistas pois permitem que o esquema de representação seja adaptado às características específicas dos domínios de aplicação.

Neste capítulo descrevem-se as características dos esquemas de representação de conhecimento utilizados no sistema MOTIN. A utilização desses esquemas nos processos de inferência é assunto do Capítulo 5.

2.2 - ARMAZENAGEM DE CONHECIMENTO EM REGRAS DE DECISÃO

As regras de decisão utilizadas no sistema MOTIN obedecem a sintaxe a seguir. Na apresentação desta gramática são utilizados símbolos meta-lingüísticos com os seguintes significados:

sublinhado	variável meta-lingüística
[...]	parte opcional
... ! ...	alternativa (uma e apenas uma deve ocorrer)
{ ... }	repetição (zero ou mais vezes)

```
regra ::=
   SE: premissa ENTÃO: conclusão
premissa ::=
   [ quantificação ] condição { & ! V condição }
quantificação ::=
   TODOS ! ALGUM ( variáveis ) [ [,] ( expressões ) ]
variáveis ::=
   variável  {, variável }
expressões ::=
   expressão relacional { , expressão relacional }
expressão relacional ::=
   expressão  relação  expressão
expressão ::=
   expressão + termo !
   expressão - termo !
   termo
termo ::=
   termo * fator !
   termo / fator !
   fator
fator ::=
   primário ^ fator !
   primário
primário ::=
```

```
    - primário !
    elemento
elemento ::=
    ( expressão ) !
    valor
condição ::=
    cláusula predicativa  !
    cláusula procedimental  !
    cláusula relacional
cláusula predicativa ::=
    predicado  função  valor !
    predicado  função  predicado
predicado ::=
    atributo ( objeto [ , restrições ] )
restrições ::=
    restrição { , restrição }
restrição ::=
    primitivo  função  valor
valor ::=
    constante  !
    variável
cláusula procedimental ::=
    nome de procedimento ( argumentos )
argumentos ::=
    argumento { , argumento }
argumento ::=
    objeto !
    atributo !
    valor
cláusula relacional ::=
    relação [ EXISTE ! NÃO-EXISTE ]
relação ::=
    designador de relação ( objeto , objeto ) !
    designador de relação ( valor , valor )
conclusão ::=
    ação { & ação }
```

ação ::=
   cláusula predicativa de ação !
   cláusula procedimental !
   cláusula relacional
cláusula predicativa de ação ::=
   cláusula predicativa [[,] fator de crença]

Alguns pontos precisam ser observados com relação a esta gramática:

- variável, primitivo, nome de procedimento e designador de relação são identificadores, onde variável representa um termo que necessita instanciação e nome de procedimento representa o nome de uma função, no sentido LISP do termo. Para facilitar a análise utilizam-se alguns prefixos preestabelecidos:
  - * ... para variável;
  - $ ... para nome de procedimento;
  - ╫ ... para designador de relação;
- expressão é uma expressão relacional que envolve variáveis e que corresponde a restrições sobre suas instanciações possíveis.
- atributo, objeto, função e valor são primitivos conceituais. O objeto é uma entidade do domínio de conhecimento, o atributo representa uma característica do objeto e valor corresponde a um valor simbólico ou numérico. A função relaciona o valor ao atributo do objeto. Novos primitivos poderão ser introduzidos por meio de primitivo.
- fator de crença é um número do intervalo [0,1] e que representa a crença depositada numa conclusão da regra dado que sua premissa é absolutamente verdadeira.

Os exemplos a seguir ilustram algumas regras para o domínio de Meteorologia, escritas de acordo com esta gramática.

EXEMPLO 1

SE: LOCAL (ZONA-FRONTAL) = PARANÁ &

VELOCIDADE (ZONA-FRONTAL) ≥ 10 &
CONDIÇÃO-ALTO-NÍVEL (SJC) = FAVORÁVEL
ENTÃO: C-G-E (SJC , TEMPO = 24) = FAVORÁVEL , 0.8

Esta regra estabelece que "se existe uma zona frontal no Paraná, com velocidade maior ou igual a 10 km/h e se as condições de altos níveis em São José dos Campos são favoráveis à precipitação pluviométrica, então acredita-se com fator de crença 0.8 que as condições de grande escala (C-G-E) em São José dos Campos, em 24 horas, também são favoráveis". Neste caso TEMPO é um novo primitivo conceitual, introduzido para indicar para que instante, ou intervalo de tempo, a evidência se refere.

EXEMPLO 2

SE: TODOS (*T1 , *T2) , (*T1 > *T2)
DIST (VÓRTICE , TEMPO = *T1) <
DIST (VÓRTICE , TEMPO = *T2)
ENTÃO: CONDIÇÃO-ALTO-NÍVEL (SJC) = FAVORÁVEL , 0.7

Neste caso o atributo DIST representa a distância em relação a São José dos Campos. Esta regra portanto, estabelece: "se existe um vórtice ciclônico se aproximando de São José dos Campos então existe sugestiva evidência (0.7) de que as condições de altos níveis são favoráveis à precipitação pluviométrica neste local".

EXEMPLO 3

SE: CONDIÇÃO-ALTO-NÍVEL (SJC) = DESFAVORÁVEL &
#INFLUI-EM (BLOQUEIO , SJC)
ENTÃO: C-G-E (SJC) = DESFAVORÁVEL , 0.9

Nesta regra tem-se a presença de uma relação (#INFLUI-EM) entre objetos para indicar quando um bloqueio no escoamento tem conseqüências nas condições de grande escala em São José dos Campos.

EXEMPLO 4

```
SE:    TAMANHO ( BLOQUEIO ) = *X                    &
       LOCAL ( BLOQUEIO ) = *Y                      &
       $FUTURO ( *X , *Y , *Z )                     &
       CONDIÇÃO-LOCAL ( SJC ) = DESFAVORÁVEL
ENTÃO: PRECIPITAÇÃO ( SJC , TEMPO = *Z ) = BAIXA
```

Este caso mostra a passagem de valores entre as cláusulas por meio de variáveis e também o emprego do procedimento $FUTURO que com base no tamanho e no local de um bloqueio no escoamento, calcula quando deverá chover em São José dos Campos.

Numa regra, os **objetos** (como por exemplo, ZONA-FRONTAL ou SJC, no Exemplo 1, VÓRTICE, no Exemplo 2 e BLOQUEIO, no Exemplo 3) são encarados como **variáveis**. Para que uma regra possa ser aplicada é necessário que seus objetos estejam todos instanciados. Como, em geral, pode haver várias instâncias de um mesmo objeto, uma determinada regra poderá ser aplicada várias vezes. A determinação das instâncias válidas para os objetos de uma regra constitui o problema de **estabelecimento de contextos de aplicação**, discutido no Capítulo 3.

As relações binárias entre objetos como ocorrem nas regras de decisão são consideradas como **relações circunstanciais** uma vez que se estes objetos puderem sempre satisfazer essa relação então ela será melhor representada na rede semântica que estrutura o conhecimento do domínio de aplicação.

2.3 - ATRIBUTOS, PREDICADOS E EVIDÊNCIAS

Os atributos no sistema MOTIN, são classificados segundo os valores que podem assumir. Dessa forma existem atributos SIMBÓLICOS, cujos valores são nomes ou "strings" e atributos NUMÉRICOS, cujos valores são números ou intervalos. Também existe associado a um

atributo o DOMÍNIO de seus valores. O domínio de valores de um atributo simbólico é dado por um conjunto e o de atributos numéricos por um intervalo.

Por exemplo:

atributo	domínio
COR	{VERMELHO, VERDE, AZUL}
VELOCIDADE	[0, 100]

Considera-se que os valores de um atributo (como especificado pelo seu domínio) são mutuamente exclusivos, o que corresponde, para os atributos de domínio finito - atributos simbólicos em geral -, segundo Shafer (1976), a seu "frame of discernment".

O esquema de armazenagem de conhecimento permite a representação do par (atributo , objeto), ou seja, de um atributo de um determinado objeto (esse par será denominado **predicado**). Essa característica é importante porque um mesmo atributo, dependendo do objeto ao qual ele se refere pode ter significações diferentes. O domínio de um atributo, por exemplo, geralmente é diferente quando se trata de objetos diferentes. Por exemplo, considere o atributo IDADE dos objetos CRIANÇA e ADULTO:

DOMÍNIO(IDADE(CRIANÇA)) = [0 , 12]
DOMÍNIO(IDADE(ADULTO)) = [20 , 120]

Denomina-se **predicado instanciado** o par (atributo , instância de objeto). Por exemplo, se VELOCIDADE é um atributo do objeto ZONA-FRONTAL e ZONA-FRONTAL-1 é uma instância de ZONA-FRONTAL, então (VELOCIDADE , ZONA-FRONTAL-1) é um predicado instanciado. Aos predicados instanciados são atribuídas **propriedades**. Em princípio (isto é, por "default") a única propriedade de um predicado instanciado é o seu VALOR. Outras propriedades poderão, no entanto, ser associadas aos predicados instanciados e irão funcionar como novos

primitivos conceituais do domínio de conhecimento. Denomina-se **evidência** um predicado instanciado juntamente com suas propriedades associadas. As evidências têm também representação explícita no esquema de armazenagem de conhecimento.

Para resumir, seja por exemplo:

atributo ..... VELOCIDADE;
objeto ..... ZONA-FRONTAL;
instância de objeto ZONA-FRONTAL-1;
predicado ..... VELOCIDADE de uma ZONA-FRONTAL;
predicado instanciado ... VELOCIDADE da ZONA-FRONTAL-1;
propriedades ..... VALOR, HORÁRIO-DE-OCORRÊNCIA;
evidência ..... a VELOCIDADE da ZONA-FRONTAL-1, às 13 horas, é 10 km/h.

2.4 – ARMAZENAGEM DE CONHECIMENTO EM REDE SEMÂNTICA

O esquema de rede semântica é usado no sistema MOTIN, principalmente, para representar a estrutura do domínio de conhecimento no que diz respeito aos objetos e valores. Com relação aos objetos o esquema representa as **relações** que são ou que devem ser satisfeitas nesse domínio e para os valores representa, além dessas relações, as **classes de interesse semântico** para uma determinada aplicação.

Seja por exemplo, a rede da Figura 2.1 na qual tem-se uma parte (**nem todas as relações estão representadas**) do que poderia ser a rede semântica para o domínio do sistema especialista em precipitação pluviométrica em São José dos Campos.

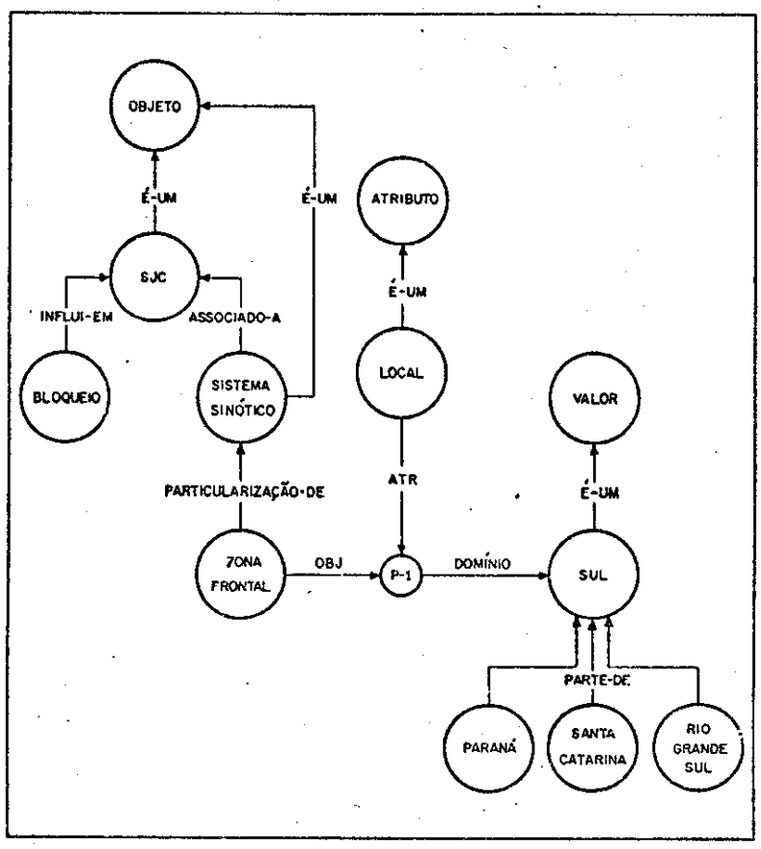

Fig. 2.1 - Rede semântica para o domínio de Meteorologia

Neste caso tem-se representado que:

(a) SJC (condições climáticas para previsão de precipitação de chuva na região de São José dos Campos), SISTEMA-SINÓTICO e ZONA-FRONTAL são **objetos** do domínio. BLOQUEIO também é, mas isto não está representado na Figura 2.1.

(b) estes objetos se relacionam da seguinte maneira:

INFLUI-EM(BLOQUEIO , SJC)

ASSOCIADO-A(SISTEMA-SINÓTICO , SJC)

PARTICULARIZAÇÃO-DE(ZONA-FRONTAL , SISTEMA-SINÓTICO)

(c) P-1 é um predicado e representa o "local de uma zona frontal".

(d) PARANÁ, SANTA-CATARINA, RIO-GRANDE-SUL e SUL são **classes de valores** de interesse semântico. SUL é a classe que corresponde a { PARANÁ , SANTA-CATARINA , RIO-GRANDE-SUL }

(e) o domínio de P-1 é SUL, isto é, o LOCAL de uma ZONA-FRONTAL pode ser SUL, PARANÁ, SANTA-CATARINA ou RIO-GRANDE-SUL.

Neste esquema de rede semântica existem relações com significados preestabelecidos como PARTICULARIZAÇÃO-DE, PARTE-DE, É-UM, ATR, OBJ, DOMÍNIO, e relações que são definidas pelo especialista, como INFLUI-EM e ASSOCIADO-A. Estas novas relações podem ser definidas como meramente **estruturais**, ou seja, que associam trivialmente uma entidade a outra, ou então podem ter condições para que sejam estabelecidas. Neste último caso a relação é denominada **condicional** e as condições que devem ser satisfeitas para que ela seja estabelecida de fato denominam-se **definição da relação condicional.**

Por exemplo, na Figura 2.1 pode-se imaginar ASSOCIADO-A como uma relação estrutural, indicando que os sistemas sinóticos de interesse são aqueles associados a São José dos Campos e INFLUI-EM como uma relação condicional cuja definição é dada por:

DISTÂNCIA(LOCAL(BLOQUEIO) , LOCAL(SJC)) < 100

na qual DISTÂNCIA corresponde a um procedimento (função LISP) estabelecido pelo especialista.

Em geral, define-se uma nova relação em função de atributos dos objetos envolvidos e de relações já conhecidas (preestabelecidas ou definidas anteriormente pelo especialista).

Para definir uma nova relação usa-se portanto, uma regra de decisão que estabelece na conclusão o relacionamento desejado entre os objetos e cuja premissa constitui sua definição. Para o exemplo acima, pode-se ter:

```
SE:     LOCAL ( BLOQUEIO ) = *X          &
        LOCAL ( SJC ) = *Y               &
        $DISTÂNCIA ( *X , *Y , *Z )      &
        $MENOR ( *Z , 100 )
ENTÃO:  #INFLUI-EM ( BLOQUEIO , SJC )
```

Quando é preciso definir uma relação entre dois objetos quaisquer, os nomes dos objetos (como BLOQUEIO e SJC, no caso acima) devem ser substituídos por variáveis.

Além de representar o conhecimento **permanente** do domínio de aplicação, o esquema de rede semântica é utilizado para representar também duas estruturas que ocorrem durante uma aplicação específica: o BANCO DE EVIDÊNCIAS e o GRAFO DE INSTÂNCIAS.

O **grafo de instâncias** contém as instâncias dos objetos do domínio para uma determinada aplicação e seus relacionamentos. Com base nesta estrutura são estabelecidos os contextos de aplicação para as regras de decisão, conforme se discute no Capítulo 3. O **banco de evidências** armazena os predicados instanciados e suas propriedades à medida que elas aparecem no curso de uma sessão de consulta. Considere, por exemplo, as evidências: "o LOCAL da ZONA-FRONTAL-1 às 12 horas é SANTA-CATARINA e às 13 horas é o PARANÁ" cuja representação é mostrada na Figura 2.2.

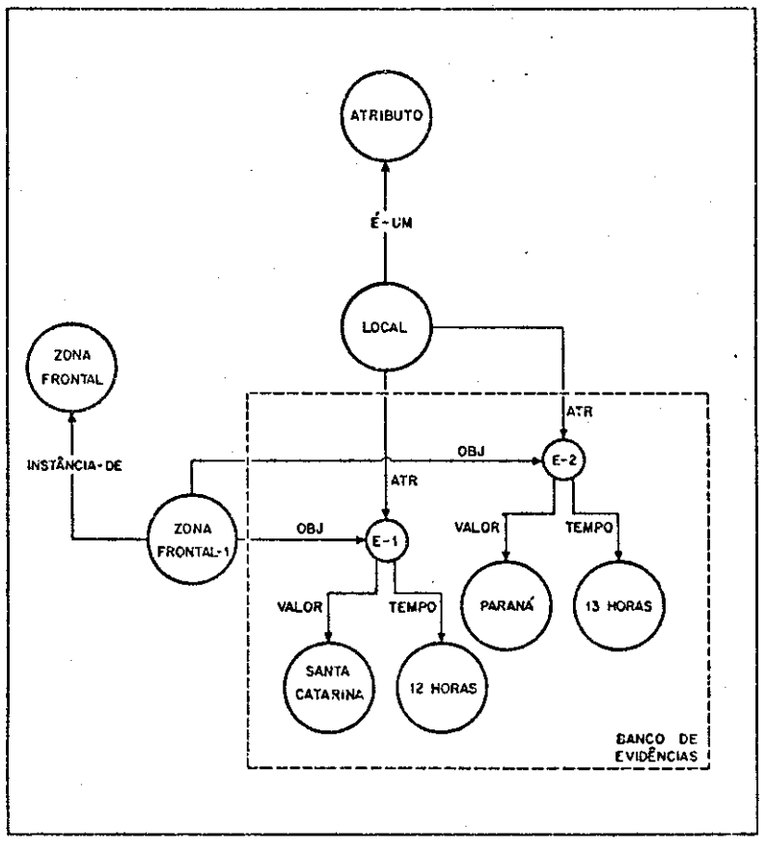

Fig. 2.2 - Representação de evidências

Tanto o grafo de instâncias como o banco de evidências são particulares a um determinado problema do domínio e portanto são estruturas **temporárias**, podendo ser armazenadas - a critério do usuário - ou descartadas, no final de uma sessão.

2.5 - AQUISIÇÃO DE CONHECIMENTO NO SISTEMA MOTIN

Apresentam-se a seguir alguns trechos de sessão de aquisição de conhecimentos pelo sistema MOTIN. O símbolo " > " corresponde ao "**prompt**" do sistema, ou seja, após este símbolo tem-se as entradas do usuário. As linhas sublinhadas não pertencem à sessão; são comentários introduzidos para explicar determinados estágios do processo de aquisição.

```
MODULO DE AQUISICAO DE CONHECIMENTO

    Inicializacao
    Nova relacao
    Nova regra ou meta-regra
    Sair do modulo de aquisicao
```

Com a opção 1 tem-se a inicialização de variáveis e a especificação do nome de um novo banco de conhecimento.

```
INICIALIZACAO DO MODULO DE AQUISICAO

Banco de conhecimento:
> PREVISAO

MODULO DE AQUISICAO DE CONHECIMENTO

    Inicializacao
```

Nova relacao

Nova regra ou meta-regra

Sair do modulo de aquisicao

AQUISICAO DE NOVA REGRA

```
REGRA $R1
SE:
     Q >
     1 > local(zona-frontal) = pr
     2 > velocidade(zona-frontal) = 10
     3 > atividade(zona-frontal) = "DESENVOLVENDO-SE"
     4 > condicao-alto-nivel(sjc) = favoravel
     5 >
ENTAO:
     1 > c-g-e(sjc,tempo = 24) = favoravel .8
     2 >
```

O sistema exibe seu entendimento da regra do usuário:

REGRA $R1

```
SE:
  1) LOCAL (ZONA-FRONTAL) = PR    &
  2) VELOCIDADE (ZONA-FRONTAL) = 10    &
  3) ATIVIDADE (ZONA-FRONTAL) = "DESENVOLVENDO-SE"    &
  4) CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL
ENTAO:
  1) C-G-E (SJC,TEMPO = 24) = FAVORAVEL  (FC = 0.80)

Ok?
 > s
```

Se a regra adquirida envolve novos conceitos, tais como: objetos, atributos, primitivos, ..., inicia-se um diálogo para a aquisição das informações necessárias sobre estes conceitos.

INFORMACOES ADICIONAIS SOBRE PRIMITIVOS

Dominio de TEMPO:
> [-24,24]

INFORMACOES ADICIONAIS SOBRE PREDICADOS

Para descobrir o valor de LOCAL (ZONA-FRONTAL)
deve-se primeiro (P)erguntar ou (D)eduzir?
> p

Dominio de valores de LOCAL (ZONA-FRONTAL):
> {pr, sc, rs} = sul

Na resposta acima o usuário indica que SUL corresponde à classe PR SC, RS . Esta classe de valores é armazenada na rede semântica, com atualização das relações PARTE-DE (PR parte-de SUL, SC parte-de SUL ...).

Para descobrir o valor de VELOCIDADE (ZONA-FRONTAL)
deve-se primeiro (P)erguntar ou (D)eduzir?
> p

Dominio de valores de VELOCIDADE (ZONA-FRONTAL):
> [0,100]

Para descobrir o valor de ATIVIDADE (ZONA-FRONTAL)
deve-se primeiro (P)erguntar ou (D)eduzir?
> p

Dominio de valores de ATIVIDADE (ZONA-FRONTAL):
> {"DESENVOLVENDO-SE", "ESTAVEL"}

Para descobrir o valor de CONDICAO-ALTO-NIVEL (SJC)
deve-se primeiro (P)erguntar ou (D)eduzir?

> d

Dominio de valores de CONDICAO-ALTO-NIVEL (SJC):
> {favoravel, desfavoravel}

Para descobrir o valor de C-G-E (SJC)
deve-se primeiro (P)erguntar ou (D)eduzir?
> d

Dominio de valores de C-G-E (SJC):
> {favoravel, desfavoravel}

A determinação do contexto de aplicação de regras de decisão - como é discutido no Capítulo 3 - pode requerer novas relações entre objetos.

CONTEXTO DE APLICACAO DE $R1

Como SJC se relaciona com ZONA-FRONTAL?

Aperte uma tecla para entrar com novas relacoes ...

AQUISICAO DE NOVA RELACAO

> particularizacao-de (zona-frontal,sistema-sinotico)

Neste caso o sistema ainda desconhece SISTEMA-SINÓTICO. Como as relações são entre objetos ou entre valores e ZONA-FRONTAL já é conhecido como objeto, SISTEMA-SINÓTICO deve ser um objeto.

SISTEMA-SINOTICO sera encarado como um OBJETO. Ok?
> s

AQUISICAO DE NOVA RELACAO

> associado-a(sistema-sinotico,sjc)
----- RELACAO ASSOCIADO-A -----

ASSOCIADO-A é uma relação ainda desconhecida do sistema. É preciso adquirir algumas informações sobre ela.

Nome da inversa de ASSOCIADO-A:
>

Se o usuário não especifica um nome para a relação inversa, o sistema o estabelece precedendo o nome da relação pelo símbolo " # ". Neste caso, #ASSOCIADO-A.

A relacao ASSOCIADO-A vale (S)empre
ou existem (C)ondicoes para estabelece-la?
> s

Deseja estabelecer propriedades para a relacao
ASSOCIADO-A?
> n

AQUISICAO DE NOVA RELACAO

CONTEXTO DE APLICACAO DE $R1

---- CONFIRMACAO DE RELACOES ----

ASSOCIADO-A (SISTEMA-SINOTICO,SJC)? > s
PARTICULARIZACAO-DE (ZONA-FRONTAL,SISTEMA-SINOTICO)? > s
Mais alguma relacao?
> n

---- CONTEXTO 1 ----

ASSOCIADO-A (SISTEMA-SINOTICO,SJC)
PARTICULARIZACAO-DE (ZONA-FRONTAL,SISTEMA-SINOTICO)

```
Ok?
> s

E' razoavel instanciar os objetos para este contexto
na ordem: SJC, SISTEMA-SINOTICO, ZONA-FRONTAL?
> s
```

FIM DE AQUISICAO

```
Regra $R1 adquirida.

Aperte uma tecla para continuar ...
```

AQUISICAO DE NOVA REGRA

```
REGRA $R2
SE:
     Q >
     1 > local(zona-frontal) = rs
     2 > velocidade(zona-frontal) = 20
     3 > atividade(zona-frontal) = "DESENVOLVENDO-SE"
     4 > condicao-alto-nivel(sjc) = favoravel
     5 >
ENTAO:
     1 > c-g-e(sjc,tempo = 24) = favoravel .6
     2 >
```

Como esta regra não introduz novos conceitos, o diálogo de aquisição é bem mais simples.

REGRA $R2

```
SE:
  1) LOCAL (ZONA-FRONTAL) = RS    &
  2) VELOCIDADE (ZONA-FRONTAL) = 20    &
  3) ATIVIDADE (ZONA-FRONTAL) = "DESENVOLVENDO-SE"    &
```

```
  4) CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL
ENTAO:
  1) C-G-E (SJC,TEMPO = 24) = FAVORAVEL  (FC = 0.60)

Ok?
> s
```

CONTEXTO DE APLICACAO DE $R2

```
---- CONFIRMACAO DE RELACOES ----

ASSOCIADO-A (SISTEMA-SINOTICO,SJC)?  >  s
PARTICULARIZACAO-DE (ZONA-FRONTAL,SISTEMA-SINOTICO)?  >  s
Mais alguma relacao?
> n

---- CONTEXTO 1 ----

    ASSOCIADO-A (SISTEMA-SINOTICO,SJC)
    PARTICULARIZACAO-DE (ZONA-FRONTAL,SISTEMA-SINOTICO)
Ok?
> s

E' razoavel instanciar os objetos para este contexto
na ordem: SJC, SISTEMA-SINOTICO, ZONA-FRONTAL?
> s
```

FIM DE AQUISICAO

```
Regra $R2 adquirida.

Aperte uma tecla para continuar ...
```

AQUISICAO DE NOVA REGRA

```
REGRA $R3
```

```
SE:
     Q>  todos(*x,*y)(*x > *y)
     1>  local (zona-frontal) = rs
     2>  velocidade(zona-frontal) = 20
     3>  intensidade(zona-frontal,tempo = *x) menor
               intensidade(zona-frontal,tempo = *y)      V
       >  velocidade(zona-frontal,tempo = *x) menor
               velocidade(zona-frontal,tempo = *y)
     4>  condicao-alto-nivel(sjc) = favoravel
     5>
ENTAO:
     1>  c-g-e(sjc,tempo = 24) = favoravel .2
     2>
```

Internamente, as variáveis são renomeadas de modo que seus escopos sejam correspondentes a apenas uma regra.

REGRA $R3

```
SE:
  Q) TODOS (*V1 *V2)
       (*V1 > *V2)
  1) LOCAL (ZONA-FRONTAL) = RS    &
  2) VELOCIDADE (ZONA-FRONTAL) = 20    &
  3.
    1) INTENSIDADE (ZONA-FRONTAL,TEMPO = *V1)   <
               INTENSIDADE (ZONA-FRONTAL,TEMPO = *V2)    OU
    2) VELOCIDADE (ZONA-FRONTAL,TEMPO = *V1)    <
               VELOCIDADE (ZONA-FRONTAL,TEMPO = *V2)
  4) CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL
ENTAO:
  1) C-G-E (SJC,TEMPO = 24) = FAVORAVEL  (FC = 0.20)

Ok?
> s
```

INFORMACOES ADICIONAIS SOBRE PREDICADOS

Para descobrir o valor de INTENSIDADE (ZONA-FRONTAL)
deve-se primeiro (P)erguntar ou (D)eduzir?
> d

Dominio de valores de INTENSIDADE (ZONA-FRONTAL):
> (10,50)

CONTEXTO DE APLICACAO DE $R3

---- CONFIRMACAO DE RELACOES ----

ASSOCIADO-A (SISTEMA-SINOTICO,SJC)? > s
PARTICULARIZACAO-DE (ZONA-FRONTAL,SISTEMA-SINOTICO)? > s
Mais alguma relacao?
> n

---- CONTEXTO 1 ----

ASSOCIADO-A (SISTEMA-SINOTICO,SJC)
PARTICULARIZACAO-DE (ZONA-FRONTAL,SISTEMA-SINOTICO)
Ok?
> s

E' razoavel instanciar os objetos para este contexto
na ordem: SJC, SISTEMA-SINOTICO, ZONA-FRONTAL?
> s

FIM DE AQUISICAO

Regra $R3 adquirida.

...

O usuário interrompe a aquisição de regras para atualizar a rede semântica.

MODULO DE AQUISICAO DE CONHECIMENTO

Inicializacao
Nova relacao
Nova regra ou meta-regra
Sair do modulo de aquisicao

AQUISICAO DE NOVA RELACAO

```
> influi-em ( bloqueio, sjc )

BLOQUEIO será encarado como um objeto. Ok?
> s

----- RELACAO INFLUI-EM -----
Nome da inversa de INFLUI-EM:
>

A relacao INFLUI-EM vale (S)empre
ou existem (C)ondicoes para estabelece-la?
> c

---- DEFINICAO DA RELACAO INFLUI-EM ----
INFLUI-EM (*X, *Y)

SE:
     Q >
     1 > local (*x) = *11
     2 > local (*y) = *12
     3 > $distancia (*11, *12, *d)
     4 > $menor (*d, 100)
     5 >
```

O usuário especifica as condições para que a relação INFLUI-EM seja estabelecida de uma forma geral, ou seja, entre quaisquer objetos ou valores. O sistema armazena estas condições como uma regra de decisão na qual a conclusão é a cláusula relacional INFLUI-EM (*X , *Y).

Deseja estabelecer propriedades para a relacao INFLUI-EM?
> n

AQUISICAO DE NOVA RELACAO

...

Após atualização da rede semântica o usuário retoma a aquisição de regras.

MODULO DE AQUISICAO DE CONHECIMENTO

Inicializacao
Nova relacao
Nova regra ou meta-regra
Sair do modulo de aquisicao

AQUISICAO DE NOVA REGRA

REGRA $R6
SE:
... Q > todos(*t) (*t ε [-24,0])
1 > condicao-pressao(sjc,tempo = *t) = favoravel
2 > dbu(sjc) < 3
3 >
ENTAO:
1 > condicao-local (sjc,tempo = 24) = favoravel .8
2 >

Esta regra introduz novos conceitos: condição-pressao, dbu (depressão do bulbo úmido) e condição-local. Notar que emprega-se "ε" como símbolo de pertinência.

REGRA $R6

```
SE:
  Q) TODOS (*V5)
       (*V5 PERTENCE [-24,0])
  1) CONDICAO-PRESSAO (SJC,TEMPO = *V5) = FAVORAVEL    &
  2) DBU (SJC) < 3
ENTAO:
  1) CONDICAO-LOCAL (SJC,TEMPO = 24) = FAVORAVEL  (FC = 0.80)
Ok?
> s
```

INFORMACOES ADICIONAIS SOBRE PREDICADOS

```
Para descobrir o valor de CONDICAO-PRESSAO (SJC)
deve-se primeiro (P)erguntar ou (D)eduzir?
> p

Dominio de valores de CONDICAO-PRESSAO (SJC):
> {favoravel, desfavoravel}

Para descobrir o valor de DBU (SJC)
deve-se primeiro (P)erguntar ou (D)eduzir?
> p

Dominio de valores de DBU (SJC):
> [0,10]

Para descobrir o valor de CONDICAO-LOCAL (SJC)
deve-se primeiro (P)erguntar ou (D)eduzir?
> d
```

Dominio de valores de CONDICAO-LOCAL (SJC):
> {favoravel, desfavoravel}

FIM DE AQUISICAO

Regra $R6 adquirida.

...

AQUISICAO DE NOVA REGRA

REGRA $R20
SE:
Q >
1 > mes(sjc) < 3 V
mes(sjc) > 10
2 >
ENTAO:
1 > estacao(sjc) = verao
2 >

REGRA $R20

SE:
1.
1) MES (SJC) < 3 OU
2) MES (SJC) > 10
ENTAO:
1) ESTACAO (SJC) = VERAO (FC = 1.00)

Ok?
> s

INFORMACOES ADICIONAIS SOBRE PREDICADOS

Para descobrir o valor de MES (SJC)

deve-se primeiro (P)erguntar ou (D)eduzir?
> p

Dominio de valores de MES (SJC):
> [1,12]

FIM DE AQUISICAO

Notar que neste caso existe referência a apenas um objeto - SJC - e portanto o grafo de contexto da regra é trivial.

Regra $R20 adquirida.

Aperte uma tecla para continuar ...

AQUISICAO DE NOVA REGRA

REGRA $R22
SE:
Q > condicao-local(sjc,tempo = 24) = favoravel
1 > ...
2 > estacao(sjc) = verao
3 >
ENTAO:
1 > precipitacao(sjc,tempo = 24) = sim , 0.6
2 >

REGRA $R22

SE:
1) CONDICAO-LOCAL (SJC,TEMPO = 24) = FAVORAVEL &
2) ESTACAO (SJC) = VERAO
ENTAO:
1) PRECIPITACAO (SJC,TEMPO = 24) = SIM (FC = 0.60)

Ok?

> s

FIM DE AQUISICAO

Regra $R22 adquirida.

...

O processo de aquisição continua até que o usuário escolha a opção 4.

MODULO DE AQUISICAO DE CONHECIMENTO

Inicializacao
Nova relacao
Nova regra ou meta-regra
Sair do modulo de aquisicao

FIM DE AQUISICAO DE CONHECIMENTO

Antes de sair do módulo de aquisição o sistema verifica se o usuário deseja salvar o banco de conhecimento produzido na sessão.

Salvar PREVISAO?
> s

...

2.6 - RESUMO

Neste capítulo discutiu-se como o conhecimento está representado no sistema MOTIN. Mostrou-se que existe uma separação entre o conhecimento empírico de resolução de problema e o conhecimento bem estabelecido do domínio de aplicação, e que para a representação empregam-se, no primeiro caso **regras de decisão** e no

segundo caso, um esquema de **rede semântica.** Foram então descritas, em detalhes, as características destes dois esquemas de representação.

Mostrou-se que, além do banco de conhecimento constituído por regras de decisão e da rede semântica que estrutura o conhecimento permanente do domínio de aplicação, existem duas outras estruturas de armazenamento: o **grafo de instâncias de objetos** e o **banco de evidências.** Estas duas estruturas estão associadas ao esquema de rede semântica e são particulares a um determinado problema do domínio, sendo portanto estruturas temporárias que podem ser armazenadas ou descartadas ao final de uma sessão de consulta ao sistema.

Foram exibidos alguns trechos de sessão de aquisição de conhecimentos pelo sistema MOTIN, a fim de ilustrar como, por meio de um diálogo com o sistema, o usuário pode atualizar o banco de conhecimentos e a rede semântica.

Um ponto importante e para o qual procurou-se chamar atenção é a **flexibilidade** dos esquemas de representação propostos. Mostrou-se então que para as regras de decisão é possível representar vários tipos de cláusulas - predicativas, relacionais e procedimentais -, além de permitir-se a introdução de novos primitivos conceituais com os quais as cláusulas são escritas. Para o esquema de rede semântica mostrou-se como é possível definir a semântica de novas relações. Estas características de flexibilidade são importantes para ferramentas de construção de sistemas especialistas pois permitem a adequação do esquema de representação aos domínios de aplicação específicos.

CAPÍTULO 3

CONTEXTOS DE APLICAÇÃO DE REGRAS DE DECISÃO

3.1 - O PROBLEMA DO CONTEXTO DE APLICAÇÃO

Como já se comentou anteriormente, numa regra de decisão do sistema MOTIN, os **objetos** são encarados como **variáveis** e dessa forma, para que uma regra seja aplicada essas variáveis precisam ser instanciadas. Comentou-se também que os objetos, à medida que vão sendo instanciados, constroem uma estrutura denominada **grafo de instâncias.** A determinação de quais instâncias são válidas para os objetos de uma regra constitui o problema de estabelecimento de **contexto de aplicação.**

Seja por exemplo:

```
SE:    LOCAL ( ZONA-FRONTAL ) = RIO-GRANDE-SUL   &
       VELOCIDADE ( ZONA-FRONTAL ) = 20          &
       CONDIÇÃO-ALTO-NÍVEL ( SJC ) = FAVORÁVEL
ENTÃO: C-G-E ( SJC ) = FAVORÁVEL
```

Para que esta regra possa ser aplicada é preciso em primeiro lugar verificar se existem, no grafo atual de instâncias, instâncias para seus dois objetos: ZONA-FRONTAL e SJC. Caso não existam é preciso criá-las. Existindo é preciso identificar quais dessas instâncias são válidas para a regra. Uma vez identificadas, as instâncias substituem os objetos e a regra pode então ser aplicada. Este capítulo discute como se processa cada um desses passos.

3.2 - GRAFOS DE CONTEXTO DE UMA REGRA

Os contextos de aplicação de uma regra são determinados com base em **grafos de contexto** estabelecidos pelo especialista quando a regra é fornecida ao sistema.

Entre dois objetos quaisquer pode existir, em geral, um conjunto de relações. Estas relações são as que aparecem na rede semântica que estrutura o conhecimento do domínio de aplicação e são entendidas como as relações possíveis entre aqueles objetos. Ao estabelecer um grafo de contexto o especialista escolhe algumas dessas relações (ou mesmo todas). Os contextos de aplicação serão estabelecidos posteriormente, instanciando-se os objetos que aparecem nos grafos de contexto de uma regra.

Para que uma regra tenha um grafo de contexto é preciso que todos os seus objetos estejam relacionados (não necessariamente todos entre si), isto é, não se permite que uma regra contenha um objeto que não se relaciona com nenhum dos demais, direta ou indiretamente. A razão disto é evitar que se escrevam regras sem sentido, nas quais o relacionamento entre premissa e conclusão não é de natureza causal.

Seja, por exemplo, a regra:

```
SE:   COR ( CAVALO , PÁREO = SEGUNDO ) = ALAZÃO   &
      COTAÇÃO-PARALELA ( DÓLAR ) > 30.00          &
      RUÍDO ( AMPLIFICADOR ) < 10
ENTÃO: COLUNA ( APOSTA , JOGO = 5 ) = MEIO
```

Vê-se por este exemplo - bastante exagerado - que se os objetos não estão relacionados, as regras podem se apresentar muito "esquisitas".

A ocorrência de uma situação dessas, ou seja, de uma regra considerada "esquisita" pelo sistema e na qual o especialista insiste

acreditar, irá indicar que o conhecimento que se dispõe do domínio de aplicação está incompleto e que novas relações deverão ser fornecidas.

A idéia de determinar os grafos de contexto junto ao especialista, baseia-se no pressuposto que o responsável por uma regra deve ter em mente os seus contextos de aplicação quando sugere esta nova regra ao sistema.

Em resumo, quando uma nova regra é fornecida ao sistema, determinam-se na rede semântica as relações que podem existir entre os objetos referenciados. Isto resulta num grafo (em geral, um multigrafo) conexo (se o grafo for desconexo, novas relações devem ser estabelecidas pelo especialista e a rede semântica, atualizada). O especialista então, com base neste grafo, estabelece os grafos de contexto da nova regra. Posteriormente, quando for preciso utilizar esta regra, através de instanciações dos objetos desses grafos, serão estabelecidos os seus contextos de aplicação.

3.3 - DETERMINAÇÃO DE GRAFOS DE CONTEXTO

Para ilustrar como se processa a determinação dos grafos de contexto, considere que a rede semântica apresenta os objetos e relações, como mostrados na Figura 3.1.

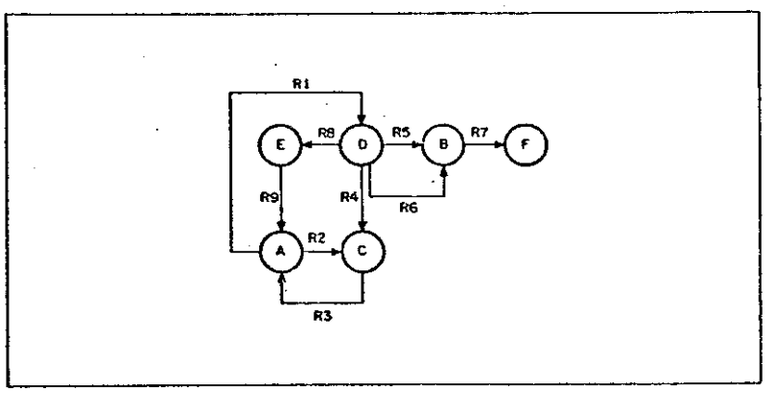

Fig. 3.1 - Estruturação de objetos na rede semântica

Considere que uma nova regra referencia os objetos A, B e C. Neste caso tem-se:

a) O sistema determina o subgrafo com menor número de objetos que envolve A, B e C, o que resulta no grafo da Figura 3.2.

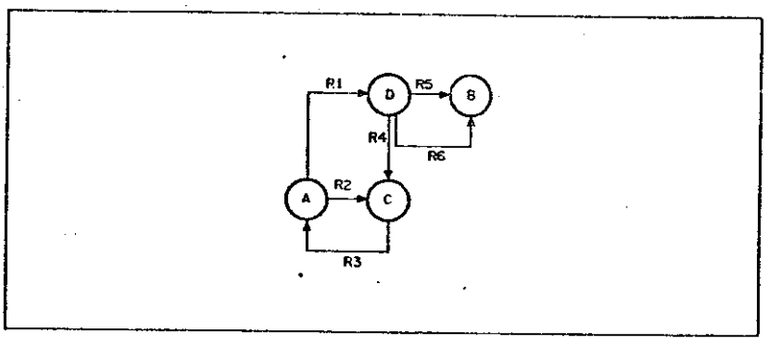

Fig. 3.2 - Grafo de objetos

A motivação para se partir do **menor** subgrafo é a de tentar estabelecer os contextos mais simples possíveis, ou seja, contextos nos quais aparece o menor número possível de objetos que não são diretamente referenciados pela regra.

b) O especialista especifica as relações que realmente devem ser estabelecidas como contexto para a nova regra. Neste processo, permite-se que outros objetos ou relações sejam introduzidas e que o subgrafo seja modificado. Seja por exemplo, o grafo da Figura 3.3.

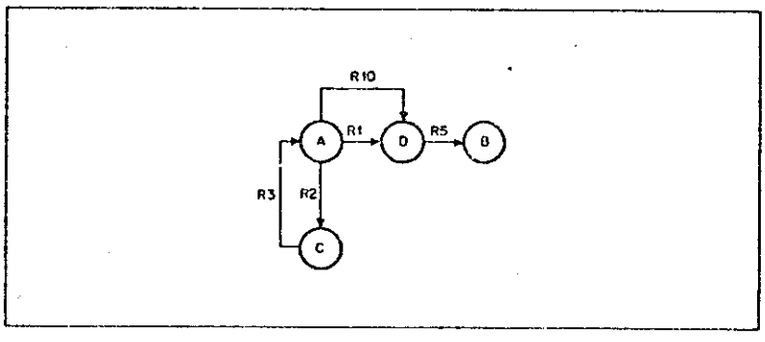

Fig. 3.3 - Grafo de objetos modificado

Este grafo ilustra o caso em que o especialista, com base no grafo da Figura 3.2:

- refuta as relações R4 e R6, isto é, estabelece que a regra não deve (ou não precisa) usar as relações R4 e R6 para ser aplicada;
- introduz uma nova relação (R10) que, como não ocorre na rede semântica, é encarada como circunstancial.

c) No caso do grafo resultante da operação (b) conter relações múltiplas entre objetos, o especialista deve especificar se essas relações são de um único contexto ou de vários contextos. Para o exemplo acima, o especialista especifica se em cada um dos pares (R2,R3) e (R1,R10), as relações devem ser consideradas num único contexto ou em contextos distintos. Considere, por exemplo, que R1 e R10 devem aparecer juntas num contexto, mas que R2 e R3 são de contextos separados. Neste caso, tem-se os **dois** grafos de contexto mostrados na Figura 3.4.

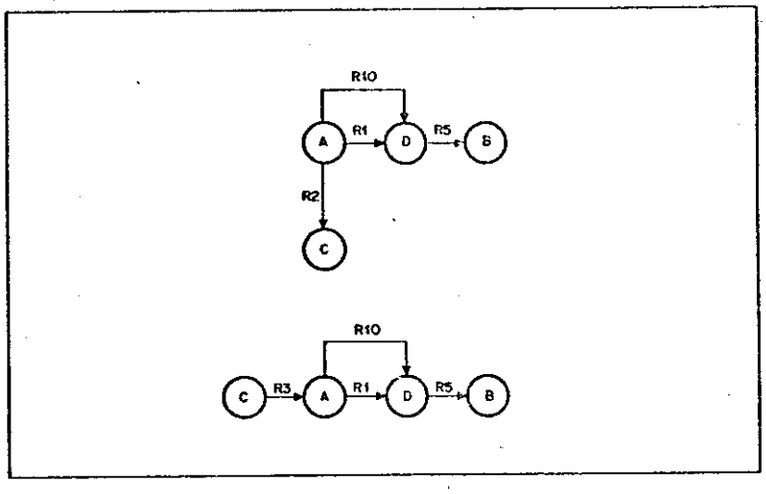

Fig. 3.4 - Grafos de contexto

Mostra-se a seguir o algoritmo para a determinação dos menores subgrafos que, como no passo (a) acima, irão orientar a especificação dos grafos de contexto de uma regra.

ALGORITMO 3.1: MENOR-SUBGRAFO (L)

L é o conjunto de objetos de uma regra. Este algoritmo determina os menores subgrafos de objetos nos quais todos os elementos de L estão envolvidos. Caso nem todos os elementos de L estiverem relacionados (direta ou indiretamente), não existirão tais subgrafos e o algoritmo fracassa. O procedimento retorna, caso existam, os conjuntos de objetos dos subgrafos encontrados ou então, em caso de fracasso, o conjunto N (N $\subseteq$ L) de objetos que não se relacionam com nenhum dos demais elementos de L. Com base no conjunto N novas relações deverão ser requisitadas ao especialista para atualização da rede semântica que estrutura o domínio de aplicação.

Seja L = {O1, O2, ..., On} e sejam:

AB - um conjunto de pares [P , F] onde P corresponde a uma **solução parcial**, isto é, ao conjunto de objetos de um subgrafo que envolve apenas parte dos elementos de L, e F (F $\subseteq$ L) corresponde ao conjunto de objetos que ainda não aparecem em P. Determina-se uma **solução** S quando existe em AB um elemento [S, ϕ].
SG - conjunto atual de soluções.
CS - custo da solução, o que corresponde à **cardinalidade** (#) do conjunto de objetos de um subgrafo solução.

Seja X um objeto e VIZINHOS(X) o conjunto {V1, ..., Vk} de objetos tais que para todo i = 1, ..., k existe pelo menos uma relação R entre X e Vi, ou seja, existe R(X,Vi) ou R(Vi,X) na rede semântica.

(1) Fazer:
AB = {[[O1, ϕ] , {O2, ..., On}]};
SG = ϕ;
CS = infinito;
N = L.
(2) Se AB = ϕ então:
Se SG = ϕ então retornar N.
Caso contrário, retornar SG.

Caso contrário:

(3) Seja AB = {[P1,F1] , ..., [Pm,Fm]}. Seja Pi = [Qi,Ri] onde:

Qi U Ri corresponde ao conjunto de objetos de uma solução parcial, com:

Qi = { q1, ..., qki} , ki ≥ 1 - conjunto de objetos da solução parcial cujos vizinhos ainda não foram considerados (objetos ainda não **expandidos**), e

Ri - conjunto de objetos da solução parcial, já expandidos.

Para cada i = 1, ..., m fazer:

(a) Se Fi = ϕ então:

Se ♯ (Qi U Ri) < CS então SG = Qi U Ri;

Se ♯(Qi U Ri) = CS então SG = SG U Qi U Ri;

Retirar [Pi,Fi] de AB.

Caso contrário:

(b) Se ♯ (Qi U Ri) ≥ CS, ou seja, se o custo de uma solução parcial é maior ou igual ao custo de uma solução já encontrada, então: retirar [Pi,Fi] de AB.

Caso contrário:

(c) Seja V = {v1, ..., vq} = VIZINHOS (q1).

(d) Retirar [Pi,Fi] de AB e para cada j = 1, ..., q, fazer:

N = N - {vj};

AB = AB U {[Pij , Fij]},

onde:

Pij = [Qij , Rij] com: Qij = Qi - {q1} U {vj};

Rij = Ri U {q1},

Fij = Fi - {vj}.

(4) Voltar ao passo (2).

Após determinar os grafos de contexto de uma regra, é importante especificar a **ordem de instanciação dos objetos** que aparecem nesses grafos. Para o caso dos objetos A e C na Figura 3.4, por exemplo, qual é a ordem mais natural de instanciação: (A C) ou (C A)? Por questões de generalidade, a ordem de instanciação é também especificada pelo especialista. O sistema, no entanto, irá sugerir uma ordenação ou poderá considerar uma ordenação estabelecida anteriormente.

Apresenta-se a seguir, o algoritmo utilizado para sugerir ao especialista uma ordem de instanciação de objetos de um conjunto de grafos de contexto.

ALGORITMO 3.2 - ORDENAR-OBJETOS (G , L)

G = { G1, ..., Gn } é um conjunto de grafos de contexto e L = { O1, ..., Ok } é o conjunto de objetos de cada um dos grafos Gi (i = 1, ..., n). Este algoritmo determina uma relação de ordenação parcial ("ANTES-DE") entre os elementos de L para sugerir ao especialista como ordem de instanciação de objetos dos grafos na especificação dos contextos de uma regra. O algoritmo considera a existência de uma relação do objeto Oi para o objeto Oj como uma indicação de que Oj deve ser instanciado **antes de** Oi.

Sejam os grafos de contexto: Gi = [L , Ri] (i = 1, ..., n) com:

Ri = conjunto de relações de Gi =
= {ri1(ai1,bi1), ..., riki(aiki,biki) / (aij,bij) pertence a L x L, com j = 1, ..., ki}.

Seja c(X,Y) = ϕ para todo (X,Y) $\in$ L x L.

(1) Para todo i = 1, ..., n tal que existe ri(aij,bij) pertencente a Ri, com j = 1, ..., ki , fazer:

c(aij, bij) = c(aij, bij) U { rij }.

(2) Para cada par (Oi, Oj) de L x L, definir a relação "ANTES-DE" por:

Oi "ANTES-DE" Oj $\leftrightarrow$ # (c(Oi,Oj)) < # (c(Oj,Oi)).

Portanto, para levar em conta os conjuntos de grafos de contexto estabelecidos e a ordem de instanciação dos objetos de cada um desses conjuntos, as regras de decisão no sistema MOTIN são representadas internamente como uma tripla:

(premissa conclusão contexto)

na qual contexto é uma lista do tipo:

((O1 (G11 G12 ... G1n1)) ... (Om (Gm1 Gm2 ... Gmnm)))

onde: Oi (i = 1, ..., m) é a ordem de instanciação dos objetos que aparecem nos grafos de contexto Gij (j = 1, ..., ni).

3.4 - ESTABELECIMENTO DE CONTEXTOS DE APLICAÇÃO

Para a utilização de uma regra é preciso estabelecer seus contextos de aplicação, o que é feito com base nos seus grafos de contexto e no grafo atual de instâncias de objetos. As relações que existem entre as instâncias de objetos no grafo de instâncias são denominadas **relações de contexto.**

Sejam:

I - grafo atual de instâncias,
R - uma regra de decisão, e
G - um grafo de contexto de R.

Diz-se que C(R,G) é um **contexto de aplicação** de R com relação ao grafo G se para toda relação r(O1,O2) de G existe uma relação de contexto r(I1,I2) em I tal que Ij (j = 1,2) é instância de Oj, ou seja, se C(R,G) é um subgrafo de I isomorfo a G.

Como se comentou anteriormente no Capítulo 2, existem dois tipos de relações entre objetos na rede semântica: **relações estruturais** e **relações condicionais.** As primeiras são relações que valem sempre entre os objetos, quaisquer que sejam suas instâncias, enquanto que as últimas, para que sejam efetivamente estabelecidas, possuem uma definição que precisa ser satisfeita. Portanto, uma relação condicional pode não existir para um par específico de instâncias de objetos. Em outras palavras: no grafo de contexto de uma regra (e na rede semântica) poderão existir relações que não existem no grafo atual de instâncias e nesse caso não existirão contextos de aplicação com relação àquele grafo de contexto.

Assim, para estabelecer os contextos de aplicação de uma regra, o primeiro passo é determinar instâncias dos objetos de seus grafos de contexto (caso elas ainda não existam no grafo atual de instâncias) e determinar também os seus relacionamentos.

Para as instanciações de objetos - feitas segundo a ordenação estabelecida para o grafo de contexto - emprega-se o Algoritmo 3.3, mostrado a seguir.

ALGORITMO 3.3 - INSTANCIAR-OBJETO (OBJ , GC , GI)

OBJ é o objeto cujas instâncias se quer determinar, GC é um grafo de contexto e GI é o grafo atual de instâncias. O algoritmo determina o conjunto $I = \{I1, \ldots, Ik\}$ $(k \geq 0)$ de instâncias de OBJ e para cada $j = 1, \ldots, k$, atualiza o grafo de instâncias com Ij e com as relações de GC nas quais OBJ participa.

(1) Fazer: $I = \phi$.

(2) Fazer ao especialista a pergunta:

EXISTE ALGUM OBJ
tal que:
R1(X11 , X12)
...
Rk(Xk1 , Xk2) (possivelmente)
...
Rn(Xn1 , Xn2)
dado que:
S1(Y11 , Y12)
...
Sm(Ym1 , Ym2) ?

onde:

(a) Ri (i = 1, ..., n) e Sj (j = 1, ..., m) são relações de GC (a mensagem "possivelmente" é mostrada para relações condicionais);

(b) para cada i = 1, ..., n tem-se que:

Xi1 = OBJ , Xi2 ε GI ou

Xi1 ε GI , Xi2 = OBJ.

Seja X = { Xij , i = 1, ..., n ; j = 1,2 tal que Xij é diferente de OBJ }.

(c) Yij são instâncias de objetos do grafo de contexto GC tais que para todo i = 1, ..., m, existe pelo menos um índice j = 1,2 tal que Yij ε X

(3) Enquanto a resposta à pergunta feita em (2) for **afirmativa,**

(a) Fazer: I = I ∪ { nova instância de OBJ };

(b) Atualizar o grafo GI representando para essa nova instância, as relações Ri (i = 1, ..., n) que puderem ser estabelecidas;

(c) Voltar ao passo (2), trocando a mensagem "EXISTE ALGUM" por "EXISTE MAIS ALGUM".

(4) Se # (I) = 0, ou seja, se não foi possível determinar instâncias para OBJ, então:

(a) Armazenar em que situação as instâncias deste objeto não ocorrem;

(b) Retornar "FRACASSO".

Caso contrário, retornar I.

Considere por exemplo:

- que o grafo atual de instâncias é como o mostrado na Figura 3.5;
- que o grafo de contexto de uma regra é como o da Figura 3.6; e
- que se deseja instanciar o objeto JATO.

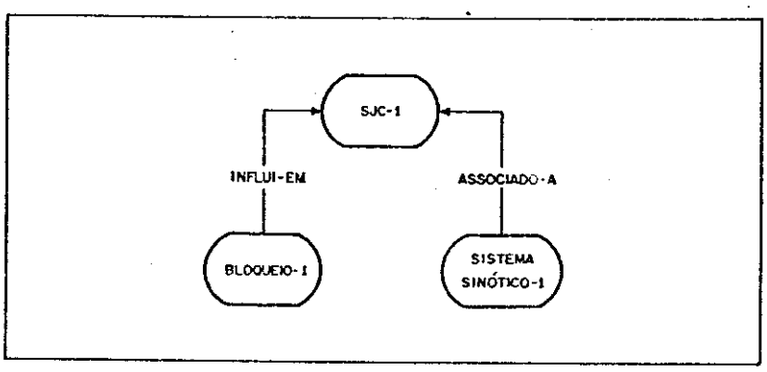

Fig. 3.5 - Grafo de instâncias

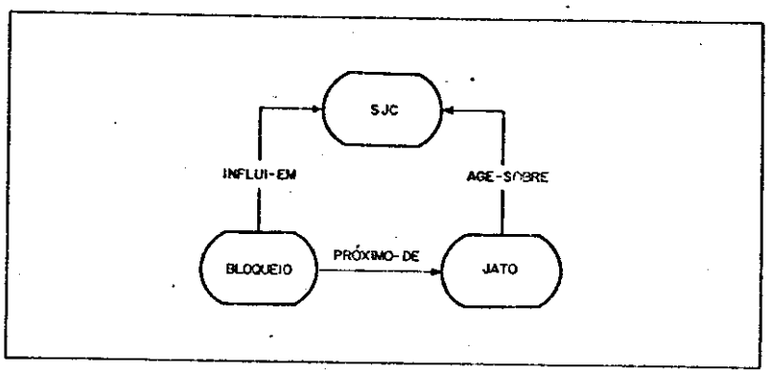

Fig. 3.6 - Grafo de contexto

Neste caso as perguntas terão a forma da Figura 3.7.

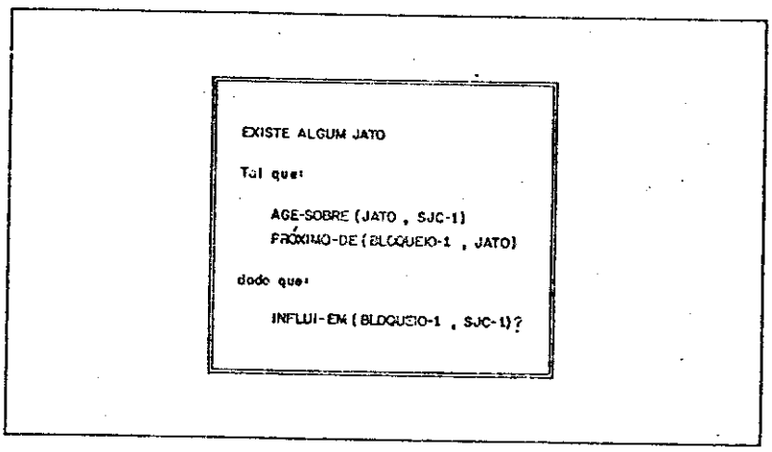

Fig. 3.7 - Exemplo de pergunta para instanciar objeto

Uma vez instanciados os objetos, as relações de contexto são representadas no grafo de instâncias. Para as relações estruturais este processo é imediato. Para as condicionais, avalia-se a definição da relação. Caso a avaliação resultar verdadeira, a relação de contexto é representada explicitamente no grafo de instâncias, a fim de que sua definição não seja avaliada novamente para aquele par específico de instâncias de objetos.

Se não for possível a instanciação dos objetos de uma regra (isto é, se o usuário responder NÃO a uma pergunta como a da Figura

3.7) ou não for possível representar alguma relação de contexto condicional, não haverá como estabelecer contextos de aplicação e portanto, a regra não é aplicada.

Considere, apenas para ilustrar, que o especialista responde que existem dois jatos em altos níveis nas condições da Figura 3.7: JATO-1 e JATO-2. Neste caso o grafo de instâncias é atualizado e passa a ser como o da Figura 3.8.

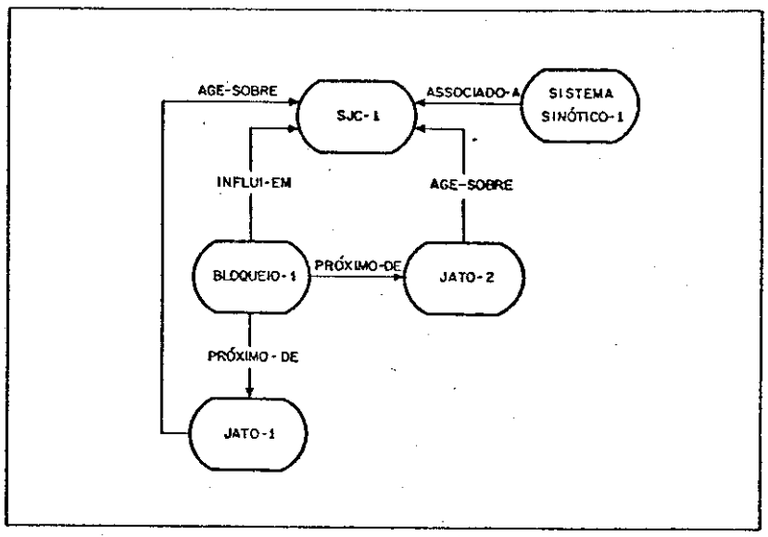

Fig 3.8 - Grafo de instâncias atualizado

Portanto para este caso, a regra teria, com relação ao grafo de contexto da Figura 3.6, dois contextos de aplicação:

(SJC-1 , BLOQUEIO-1 , JATO-1) e
(SJC-1 , BLOQUEIO-1 , JATO-2).

Como foi observado anteriormente (ver passo (4) do Algoritmo 3.3), quando não for possível instanciar um determinado objeto, isto é, quando o especialista responder "NÃO" a uma pergunta como a da Figura 3.7, a fim de evitar a repetição de perguntas, é preciso armazenar a situação em que instâncias daquele objeto não ocorrem. Para isso, representa-se explicitamente (utilizando o prefixo

"NÃO-") a não existência de cada uma das relações mostradas na parte **"tal que:"** da pergunta.

Para o caso da Figura 3.7, por exemplo, seriam representadas:

NÃO-AGE-SOBRE(JATO , SJC-1) e
NÃO-PRÓXIMO(BLOQUEIO-1 , JATO).

Deve-se observar que estas relações, sendo estabelecidas entre um **objeto** e uma **instância de objeto**, não são relações de contexto. Evidentemente tais relações são temporárias, sendo removidas quando o grafo de instâncias é destruído ao final de uma sessão.

Apresenta-se a seguir o algoritmo utilizado para estabelecer os contextos de aplicação de uma regra de decisão no sistema MOTIN.

ALGORITMO 3.4 - INSTANCIAR-GRAFO-CONTEXTO (O , G)

O = [O1, ..., On] é a ordem em que os objetos do grafo de contexto G devem ser instanciados. Este algoritmo considera a cada passo, o grafo de instâncias I atualizado e retorna um conjunto de contextos de aplicação relativos a G, ou seja, um conjunto C = { C1, ..., Cm } onde cada contexto Ci = [Ii1, ..., Iin] (i = 1, ..., m) é tal que Iij é instância de Oj (j = 1, ..., n) e para toda relação R(Ou,Ov) G, existe R(Iu,Iv) ε I.

(1) Aplicar o algoritmo INSTANCIAR-OBJETO (O1 , G , I) para determinar o conjunto { I11, ..., I1m1 } de instâncias de O1.
Se m1 = 0 então retornar ϕ .
Caso contrário:
(2) Fazer:
(a) m = m1;
(b) Para cada j = 1, ..., m fazer: Cj = {I1j}.
(3) Para cada i = 2, ..., n fazer:
(a) Para cada j = 1, ..., m fazer: Aj = Cj U {Oi};

(b) Aplicar o algoritmo INSTANCIAR-OBJETO (Oi, G, I) para determinar o conjunto {Ii1, ..., Iimi} de instâncias de Oi.
Se mi = 0 então retornar ϕ .
Caso contrário:

(c) Para cada k = 1, ..., mi fazer:
Para cada j = 1, ..., m fazer:
Cj+(k-1)*m = conjunto resultante de substituir em Aj, Oi por Iik.

(d) Fazer: m = m * mi.

(4) Retornar C = {C1, ..., Cm}.

3.5 - RESUMO

Neste capítulo discutiu-se o problema do contexto de aplicação das regras de decisão que compõem o banco de conhecimento de um sistema especialista.

Numa regra de decisão do sistema MOTIN os **objetos** são considerados como variáveis e para que a regra possa ser aplicada estas variáveis precisam estar instanciadas. Em sistemas especialistas já desenvolvidos e que trabalham com domínios nos quais podem existir várias instâncias de um mesmo objeto a aplicação de regras é sempre orientada por uma **árvore de contexto.** Isto vale apenas quando os objetos se relacionam de maneira trivial.

Mostrou-se neste capítulo que é possível empregar o conhecimento armazenado em rede semântica, no que diz respeito à estrutura mantida pelos objetos do domínio, para resolver o problema do contexto de aplicação de regras e desta forma generalizar o mecanismo que vem sendo empregado.

CAPÍTULO 4

TRATAMENTO DA INCERTEZA E COMBINAÇÃO DE EVIDÊNCIAS

4.1 - INTRODUÇÃO

A representação e a utilização de conhecimento incompleto e impreciso são muito importantes num sistema especialista pois, como já se comentou anteriormente no Capítulo 1, para muitas aplicações pode-se dispor apenas de informação parcial e dados incertos ou contraditórios.

Muitas abordagens para o tratamento de incerteza já foram propostas e utilizadas em sistemas especialistas para diversos domínios de conhecimento. Podem-se citar, por exemplo, o enfoque probabilístico com o emprego do **teorema de Bayes** (Duda et alii, 1977), o modelo de **fatores de certeza** (Shortliffe and Buchanan, 1975) e a **lógica nebulosa** ("fuzzy logic") que utiliza o conceito de distribuição de possibilidades para modelar o conhecimento de senso comum (Zadeh, 1979). As vantagens e desvantagens dessas abordagens são bastante conhecidas e largamente repertoriadas na literatura (Prade, 1982; Weiss and Kulikowski, 1984; Negoita, 1985).

Nos últimos anos a atenção tem se voltado para o emprego da teoria de Dempster-Shafer (Shafer, 1976) para o tratamento da informação incompleta e incerta em sistemas especialistas (Friedman, 1981; Garvey et alii, 1981; Wesley and Hanson, 1982). Uma vantagem desta abordagem está na sua capacidade de modelar a "granularidade" da resposta a um certo problema, à medida que as evidências se acumulam. Um médico, por exemplo, como apontam Gordon e Shortliffe (1985), pode iniciar o tratamento de um caso de icterícia colestática sabendo apenas que o problema se deve a um distúrbio dentro do próprio fígado (ou seja, que se trata de um caso de colestase intrahepática) até que,

dispondo de novas evidências pode verificar se o problema decorre de uma doença específica como hepatite ou cirrose. Essa maneira de pensar em níveis "altos" de abstração - no exemplo acima, o médico pode descartar uma série de doenças sabendo apenas que não se trata de um caso de colestase extrahepática - parece caracterizar o raciocínio humano, o que torna interessante poder-se dispor dessa capacidade em sistemas especialistas.

A complexidade da teoria de Dempster-Shafer e sua conseqüente ineficiência computacional, no entanto, vem desencorajando o seu uso em sistemas práticos e motivando o estudo de adaptações para situações específicas, como por exemplo:

- quando os dados do problema confirmam ou rejeitam apenas proposições individuais, sem a possibilidade de confirmar ou rejeitar classes de proposições (Barnett, 1981);
- quando as classes de proposições formam uma estrutura de árvore e é razoável transferir a crença em uma classe que não pertence à árvore (classe para a qual não há interesse) para uma classe mais interessante (Gordon and Shortliffe, 1985); ou
- quando as classes formam uma estrutura de árvore mas não se transferem crenças de uma classe para outra, obtendo resultados exatos dentro da teoria (Shafer and Logan, 1985).

Embora esses enfoques aumentem bastante a eficiência do processo, existem problemas em empregá-los. No caso do enfoque de Barnett (1981) - cujo algoritmo é de complexidade linear - não se trabalha com classes de valores, inviabilizando-o quando se quer raciocinar em níveis "altos" de abstração. Já nos enfoques de Gordon e Shortliffe (1985) e Shafer e Logan (1985) apesar de ser possível trabalhar com classes de valores, a estrutura construída por essas classes - uma árvore - pode inviabilizar o seu emprego em determinados domínios de aplicação.

Neste capítulo discutem-se as questões relativas ao tratamento de incerteza e combinação de evidências adotadas no sistema MOTIN, dentro de uma aproximação da teoria de Dempster-Shafer na qual

é possível trabalhar com classes de valores estruturados pelo especialista, de acordo com as necessidades de uma aplicação e sem se restringir a uma estrutura de árvore.

4.2 - FUNDAMENTOS DA TEORIA DE DEMPSTER-SHAFER

Para motivar a exposição da teoria de Dempster-Shafer, considere, por exemplo, que os valores do atributo FORMA de um objeto X podem ser estruturados em classes de interesse semântico, conforme a Figura 4.1.

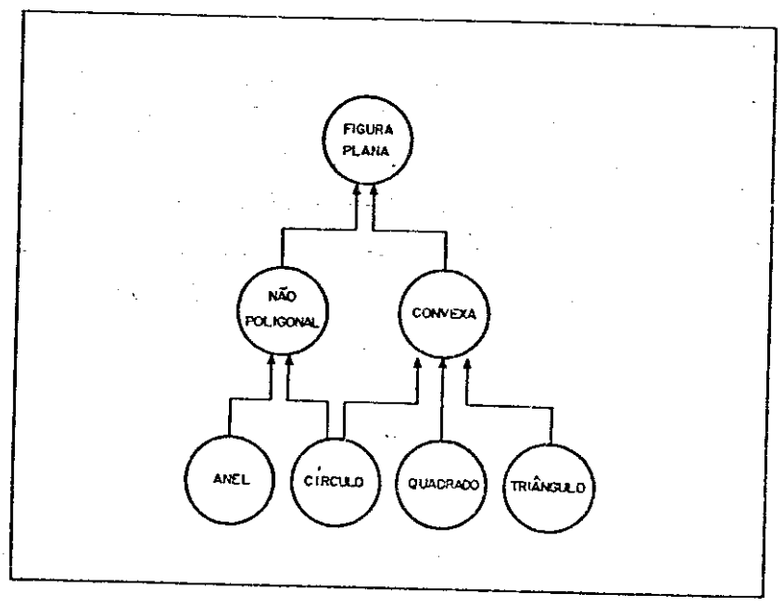

Fig. 4.1 - Estruturação de valores

Portanto, a FORMA(X) é a de uma FIGURA-PLANA que pode ser uma figura NÃO-POLIGONAL ou uma figura CONVEXA. No caso de NÃO-POLIGONAL tem-se que a FORMA(X) pode ser a de um ANEL ou a de um CÍRCULO e no caso de CONVEXA, a de um CÍRCULO, a de um QUADRADO ou a de um TRIÂNGULO. Em outras palavras, as classes FIGURA-PLANA, NÃO-POLIGONAL e CONVEXA são tais que:

FIGURA-PLANA = { NÃO-POLIGONAL, CONVEXA }

NÃO-POLIGONAL = {ANEL, CÍRCULO}
CONVEXA = { CÍRCULO, QUADRADO, TRIÂNGULO }

Na teoria de Dempster-Shafer as quatro formas básicas para o objeto X, constituem o conjunto θ = { ANEL, CÍRCULO, QUADRADO, TRIÂNGULO } denominado "frame of discernment" de FORMA(X). Admite-se nesta teoria que os valores de θ são **exaustivos** e **mutuamente exclusivos.**

Imagine, para o caso acima, que as evidências disponíveis sugerem que a crença em FORMA(X) = CONVEXA é de 0.8, que a crença em FORMA (X) = POLIGONAL é de 0.5 e que a crença em FORMA(X) = NÃO-QUADRADO é de 0.9 (notar que NÃO-QUADRADO corresponde à classe { ANEL, CÍRCULO, TRIÂNGULO }). Neste caso qual será a crença em FORMA(X) = TRIÂNGULO? Pela teoria de Dempster-Shafer esta crença será obtida considerando que:

crença(TRIÂNGULO) =
crença({ CÍRCULO, QUADRADO, TRIÂNGULO }
∩ { QUADRADO, TRIÂNGULO }
∩ { CÍRCULO, ANEL, TRIÂNGULO }) =
crença(CONVEXA ∩ POLIGONAL ∩ NÃO-QUADRADO) = 0.8 * 0.5 * 0.9 = 0.36

Nesta teoria portanto, a combinação de evidências baseia-se em noções da teoria dos conjuntos, sendo importante poder associar crença a qualquer subconjunto de θ . Observe que para o caso acima, existe evidência que sugere FORMA(X) = POLIGONAL e que, segundo a estrutura da Figura 4.1, a classe { QUADRADO, TRIÂNGULO} não é de interesse semântico.

A necessidade de considerar todos os subconjuntos do "frame of discernment", leva a algoritmos de combinação de evidências de complexidade exponencial, desmotivando o emprego da teoria de Dempster-Shafer em sistemas práticos.

Mais formalmente, seja $\theta = \{ v1, \ldots, vn \}$ o conjunto de valores para um atributo A de um objeto O, tal que:

$A(O) = v1$ ou ... ou $A(O) = vn$ e

$A(O) = x \leftrightarrow x \in \theta$

ou seja, v1, ..., vn são valores **exaustivos** e **mutuamente exclusivos** de A(O).

Seja $P(\theta)$ o conjunto dos subconjuntos de θ. Define-se uma **função de atribuição de crença** f sobre θ como:

$f : P(\theta) \rightarrow [0,1]$ tal que:

(a) $f(\phi) = 0$;

(b) $\Sigma\ f(x) = 1; \quad x \in P(\theta)$

Seja por exemplo, para as classes da Figura 4.1, a função de atribuição de crença dada por $f : P(\theta) \rightarrow [0,1]$ tal que:

$f(A) = 0.2$; $f(C) = 0.3$; $f(CQT) = 0.4$; $f(\theta) = 0.1$

e

$f(\phi) = f(Q) = f(T) = f(AC) = f(CQ) = f(CT) = f(AQ) = f(AT) = f(QT) = f(ACQ) = f(ACT) = f(AQT) = 0$,

onde A corresponde à classe { ANEL }, C à classe { CÍRCULO }, CQ à classe { CÍRCULO, QUADRADO } e assim por diante (observe que θ corresponde à classe ACQT).

Dada uma função de atribuição de crença f sobre θ, define-se a **credibilidade** (Cr) de uma classe $x \in P(\theta)$ como:

$$Cr(x) = \Sigma\ f(y); \quad y \subseteq x$$

Considerando o exemplo anterior, tem-se:

$Cr(A) = f(A) = 0.2$

$Cr(C) = f(C) = 0.3$

$Cr(CQT) = f(C) + f(Q) + f(T) + f(CQ) + f(CT) + f(QT) + f(CQT)$

$= 0.3 + 0 + 0 + 0 + 0 + 0 + 0.4 = 0.7$

$Cr(ACQT) = Cr(\theta) = 1$

A **plausibilidade** (Pl) de uma classe $x \in P(\theta)$ é definida por:

$$Pl(x) = 1 - Cr(-x)$$

onde $-x$ é o complemento da classe x em relação a θ .

Considerando os valores anteriores, tem-se por exemplo:
x = CONVEXA = {CÍRCULO, QUADRADO, TRIÂNGULO} = CQT
-x = θ - {CÍRCULO, QUADRADO, TRIÂNGULO} = {ANEL} = A
e portanto: Pl(CQT) = 1 - Cr(A) = 1 - 0.2 = 0.8

Pode-se mostrar (vide Apêndice A) que $Cr(x) \leq Pl(x)$ para todo $x \in P(\theta)$ isto é, $Cr(x) \leq 1 - Cr(-x)$, ou seja: $Cr(x) + Cr(-x) \leq 1$.

Define-se **intervalo de crença** I de uma classe x de $P(\theta)$ por:

$$I(x) = [Cr(x) , Pl(x)].$$

Portanto, num intervalo de crença I(x) tem-se:
Cr(x) = **crença mínima** de que a evidência **confirma** x;
Pl(x) = **crença máxima** de que a evidência **confirma** x
= 1 - **crença mínima** de que a evidência **rejeita** x.

Por exemplo: I(CQT) = [0.7, 0.8], ou seja acredita-se na classe CONVEXA com um fator não menor do que 0.7 e não maior do que 0.8.

A largura do intervalo de crença I(x) é portanto:

$$1 - Cr(x) - Cr(-x) \geq 0$$

e representa a crença que não pode ser associada nem a x nem à negação de x, dada a evidência disponível (notar que no caso Bayesiano, Cr(x) + Cr(-x) = 1 e a largura do intervalo de crença é zero).

Pode-se mostrar também que a largura do intervalo de crença I(x) corresponde à soma das crenças associadas aos subconjuntos de que interceptam x mas que não são subconjuntos de x. Para o exemplo acima, a largura de I(CQT) = 0.8 - 0.7 = 0.1. Neste caso os

subconjuntos de θ que interceptam CONVEXA e que não são seus subconjuntos são NÃO-POLIGONAL e θ e

$$f(\text{NÃO-POLIGONAL}) + f(\theta) = 0 + 0.1 = 0.1.$$

Num intervalo de crença para uma classe x, a credibilidade aumenta à medida que novas evidências confirmam x e a plausibilidade diminui à medida que novas evidências rejeitam x. Um intervalo de crença [0,1] indica **ignorância total** sobre a veracidade ou falsidade de uma classe; [1,1] indica que a classe é **absolutamente verdadeira** e [0,0], **absolutamente falsa.**

Sejam f1 : $P(\theta) \rightarrow [0,1]$ e f2 : $P(\theta) \rightarrow [0,1]$ duas funções de atribuição de crença sobre θ . A função de **atribuição de crença total** f é obtida pela **combinação** de f1 e f2 dada por:

f : $P(\theta) \rightarrow [0,1]$ = f1 (+) f2 (**soma ortogonal** de f1 e f2)

tal que:

(a) $f(\phi) = 0$;

(b) $f(x) = k \sum f1(y) * f2(z)$

com: $x = y \cap z$, $x \neq \phi$;

$k = 1/(1 - \sum f1(y) * f2(z))$ com: $y \cap z = \phi$.

Seja por exemplo:

$$\theta = A , C , Q , T$$

e

x	f1(x)
A	0.2
C	0.3
CQT	0.4
θ	0.1

x	f2(x)
Q	0.5
AQ	0.3
θ	0.2

A soma ortogonal de f1 e f2 será:

f1 / f2	(0.2) A	(0.3) C	(0.4) CQT	(0.1) θ
(0.5) Q	ϕ (0.10)	ϕ (0.15)	Q (0.20)	Q (0.05)
(0.3) AQ	A (0.06)	ϕ (0.09)	Q (0.12)	AQ (0.03)
(0.2) θ	A (0.04)	C (0.06)	CQT (0.08)	θ (0.02)

Logo:

$$k = \frac{1}{1 - (0.15 + 0.10 + 0.09)} = \frac{1}{1 - 0.34} = 1.52$$

e portanto, f = f1 (+) f2 é dada por:

f(A) = 1.52 * (0.04 + 0.06) = 0.15
f(C) = 1.52 * 0.06 = 0.09
f(Q) = 1.52 * (0.20 + 0.05 + 0.12) = 0.56
f(AQ) = 1.52 * 0.03 = 0.05
f(CQT) = 1.52 * 0.08 = 0.12
f(θ) = 1.52 * 0.02 = 0.03
f(x) = 0 para as demais classes x ε P(θ).

Pode-se observar que dadas duas funções de atribuição de crença f e g sobre tais que:

f(x) = m g(x) = n
f(θ) = 1 - m g(θ) = 1 - n

ou seja, duas funções que atribuem crença à classe x, independentemente, a crença total em x será dada pela soma ortogonal de f e g, ou seja:

f \ g	x (n)	θ (1 - n)
x (m)	x (m * n)	x (m * (1 - n))
θ (1 - m)	x (n * (1 - m))	θ (1 - m) * (1 - n)

portanto:

f (+) g(x) = m * n + m * (1 - n) + n * (1 - m)
= m * n + m - m * n + n - m * n

$= m + n - m * n$

o que corresponde à regra de Bernoulli de combinação de evidências do **modelo de fatores de certeza.**

4.3 - REGRAS DE DECISÃO COMO FUNÇÕES DE ATRIBUIÇÃO DE CRENÇA

A certeza depositada nos valores conhecidos para os atributos dos objetos de uma determinada aplicação, é representada no sistema MOTIN, por um **intervalo de crença** que armazena a **credibilidade** (crença mínima) e a **plausibilidade** (crença máxima) de uma determinada evidência. Por exemplo, I (VELOCIDADE (ZONA-FRONTAL-1) > 10) = [0.5,0.8] significa que acredita-se que a velocidade da zona-frontal-1 é maior do que 10 km/h, com um fator de crença não menor do que 0.5 e que no máximo é igual a 0.8.

Para utilizar uma regra de decisão é preciso avaliar suas cláusulas de premissa. A avaliação destas cláusulas resulta também num intervalo de crença. No sistema MOTIN emprega-se um modelo de raciocínio inexato que, com base nestes intervalos de crença:

(a) decide se uma cláusula de premissa deve ser considerada satisfeita ou não;
(b) calcula o intervalo de crença numa combinação de cláusulas;
(c) determina os intervalos de crença das cláusulas de ação de uma regra.

Seja então:

(a) $I(x) = [Cr(x) , Pl(x)]$, o intervalo de crença em x.
x é satisfeita $\leftrightarrow$ $Cr(x) \geq CMIN$ e $Pl(x) \geq PMIN$
onde, por "default", CMIN = 0.2 e PMIN = 0.5.

(b) x e y cláusulas de premissa com intervalos de crença I(x) e I(y). Tem-se que:

1) I(x & y) = intervalo de crença da **conjunção** de x e y
= [min(Cr(x) , Cr(y)) , min(Pl(x) , Pl(y))]

2) I(x v y) = intervalo de crença da **disjunção** de x e y
= [max(Cr(x) , Cr(y)) , max(Pl(x) , Pl(y))]

(c) P a premissa de uma regra de decisão R e I(P) o intervalo de crença em P. Seja A uma cláusula de ação de R cujo fator de crença é F. Se P é satisfeita então o intervalo de crença em A é dado por:

$$I(A) = [F * Cr(P) , 1 - F * (1 - Pl(P))]$$

Por exemplo, sejam as regras de decisão:

(1) A & (B v C) --0.8--> D

(2) B & D --0.5--> E

e os intervalos de crença:

I(A) = [0.4 , 0.8]

I(B) = [0.3 , 0.4]

I(C) = [0.6 , 0.7]

Neste caso, para a regra (1) tem-se que:

I(A & (B v C)) = [min(0.4, max(0.3,0.6)), min(0.8, max(0.4,0.7))]
= [0.4 , 0.7]

Logo, admitindo os valores "default" para CMIN e PMIN, como A & (B v C) é satisfeita, pode-se concluir D com intervalo de crença:

I(D) = [0.8 * 0.4 , 1 - 0.8 * (1 - 0.7)]
= [0.32 , 0.76]

Para a regra (2) tem-se que:

I(B & D) = [min(0.3,0.32), min(0.4,0.76)]
= [0.3 , 0.4]

e neste caso, não se pode concluir E porque (B & D) não é satisfeita.

Deve-se observar que o emprego da **plausibilidade**, além da credibilidade, para verificar se uma cláusula é satisfeita ou não, irá limitar a aplicação de regras de decisão, orientando o raciocínio na

direção das evidências para as quais **pode-se** associar altos graus de crença, descartando as evidências que mesmo críveis são pouco plausíveis. De qualquer forma, como o usuário pode alterar os valores da credibilidade e plausibilidade mínimas, no caso de isto não ser desejado, pode-se fazer, por exemplo, PMIN = CMIN, e recair no modelo de raciocínio empregado em sistemas como o sistema DIAGNÓSTICO.

4.4 - PROPAGAÇÃO DE CRENÇA NO SISTEMA MOTIN

Na tentativa de evitar a complexidade exponencial do algoritmo de combinação de evidência de Dempster-Shafer e ao mesmo tempo permitir que as classes de valores para um atributo de um objeto do domínio possam ser estruturadas segundo as necessidades de uma aplicação, adota-se no sistema MOTIN uma aproximação da teoria descrita na Seção 4.2, que consiste em manter crenças apenas aos subconjuntos de θ que o especialista definir como de interesse semântico e aos seus complementos. Com isso a complexidade do processo irá depender da estruturação dos valores, sendo tanto maior quanto maior for o número de classes de interesse.

Em outras palavras: seja θ o conjunto de valores para um atributo de um objeto e V um subconjunto de $P(\theta)$ definido pela rede semântica que estrutura o conhecimento do domínio de aplicação (admite-se que $\theta \in V$, para qualquer domínio de aplicação). Seja V' = { v / -v pertence a V } . Na aproximação adotada mantem-se crenças nas classes de V U V' ao invés de nas classes de $P(\theta)$.

Considerando a Figura 4.1, por exemplo, seriam mantidas crenças nas classes: A, C, Q, T, CA, CQT e θ (classes que aparecem na rede) e em seus complementos: AQT, CQT, ACT, ACQ, QT, A e ϕ, respectivamente. Isto é, ao invés de considerar $2^4 = 16$ classes, seriam consideradas apenas 12 classes. Espera-se que essa diferença seja tanto mais significativa quanto maior for o número de elementos de (observar que se $\#(\theta) \leq 3$, essa aproximação resulta exata,

qualquer que seja a estruturação de classes definida pelo especialista).

Deve-se observar que como os elementos de V constituem as únicas classes de interesse, as regras de decisão irão referenciar (e atribuir crença) apenas a essas classes. Atribui-se crença a um elemento -v de V' quando faz-se referência a NÃO-v ($v \varepsilon V$).

Dentro dessa aproximação, considerando um conjunto qualquer, tem-se inicialmente uma função de atribuição de crença f tal que:

$f(\phi) = 0$; $f(\theta) = 1$ e $f(x) = 0$ para os demais $x \varepsilon V \cup V'$.

À medida que se dispõe de novas evidências sobre elementos de $V \cup V'$, o que pode ser representado por uma função de atribuição de crença f', atualiza-se a crença no conjunto $V \cup V'$ calculando a função de atribuição de crença total $f (+) f'$ através da regra de combinação de evidências de Dempster-Shafer. O problema é que neste processo, devido às interseções entre classes, podem surgir elementos x que não pertencem a $V \cup V'$. Para estes casos, a crença em x é transferida para os menores elementos de $V \cup V'$ que contêm x. Como dentro da teoria de Dempster-Shafer, a crença numa classe x implica crença numa superclasse de x, essa transferência, segundo observam Gordon e Shortliffe (1985), parece ser razoável.

Deve-se observar que no enfoque de Gordon e Shortliffe, como as classes mantêm uma estrutura de árvore, há sempre apenas uma classe para a qual a crença numa classe x que não pertence à árvore, deve ser transferida. Na aproximação proposta pode haver mais de uma classe para a qual a crença em x deve ser transferida. No enfoque de Gordon e Shortliffe no entanto, como nem sempre é possível calcular a crença no complemento de uma classe que pertence à árvore, perde-se a noção de intervalo de crença. No esquema proposto esta noção continua, uma vez que mantêm-se crença nas classes de $V \cup V'$.

Seja por exemplo, θ = { A; B, C, D } e as classes de interesse como mostra a Figura 4.2. Neste caso:

V U V' = {A, B, C, D, AB, AC, BD, CD, ABC, ABD, ACD, BCD, θ }

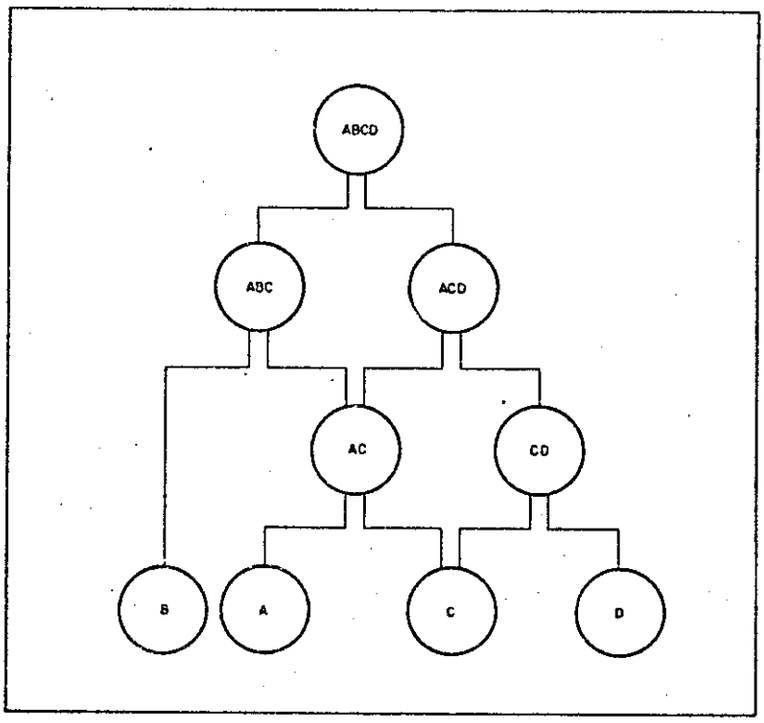

Fig. 4.2 - Classes de valores

Imagine que a crença atual nas classes de V U V' é dada por uma função de atribuição de crença f qualquer. Considere então que uma regra conclui NÃO-B com um intervalo de crença [x,y], o que pode ser representado pela função de atribuição de crença f' dada por:

$f'(ACD) = x$;

$f'(B) = 1 - y$;

$f'(\theta) = 1 - x - (1 - y) = y - x$;

e $f'(x) = 0$ para as demais classes.

Para propagar as conseqüências dessa nova conclusão deve-se calcular a soma ortogonal de f e f', ou seja:

f' \ f	A	B	C	D	AB	AC	BD	CD	ABC	ABD	ACD	BCD	θ
ACD	A	ϕ	C	D	A	AC	D	CD	AC	**AD**	ACD	CD	ACD
B	ϕ	B	ϕ	ϕ	B	ϕ	B	ϕ	B	B	ϕ	B	B
θ	A	B	C	D	AB	AC	BD	CD	ABC	ABD	ACD	BCD	θ

Neste caso deve-se transferir a crença em AD (notar que AD não pertence a V U V') para as menores classes de V U V' que contêm AD: ABD e ACD.

Mais formalmente, seja:

N - classe que não pertence a V U V' e que surgiu devido ao cálculo da soma ortogonal.

A crença em N deve ser repartida entre os elementos de R, onde R é o conjunto obtido pelo Algoritmo 4.1.

ALGORITMO 4.1

(1) Fazer: R = θ.

(2) Determinar:

$S = \{x \in V \cup V' \;/\; N \trianglelefteq x \text{ e } \exists\, y \in R \text{ tal que } x \trianglelefteq y\}$.

(3) Se S = ϕ então retornar R.
Caso contrário, fazer R = S e voltar ao passo (2).

Portanto, sejam:

V - conjunto de classes de valores para um atributo de um objeto cujo domínio é θ, definido pelo especialista como de interesse semântico;

V' - complemento de V em relação a θ;

f - função de crença atual para as classes de V U V';

x - uma conclusão obtida, com intervalo de crença [A,B], pela aplicação de uma regra de decisão;

f' - função de atribuição de crença dada por:

$f'(x) = A$;

$f'(-x) = 1 - B$;

$f'(\theta) = 1 - A - (1 - B) = B - A$;

e $f'(x) = 0$ para os demais $x \in P(\theta)$.

A crença em x é **propagada** no conjunto V U V' através do Algoritmo 4.2:

ALGORITMO 4.2

(1) Determinar $g(v) = f(v) * f'(\theta)$; $v \in V \cup V'$.

(2) Para todo $v \in V \cup V'$, fazer:

(a) Determinar $i = v \cap x$;

(b) Se $i \in V \cup V'$ então fazer:

$g(i) = g(i) + f(v) * f'(x)$.

Caso contrário:

- calcular o conjunto R pelo Algoritmo 4.1;
- repartir a crença em i entre os elementos de R, ou seja:

$$g(r) = g(r) + \frac{f(v) * f'(x)}{\sharp(R)}, \quad r \in R$$

(c) Repetir os passos (a) e (b) acima, considerando, ao invés de x e f'(x), -x e f'(-x);

(3) Determinar $g(v) = \frac{g(v)}{1 - g(\phi)}$ para todo $v \in V \cup V'$.

(4) Estabelecer como nova função de crença total em V U V', a função f dada por:

$f(\phi) = 0$

$f(x) = g(x)$ para todo $x \in V \cup V'$, $x \neq \phi$.

4.5 - EVIDÊNCIAS DE ATRIBUTOS NUMÉRICOS

A combinação de evidências pela aproximação à teoria de Dempster-Shafer discutida na Seção 4.4 exige que o domínio de valores de um atributo seja finito. No sistema MOTIN, no entanto, existe a

possibilidade de se utilizar atributos numéricos cujos domínios são definidos por intervalos de números reais. Para esses atributos portanto, o tratamento é especial.

A manutenção de crenças nos valores possíveis de um atributo de um objeto (o que é feito pelo Algoritmo 4.2) é importante para que se possa avaliar condições de premissa de regras de decisão. Por exemplo, se numa regra é preciso avaliar uma cláusula como FORMA(X) = NÃO-POLIGONAL, recupera-se o intervalo de crença associado à classe NÃO-POLIGONAL e relativo à FORMA(X) e faz-se a avaliação com base neste intervalo (conforme o modelo descrito na Seção 4.3).

No caso de atributos numéricos, apesar de não ser factível manter crenças em todas as "classes possíveis", pode-se ainda utilizar o esquema proposto para tratamento de incerteza, considerando-se "interseções" de intervalos de valores. Seja por exemplo, X um objeto e A um atributo de X que assume valores no intervalo [0,50], ou seja, para este atributo θ = [0,50]. Tem-se inicialmente uma função de atribuição de crença f tal que:

$f(\phi) = 0$; $f([0,50]) = 1$; $f(x) = 0$ para os demais valores.

Se uma regra conclui que $A(X) \geq 20$, com intervalo de crença [0.8, 1], pode-se escrever uma função de atribuição de crença:

$$f'([20,50]) = 0.8 \ ; \ f'([0,50]) = 0.2$$

que passa a ser portanto a função de atribuição de crença total para os valores de A(X) (notar que como f é a **função de atribuição inicial de crença,** f (+) f' = f'). Considere então que uma nova conclusão estabelece que $A(X) \leq 30$, com um intervalo de crença [0.7, 0.8], ou seja:

$f''([0,30]) = 0.7$;

$f''((30,50]) = 1 - 0.8 = 0.2$;

$f''([0,50]) = 1 - 0.7 - 0.2 = 0.1$

Para determinar a crença total, dada essa nova evidência, é preciso, como antes, calcular a soma ortogonal de f' e f", ou seja:

f" \ f'	(0.8)	(0.2)
	[20,50]	[0,50]
(0.7) [0,30]	[20,30] (0.56)	[0,30] (0.14)
(0.2) (30,50]	(30,50] (0.16)	(30,50] (0.04)
(0.1) [0,50]	[20,50] (0.08)	[0,50] (0.02)

isto é, a crença atual em A(X) é dada por g = f' (+) f", com:

g([20,30]) = 0.56
g((30,50]) = 0.20
g([0,30]) = 0.14
g([20,50]) = 0.08
g([0,50]) = 0.02

A obtenção do intervalo de crença resultante da avaliação de uma cláusula de premissa com atributo numérico não é tão direta como no caso dos atributos simbólicos para os quais, mantém-se crença nas classes de valores e em seus complementos.

Seja por exemplo, X um objeto e IDADE um atributo numérico de X assumindo valores em [0,100] e seja f1 a função de atribuição de crença atual em IDADE(X):

IDADE(X)	f1
[0,30]	0.5
[35,50]	0.3
[0,100]	0.2

ou seja: acredita-se com fator de crença 0.5 que a IDADE(X) pertence a [0,30], com fator 0.3 que a IDADE(X) pertence a [35,50], e assim sucessivamente.

Imagine que se quer saber se IDADE(X) ≦ 40, ou equivalentemente, se IDADE(X) pertence a [0,40]. Para a avaliação dessa cláusula deve-se observar que:

a) se IDADE(X) pertence a [0,30] então **necessariamente** IDADE(X) pertence a [0,40], ou seja, se IDADE(X) pertence a [0,30] existe **credibilidade** em IDADE(X) pertencer a [0,40];

b) se IDADE(X) pertence a [0,100] então **pode ser que** IDADE(X) pertença a [0,40], ou seja, é **plausível** que IDADE(X) pertença a [0,40].

Portanto, considerando os fatores de crença nos intervalos individuais pode-se escrever:

$$Cr(IDADE\ (X) \leq 40) = 0.5$$

$$Pl(IDADE\ (X) \leq 40) = 0.5 + 0.3 + 0.2 = 1$$

ou seja:

$$I\ (IDADE\ (X) \leq 40) = [0.5\ ,\ 1]$$

Portanto a avaliação de uma cláusula:

atributo (objeto) função valor

na qual atributo é um atributo numérico, pode sempre ser colocada como um problema do tipo:

" atributo (objeto) $\in$ [a,b] ? "

onde [a,b] é determinado considerando-se função e o domínio de atributo de objeto.

Neste caso, para a determinação do intervalo de crença I = [Cr, Pl] considerando f como a função de atual de atribuição de crença em atributo (objeto), tem-se que:

$$Cr = \Sigma\ f(x)\ ,\quad x = [x1,x2],\quad x \subseteq [a,b]$$

$$Pl = \Sigma\ f(x)\ ,\quad x = [x1,x2],\quad x \cap [a,b] \neq \phi$$

onde:

$[x1,x2] \subseteq [a,b] \quad \leftrightarrow \quad x1 \geq a \ \text{ e } \ x2 \geq b$

$[x1,x2] \cap [a,b] \neq \phi \quad \leftrightarrow \quad \exists\ y \in [x1,x2]$ tal que $y \in [a,b]$.

Para cláusulas que envolvem **relações entre valores** isso nem sempre vale. A avaliação dessas cláusulas portanto, deverá levar em conta a relação em si. Seja por exemplo, Y um outro objeto tal que f2 representa a função de atribuição de crença atual em IDADE(Y):

IDADE(Y)	f2
[10,25]	0.8
[0,100]	0.2

Considere que é preciso avaliar se "X é mais velho do que Y", ou seja, se IDADE(X) > IDADE(Y). Para isso deve-se observar que:

a) se IDADE(X) pertence a [35,50] e IDADE(Y) pertence a [10,25] então X é **necessariamente** mais velho do que Y, ou seja, existe **credibilidade** em "X é mais velho do que Y";
b) se IDADE(X) pertence a [0,30] e IDADE(Y) pertence a [10,25] então existe **possibilidade** de X ser mais velho do que Y, ou seja, é **plausível** que "X é mais velho do que Y".

Considerando os fatores de crença, tem-se:

Cr(IDADE (X) > IDADE (Y)) = 0.3*0.8 = 0.24

Pl(IDADE (X) > IDADE (Y)) = 0.5*0.8 + 0.5*0.2 + 0.3*0.8 +
0.3*0.2 + 0.2*0.8 + 0.2*0.2
= 1

e portanto, para IDADE(X) > IDADE(Y) tem-se o intervalo de crença:

I (IDADE (X) > IDADE (Y)) = [0.24 , 1].

Em casos como esse, isto é para cláusulas de atributos numéricos da forma:

predicado 1 função predicado 2

a determinação do intervalo de crença é feita também pelo Algoritmo 4.2 mas levando em conta para as "interseções", como a função relaciona os dois predicados.

4.6 - RESUMO

Neste capítulo discutiu-se o tratamento de incerteza e a combinação de evidências adotados no sistema MOTIN. Apresentou-se

inicialmente a teoria de Dempster-Shafer, na qual modela-se a informação incerta com base em funções de atribuição de crença. Definiram-se então os conceitos de credibilidade, plausibilidade e intervalo de crença para classes de valores e o mecanismo de soma ortogonal para a combinação de funções de crença.

Mostrou-se a seguir como empregar uma aproximação desta teoria para o tratamento de incerteza em sistemas especialistas. Nesta aproximação faz-se uso da estrutura de classes de valores representada na rede semântica que estrutura o conhecimento do domínio de aplicação e com isso mantém-se crença apenas em classes de interesse, com o conseqüente aumento de eficiência computacional. Mostrou-se que o mesmo mecanismo aplica-se também para valores numéricos, permitindo ter-se em sistemas especialistas o tratamento de incerteza tanto para atributos simbólicos como para atributos que assumem valores em intervalos.

Definiu-se então um modelo de verdade para o raciocínio inexato, que com base em intervalos de crença, permite:

- decidir se uma cláusula de premissa de uma regra de decisão deve ser considerada satisfeita ou não;
- calcular o intervalo de crença numa combinação de cláusulas; e
- determinar, com base no intervalo de crença da premissa e do fator de crença de uma regra, o intervalo de crença de sua conclusão.

CAPÍTULO 5

MECANISMO DE INFERÊNCIA E CONTROLE DE RACIOCÍNIO

5.1 - INTRODUÇÃO

Neste capítulo discute-se o processo de inferência empregado no sistema MOTIN e seu relacionamento com os esquemas de armazenagem de conhecimento e com as estruturas que se dispõe para controle de raciocínio.

Como já se enfatizou anteriormente, a generalidade é um requisito indispensável numa ferramenta para construção de sistemas especialistas. No Capítulo 2 discutiu-se como este requisito é satisfeito no sistema MOTIN, no que diz respeito aos esquemas de representação de conhecimento.

Com relação ao processo de inferência dispõe-se de um mecanismo capaz de orientar o raciocínio de forma progressiva, retroativa ou mista, controlado por estratégias que se baseiam em meta-conhecimento. Com isto o mecanismo de inferência ganha **flexibilidade** pois remove-se parte da informação de controle de sua representação procedimental. Além disso, com a utilização de meta-regras, pode-se prover a semântica de relações e com isso indicar como deve proceder a inferência sobre o conhecimento armazenado em rede associativa. Essas características são importantes porque - à semelhança com os esquemas de representação de conhecimento - permite que o processo de inferência seja adaptadò às características de um domínio específico de aplicação.

O processo de **resolução de problema** empregado no sistema MOTIN está dividido em 4 fases como mostra a Figura 5.1. Nessas fases

empregam-se várias estruturas de controle e de armazenagem de conhecimento que são **destacadas** a seguir.

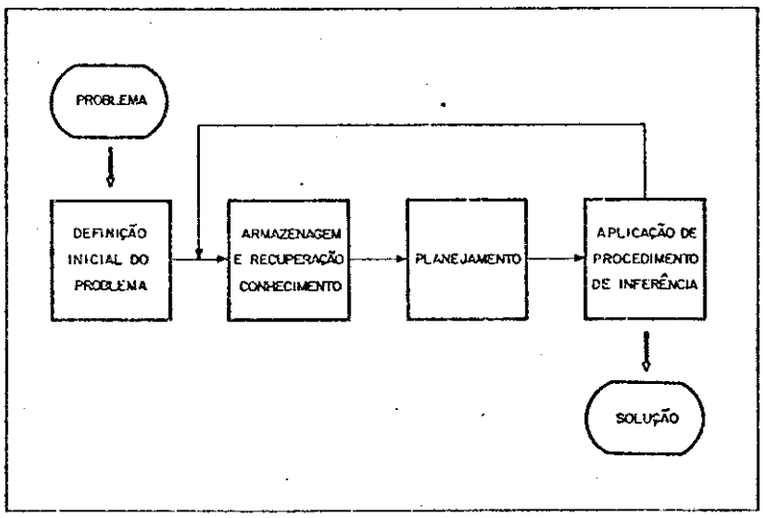

Fig. 5.1 - Paradigma de resolução de problema

DEFINIÇÃO INICIAL DO PROBLEMA - definem-se inicialmente as instâncias possíveis para os objetos do problema assim como seus relacionamentos e introduzem-se os dados previamente conhecidos. Com esta fase portanto, são constituídos o **grafo inicial de instâncias** e o **conjunto inicial de evidências a propagar.** Definem-se também nesta fase, como tarefas, os objetivos que se deseja alcançar.

ARMAZENAGEM E RECUPERAÇÃO DE CONHECIMENTO - recupera-se no **banco de conhecimento** do sistema, as regras de decisão que podem empregar as evidências do **conjunto de evidências a propagar** e atualizam-se a **agenda de tarefas**, a **lista de regras ativas** e o **banco de evidências.**

PLANEJAMENTO - recuperam-se, no **banco de meta-conhecimento,** as meta-regras disponíveis para atribuir prioridades às tarefas da agenda e às regras de decisão referentes ao objetivo atual, ou seja, faz-se o planejamento da solução, descartando regras e tarefas que se mostrarem inúteis e estabelecendo a ordem com que as demais devem ser utilizadas. Estabelecem-se também os contextos de aplicação para as

regras de decisão e atualizam-se o **grafo de instâncias,** além da **lista de regras ativas** e da **agenda de tarefas.**

APLICAÇÃO DE PROCEDIMENTO DE INFERÊNCIA - tenta-se validar a premissa de regras ativas de acordo com o **mecanismo de inferência** estabelecido. Neste processo são atualizadas estruturas de armazenagem de conhecimento como o **conjunto de evidências a propagar,** e estruturas de controle como a **lista de regras ativas** e a **lista de regras desativadas.** Além disso, pode-se, com base em meta-regras, efetuar inferências sobre o conhecimento armazenado na **rede semântica.** Nesta fase, caso os objetivos iniciais tenham sido alcançados, exibem-se as soluções encontradas.

A Figura 5.2 ilustra como essas estruturas se relacionam no sistema MOTIN (as setas indicam fluxo de informação).

Nas seções seguintes discute-se mais detalhadamente o paradigma de resolução de problema empregado pelo sistema MOTIN, assim como seu relacionamento com essas estruturas de controle e de armazenagem de conhecimento.

5.2 - ESTRUTURAS PARA UTILIZAÇÃO DE REGRAS DE DECISÃO

Uma vez adquiridas, as regras de decisão são armazenadas internamente segundo uma representação que economiza espaço, facilita sua utilização e permite implementar alguns mecanismos de controle sobre o processo de inferência que as emprega. As regras assim representadas constituem o **banco de conhecimento** do sistema MOTIN.

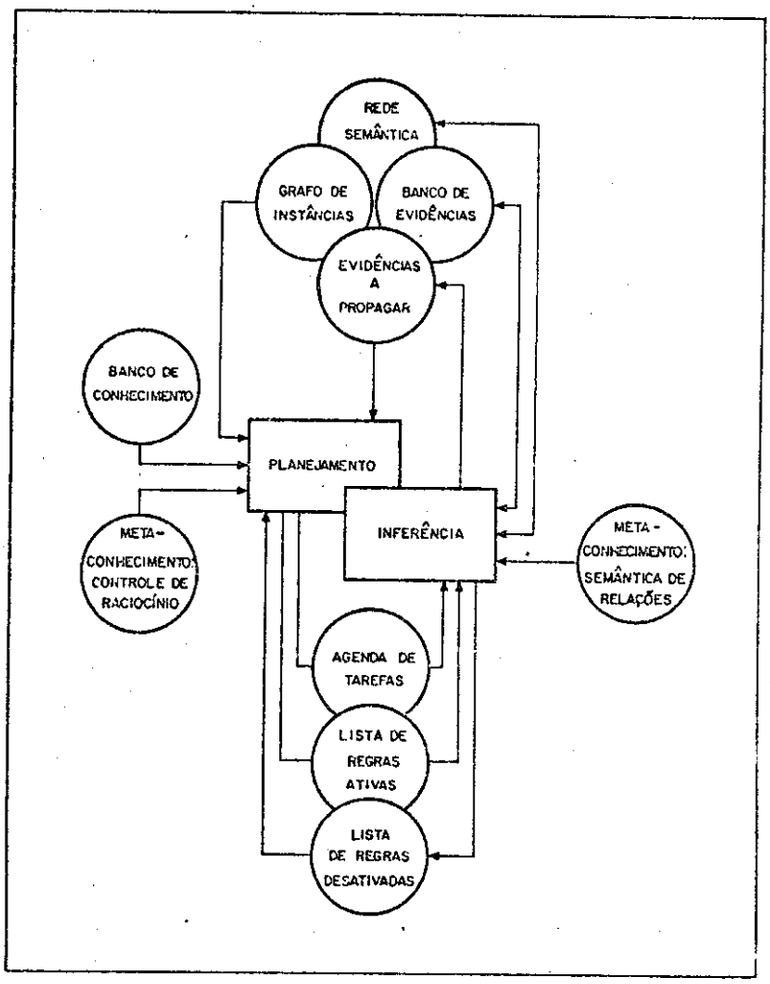

Fig. 5.2 - Arquitetura do sistema MOTIN

Para ilustrar como é formado o banco de conhecimento, considere, por exemplo, as regras:

REGRA R1

SE: 1.
1) LOCAL (SISTEMA-SINÓTICO) = PARANÁ V
2) LOCAL (SISTEMA-SINÓTICO) = SANTA-CATARINA &
2) VELOCIDADE (ZONA-FRONTAL) > 10
ENTÃO: 1) PRECIPITAÇÃO (SJC) = ALTA , 0.7

REGRA R2

SE: 1) LOCAL (SISTEMA-SINÓTICO) = PARANÁ

ENTÃO: 1) ╫INFLUI-EM (SISTEMA-SINÓTICO , SJC)

Neste caso, o banco de conhecimento irá conter:

identificador	valor associado
R1	((& (OU C1 C2) C3) C4 contexto 1)
R2	(C1 C5 contexto 2)
C1	(= P1 PARANÁ)
C2	(= P1 SANTA-CATARINA)
C3	(> P2 10)
C4	(= P3 ALTA)
C5	(∃ INFLUI-EM SISTEMA-SINÓTICO SJC)
P1	(LOCAL SISTEMA-SINÓTICO)
P2	(VELOCIDADE ZONA-FRONTAL)
P3	(PRECIPITAÇÃO SJC)

onde contexto 1 e contexto 2 são os grafos de contexto das regras R1 e R2, respectivamente, conforme discutiu-se no Capítulo 3.

Vê-se portanto, que o banco de conhecimento armazena as regras de decisão numa **forma compacta** referenciando apenas os nomes das cláusulas, que são armazenadas separadamente. Além disto, as próprias cláusulas têm uma representação compacta, com referências a predicados: pares (atributo objeto). Deve-se observar que a representação tanto de cláusulas como de predicados é **única**, ou seja, não existem duas cláusulas ou dois predicados exatamente iguais armazenados no banco de conhecimento.

A representação de cláusulas é feita de uma das seguintes formas:

(a) **cláusula predicativa**

(função predicado com restrições valor)

onde:

função é uma das funções conhecidas:

= , <>, > , < , ≥ , ≤ , ε (pertence) , NAOε ;

predicado com restrições é uma lista da forma:

(predicado [restrições]);

valor é uma constante, uma variável ou uma lista como predicado com restrições.

(b) **cláusula relacional**

(função relação origem destino)

onde:

função é ∃ (EXISTE-RELAÇÃO) ou ∄ (NÃO-EXISTE-RELAÇÃO);

relação é um designador de relação;

origem e destino são ambos objetos ou ambos constantes simbólicas.

(c) **cláusula procedimental**

(FUNCAO procedimento argumento1 ... argumenton)

onde:

FUNCAO é um procedimento que verifica se procedimento está definido e se algum argumentoi corresponde a uma variável que necessita ser instanciada, e então aplica procedimento à lista de argumentos.

Para a utilização de uma regra é preciso, em primeiro lugar, estabelecer os seus contextos de aplicação. Em seguida deve-se avaliar suas cláusulas de premissa e, caso esta avaliação resultar verdadeira, executar suas cláusulas de ação. Para isto atualiza-se a **agenda de tarefas** com as cláusulas a serem avaliadas e a **lista de regras ativas** com elementos que correspondem à forma compacta das regras que estão sob avaliação.

O Algoritmo 5.1 apresenta, mais formalmente, este processo.

ALGORITMO 5.1 - ATIVAR-REGRAS (C)

C é o conjunto de regras de decisão a serem ativadas.

(1) Estabelecer contextos de aplicação para as regras do conjunto C. Seja U = { u1, ..., um } o conjunto de regras de C para as quais existe pelo menos um contexto de aplicação estabelecido.

(2) Recuperar e aplicar as meta-regras que atribuem prioridades às regras do conjunto U. Neste processo, descartam-se as regras consideradas inúteis (UTILIDADE = 0). Seja R = { r1, ..., rn}, com n ≦ m, o conjunto de regras úteis de U tal que ri é mais prioritária que ri+1 (i = 1, ..., n-1).

(3) Para cada ri (i = 1, ..., n), sejam cj (j = 1, ..., ni ; ni ≧ 1), os contextos de aplicação de ri. Para cada i = 1, ..., n fazer:

   (a) Acrescentar à **lista de regras ativas** as ativações de ri correspondentes aos contextos ck (1 ≦ k ≦ ni) que são compatíveis com o contexto atual e acrescentar à **agenda de tarefas**, os elementos que correspondem às cláusulas de premissa de ri;

   (b) Armazenar, para processamento futuro, as demais ativações de ri.

A agenda de tarefas é uma estrutura da forma (T1 ... Tn), onde Ti é tipicamente:
(nome função (predicado [restrições]) instância de objeto valor)
e entendida como uma **questão** a ser respondida ou um **objetivo** a alcançar.

Na avaliação de tarefas correspondentes a cláusulas predicativas, buscam-se no **banco de evidências** os valores de atributo de objeto que satisfazem às restrições. A avaliação é feita então, através da função que irá relacionar as constantes disponíveis.

Por exemplo, para a cláusula C3:

(> P2 10)

onde P2 corresponde ao predicado (VELOCIDADE ZONA-FRONTAL), pode-se ter uma tarefa da forma:

(Tj > (P2) ZONA-FRONTAL-1 10).

Neste caso, buscam-se no banco de evidências os valores conhecidos para **VELOCIDADE** de **ZONA-FRONTAL-1** (notar que numa tarefa tem-se especificados as instâncias correspondentes aos objetos de seus predicados, conforme estabelecido pelo contexto de aplicação). Nesse

caso, como a cláusula não estabelece restrições é possível que se tenha vários valores para VELOCIDADE de ZONA-FRONTAL-1, por exemplo: velocidade da zona-frontal-1 às 8 horas, velocidade da zona-frontal-1 ao meio- dia, ...). Uma vez feito isso, a função "MAIOR" irá verificar se existe algum valor disponível que seja maior do que 10 e irá retornar, como avaliação da cláusula, a crença resultante dessa verificação.

Para as tarefas correspondentes a cláusulas relacionais e procedimentais a avaliação é mais simples. No caso das cláusulas relacionais basta verificar se existe (ou não existe) a relação entre as entidades especificadas. É preciso contudo, levar em conta se a relação é do tipo condicional e se a sua definição ainda não foi avaliada. Para as cláusulas procedimentais basta aplicar o procedimento à sua lista de argumentos.

As tarefas da agenda (ver passo (4) do Algoritmo 5.3) estão sujeitas à ação de meta-regras que poderão **atribuir prioridades** a elas ou então **retirar** da agenda tarefas consideradas inúteis. Considera-se, a qualquer instante, que a tarefa a ser executada é sempre a de maior prioridade.

Uma vez executada uma tarefa verifica-se a ativação de regras, ou seja, atualiza-se, com base no intervalo de crença resultante da avaliação da tarefa, a crença disponível para a premissa das regras ativas e determinam-se quais destas regras devem continuar ativas e quais podem ser desativadas.

Os elementos da **lista de regras ativas** são da forma:

(nome premissa e crença conclusão)

onde premissa e conclusão referenciam nomes de tarefas da agenda para um determinado contexto de aplicação.

Considere, para ilustrar, que a regra R1 é ativada (porque, por exemplo, deseja-se concluir sobre PRECIPITAÇÃO em SJC) para o

contexto C = (SJC-1, ZONA-FRONTAL-1, SISTEMA-SINÓTICO-1). Então, neste caso, atualiza-se a lista de regras ativas com:

(R1 (& [1,1] (OU [0,0] C1 C2) C3) (C4 0.7))

onde [1,1] é o intervalo de crença que se tem inicialmente na premissa de R1 e [0,0] é o intervalo de crença inicial das disjunções. À medida que as cláusulas de premissa são avaliadas, estes intervalos são atualizados. O par (C4 0.7) corresponde à clausula de ação e seu fator de crença associado.

Se, por exemplo, a cláusula C1 é avaliada com intervalo de crença [0.6,1], este elemento da lista de regras ativas sofre, sucessivamente as seguintes transformações:

(a) retira-se a cláusula avaliada e atualiza-se o intervalo de crença da disjunção:

(R1 (& [1,1] (OU [0.6,1] C2) C3) (C4 0.7))

(b) a disjunção ao tornar-se satisfeita (considerando os valores "default" para CMIN e PMIN), pode ser eliminada com atualização do intervalo de crença de R1:

(R1 (& [0.6,1] C3) (C4 0.7))

Neste caso, o intervalo de crença [0.6,1] indica que esta regra deve continuar ativa. Se, em seguida, C3 é avaliada com intervalo de crença [0.8,0.9] tem-se:

(c) retira-se a cláusula avaliada e atualiza-se o intervalo de crença da conjunção:

(R1 (& [0.6,0.9]) (C4 0.7))

(d) como a premissa é considerada satisfeita, toma-se a conclusão C4 com intervalo de crença [0.42,0.93] e retira-se este elemento da lista de regras ativas.

Portanto a todo instante, a lista de regras ativas contêm regras **prontas para serem utilizadas** - regras para as quais existe um

contexto de aplicação determinado - ou regras que já foram **utilizadas parcialmente**, isto é, que tiveram algumas cláusulas de premissa avaliadas como verdadeiras, estando as demais cláusulas ainda por avaliar. Em outras palavras, toda vez que uma cláusula de premissa de um elemento da lista de regras ativas for avaliada, atualiza-se o intervalo de crença, retira-se essa cláusula deste elemento e verifica-se, de acordo com o modelo de verdade adotado, se a regra ainda deve continuar na lista ou se ela deve deixar de estar ativa. Desativa-se uma regra quando a crença em sua premissa cai abaixo de limiares de aceitação ou quando sua premissa já foi completamente avaliada e conseqüentemente sua ação já foi tomada. As regras, quando deixam de estar ativas, são armazenadas na **lista de regras desativadas**, uma estrutura cujos elementos são da forma:

(nome contexto de aplicação).

Com esta lista evita-se ativar uma determinada regra para um mesmo contexto de aplicação, mais de uma vez.

Portanto, de uma maneira geral, uma vez avaliadas **com sucesso** todas as cláusulas da premissa de uma regra, executam-se suas cláusulas de ação, ou seja:

- para as **predicativas**, determina-se a crença na cláusula com base no fator de crença da ação e no intervalo de crença da premissa, segundo o modelo discutido na Seção 4.3 e atualiza-se o banco de evidências (ou o conjunto de evidências a propagar);
- para as **relacionais**, simplesmente representa-se a relação entre as entidades envolvidas na cláusula.

Cláusulas de ação do tipo **procedimental** também são possíveis no sistema MOTIN. Neste caso, os efeitos da execução da cláusula são especificados pelo próprio procedimento. Cláusulas que empregam o procedimento - preestabelecido - MOSTRE, por exemplo, exibem mensagens e valores de constantes ou variáveis.

5.3 - EMPREGO DE META-REGRAS PARA CONTROLE DO RACIOCÍNIO

Como se comentou na seção anterior, a ordem de avaliação das cláusulas que ocorrem nas regras de decisão ativas depende das prioridades que essas cláusulas recebem enquanto permanecem na agenda de tarefas. Comentou-se também que tais prioridades são estabelecidas por meta-regras as quais, em última análise, determinam quais regras ativas devem ser avaliadas e em que ordem.

A própria ativação de regras por outro lado, é também um processo sobre o qual atuam meta-regras. Neste caso as regras **possíveis** de serem usadas são analisadas por meta-regras que verificam se todas elas são realmente úteis e se existem regras que são mais prioritárias do que outras.

Portanto através desses dois controles - ativação de regras e ativação de cláusulas - garante-se que serão buscados apenas os objetivos considerados úteis e que os objetivos considerados mais prioritários serão buscados primeiro. Deve-se observar que dessa forma é possível, em tempo de execução, ordenar as regras e em seguida, ordenar as suas cláusulas, em termos de importância para uma aplicação específica.

Para a codificação de cláusulas de meta-regras empregam-se:
(1) como OBJETOS:

(a) REGRA
(b) META-REGRA
(c) TAREFA

Ou seja, o foco de atenção de meta-regras é um conjunto de regras, de meta-regras ou de tarefas.

(2) como **FUNÇÕES DE PREMISSA:**

(a) OBJETIVO: (X)

verifica se o objetivo atual é X.

(b) EXISTEM (O,M)

NÃO-EXISTEM (O,M)

retorna os objetos do tipo O (regras, meta-regras ou tarefas) que satisfazem M. M é uma referência à função MENCIONAM / NÃO-MENCIONAM, que verifica quais elementos mencionam (ou não mencionam) um determinado texto. Para o caso de regras ou meta-regras, a procura por texto pode ser na premissa ou na conclusão.

(c) CONHECIDO: (X)

DESCONHECIDO: (X)

verifica se X aparece no banco de evidências (CONHECIDO) ou não (DESCONHECIDO).

(d) POSSÍVEL-CONCLUIR: (X)

IMPOSSÍVEL-CONCLUIR: (X)

verifica a existência (ou não-existência) de regras com X na conclusão.

(e) FOI-IMPOSSÍVEL-CONCLUIR: (X)

verifica se X não ocorre no banco de evidências e se existem regras com X na conclusão, na lista de regras desativadas.

Essas funções são usadas especificamente para a codificação de cláusulas de premissa de **meta-regras**. Além dessas, pode-se empregar também qualquer função de cláusula de regras de decisão, como por exemplo, = , ≧ ou EXISTE-RELAÇÃO.

(3) como FUNÇÕES DE AÇÃO

(a) UTILIDADE (X) U

estabelece em U a utilidade dos elementos do conjunto X.

(b) PRIORIDADE (X) P

estabelece P como o nível de prioridade para os elementos do conjunto X.

Para ilustrar, sejam por exemplo, as seguintes meta-regras:

META-REGRA 1

SE: 1) OBJETIVO: PRECIPITAÇÃO (SJC) &
2) LOCAL (ZONA-FRONTAL) = PARANÁ &
3) EXISTEM REGRAS *R1
MENCIONAM "velocidade da zona frontal" PREMISSA &
4) EXISTEM REGRAS *R2
MENCIONAM "intensidade da zona frontal" PREMISSA

ENTÃO: 1) PRIORIDADE (*R1) 1
2) PRIORIDADE (*R2) 0.9

Neste caso a função MENCIONAM recupera dois conjuntos de regras (*R1 e *R2) e a meta-regra estabelece, quando deseja-se deduzir o valor da precipitação em SJC e o local da zona frontal é o Paraná, que as regras do conjunto *R1 (regras que mencionam **velocidade da zona frontal** na premissa) têm maior prioridade que as regras do conjunto *R2 (regras que mencionam **intensidade da zona frontal** na premissa).

META-REGRA 2

SE: 1) OBJETIVO: C-G-E (SJC) &
2) EXISTEM REGRAS *R
MENCIONAM "condições desfavoráveis" PREMISSA

ENTÃO: 1) UTILIDADE (*R) 0

Neste caso a meta-regra estabelece que mesmo que existam regras que concluem sobre o valor das condições de grande escala em SJC, é inútil (UTILIDADE = 0) tentar aplicá-las caso elas mencionem **condições desfavoráveis** em cláusulas de premissa.

Estes dois exemplos mostram meta-regras que se aplicam **especificamente** a um domínio de conhecimento e que devem ser usadas para se buscar **eficiência** no processo de raciocínio, ou seja, para descartar regras que não são muito úteis (evitando assim a utilização **exaustiva** de regras) ou para desenvolver o raciocínio numa ordem que se mostra mais adequada a uma determinada aplicação. Além de meta-regras desse tipo existem as **meta-regras independentes de domínio,**

utilizadas para estabelecer estratégias gerais de raciocínio. Considere-se, por exemplo, a meta-regra a seguir:

META-REGRA 3

SE: 1) FOI-IMPOSSÍVEL-CONCLUIR: *ATR (*OBJ1) &
2) #PARTICULARIZAÇÃO-DE (*OBJ2 , *OBJ1) &
3) EXISTEM REGRAS *R
MENCIONAM "*ATR de *OBJ2" CONCLUSÃO
ENTÃO: 1) UTILIDADE (*R) 1

Esta meta-regra estabelece de uma forma geral, que regras que concluem sobre o valor de um atributo *ATR de um determinado objeto são úteis (UTILIDADE = 1) quando se quer concluir sobre *ATR de uma generalização daquele objeto. Uma meta-regra deste tipo irá permitir, por exemplo, que no caso de não ter sido possível concluir sobre o LOCAL de um SISTEMA-SINÓTICO, que se utilizem, para isso, regras que concluem sobre o LOCAL de ZONA-FRONTAL.

Deve-se observar que a aplicação de meta-regras não se restringe às regras de decisão e às tarefas: é possível a existência de meta-regras que se aplicam a outras meta-regras (notar que META-REGRA é um **objeto** de meta-regras, assim como REGRA e TAREFA). Considere, por exemplo:

(a) meta-regras utilizadas para ordenar conjuntos de regras;
(b) meta-regras que estabelecem a utilidade ou não de conjuntos de regras.

Como não interessa atribuir prioridades a regras inúteis para uma determinada tarefa, deve-se sempre que possível, utilizar as meta-regras do grupo (b) antes das meta-regras do grupo (a). Esta estratégia pode ser estabelecida por uma meta-regra que se aplica a conjuntos de meta-regras:

META-REGRA 4

SE: 1) EXISTEM META-REGRAS *M
MENCIONAM "utilidade" CONCLUSÃO

ENTÃO: 1) PRIORIDADE (*M) 1

As meta-regras portanto, são armazenadas segundo **níveis de aplicação:** nível 0 - meta-regras que se aplicam a terefas; nível 1 - meta-regras que se aplicam a regras de decisão; nível 2 - meta-regras que se aplicam a meta-regras de nível 1; e assim sucessivamente.

Mostra-se a seguir, o algoritmo de utilização de meta-regras para controle de raciocínio.

ALGORITMO 5.2 - APLICAR-METAREGRAS (L , N)

L é um conjunto de tarefas, de regras de decisão (nestes casos, N = 0) ou de meta-regras (N > 0) sobre as quais pretende-se aplicar meta-regras de nível N + 1. Este algoritmo aplica as meta-regras adequadas e retorna o conjunto L resultante, ou seja, o conjunto L que contém somente elementos úteis e em ordem decrescente de prioridade.

(1) Recuperar as meta-regras de nível N + 1 e fazer uma pré-análise de cada uma delas através da qual suas variáveis são instanciadas (algumas meta-regras podem ser descartadas devido à pré-análise).
(2) Seja M o conjunto de meta-regras que resulta do passo anterior. Enquanto houver meta-regras de nível mais alto, empregar recursivamente o algoritmo APLICAR-METAREGRAS com parâmetros (M , N + 1) para verificar a utilidade e a ordem de aplicação das regras do conjunto M.
(3) Aplicar as meta-regras do conjunto M que resulta do passo (2) sobre os elementos de L e retornar o conjunto L resultante.

5.4 - MECANISMO DE INFERÊNCIA DO SISTEMA MOTIN

A realização de inferências no sistema MOTIN ocorre principalmente com a utilização das regras de decisão que compõem o

seu banco de conhecimento. Entretanto algumas inferências podem ser feitas com base no conhecimento armazenado em rede associativa.

Existe no sistema MOTIN, devido ao requisito de generalidade, pouca informação semântica preestabelecida para relações da rede associativa. Basicamente utiliza-se a transitividade das relações PARTE-DE e PARTICULARIZAÇÃO-DE e, no caso de objetos, a possibilidade de transferência de evidências de um determinado objeto para suas particularizações. Em geral, a semântica de relações é estabelecida por meio de meta-regras que atualizam o conhecimento do domínio armazenado na rede associativa e também o banco de evidências.

Seja, por exemplo, a seguinte meta-regra:

META-REGRA 5

SE: 1) PARTE-DE (*X , *Y) &
2) PARTICULARIZACAO-DE (*Z , *Y)
ENTÃO: 1) PARTE-DE (*X , *Z)

Esta meta-regra estabelece de uma forma geral, como inferir novos relacionamentos PARTE-DE. Por exemplo, instanciando-se as variáveis *X, *Y e *Z para MEMÓRIA, COMPUTADOR e NEXUS-1600 respectivamente, pode-se inferir que MEMÓRIA é parte de NEXUS-1600 dado que MEMÓRIA é parte de COMPUTADOR e que NEXUS-1600 é particularização de COMPUTADOR.

Estas inferências são importantes pois permitem avaliar cláusulas relacionais de premissa de regras de decisão, para relações que não ocorrem diretamente na rede e são realizadas de modo progressivo quando uma determinada relação é estabelecida. O objetivo de efetuar estas inferências de forma exclusivamente progressiva é o de evitar consumir muito tempo neste processo - a aplicação de regras de decisão deve ser prioritária à atualização exaustiva da rede semântica - e o de controlar o raciocínio, principalmente para meta-regras recursivas como a META-REGRA 5. É claro que com isto algumas

inferências possíveis podem deixar de ser realizadas (Smith et alii, 1986).

Outro tipo de meta-regra permite transferir evidências de um objeto para outros objetos, sob determinadas condições. Neste caso emprega-se o procedimento preestabelecido HERDAR como cláusula de ação:

HERDAR (objeto1, objeto2 [,atributo1, ...,atributon])

que transfere evidências relativas aos atributos atributo1, ..., atributon relativas ao objeto2, para o objeto1. No caso de se utilizar apenas HERDAR (objeto1, objeto2) transferem-se para objeto1 todas as evidências disponíveis para objeto2.

É inegável que além da flexibilidade que o emprego dessas meta-regras proporciona ao sistema, existe uma economia de representação conseqüente desse tipo de inferência. Existe também, evidentemente, uma perda de eficiência computacional em relação a um mecanismo que mantém seu processo de inferência embutido em procedimentos.

A utilização de regras de decisão, uma vez definido o problema inicialmente (Fase 1 da Figura 5.1), se dá de acordo com o Algoritmo 5.3. Regras que possuem variáveis quantificadas, antes de serem utilizadas são submetidas a uma pré-análise através da qual suas variáveis são instanciadas e suas condições de premissa, modificadas.

Seja, por exemplo, a regra:

SE:
  Q) TODOS (*V3, *V4), (*V3 > *V4)
  1) DISTÂNCIA (SISTEMA-SINÓTICO, TEMPO = *V3) <
       DISTÂNCIA (SISTEMA-SINÓTICO, TEMPO = *V4) &
  2) INTENSIDADE (SISTEMA-SINÓTICO) = POUCA
ENTÃO:
  1) C-G-E (SJC, TEMPO = 24) = FAVORÁVEL (FC = 0.70)

Neste caso, se os instantes de TEMPO a serem considerados são, por exemplo, 12, 18 e 24 horas, tem-se a regra modificada para:

SE:

1) DISTÂNCIA (SISTEMA-SINÓTICO, TEMPO = 18) <
   DISTÂNCIA (SISTEMA-SINÓTICO, TEMPO = 12) &
2) DISTÂNCIA (SISTEMA-SINÓTICO, TEMPO = 24) <
   DISTÂNCIA (SISTEMA-SINÓTICO, TEMPO = 18) &
3) DISTÂNCIA (SISTEMA-SINÓTICO, TEMPO = 24) <
   DISTÂNCIA (SISTEMA-SINÓTICO, TEMPO = 12) &
4) INTENSIDADE (SISTEMA-SINÓTICO) = POUCA

ENTÃO:

1) C-G-E (SJC, TEMPO = 24) = FAVORÁVEL (FC = 0.70)

ALGORITMO 5.3

(1) Se todos os objetivos da sessão foram alcançados, exibir os resultados e terminar a execução.

(2) Seja E o conjunto de evidências a propagar. Se E = ϕ, então ir para o passo (4).

Caso contrário:

(3) Seja E = {e1, ..., en}. Para cada evidência ei (i = 1, ..., n) a propagar:

   (a) recuperar o conjunto P de regras que possuem cláusulas de premissa que podem ser satisfeitas pela evidência ei;
   (b) aplicar o **algoritmo de ativação de regras**, para o conjunto P;
   (c) atualizar o banco de evidências com ei;
   (d) retirar ei do conjunto E.

(4) Recuperar e aplicar as meta-regras que atribuem prioridades às tarefas da agenda. Neste processo, descartam-se as tarefas consideradas inúteis.

(5) Seja T o conjunto que representa a agenda de tarefas. Se T = ϕ então exibir os resultados disponíveis para os objetivos iniciais e terminar a execução.

Caso contrário:

(6) Seja T = {t1 , ... , tk } tal que ti tem maior prioridade que ti+1 (i = 1, ..., k-1). Retirar t1 de T e verificar, com base no banco atual de evidências, se existe alguma evidência que se refere a t1, ou seja, evidência que satisfaz ou que rejeita t1. Se não existem tais evidências, então ir para o passo (9).

Caso contrário:

(7) Se t1 não é satisfeita então, acrescentar à lista de regras desativadas as regras ativas que possuírem cláusulas de premissa correspondentes a t1 e ir para o passo (5).

Caso contrário:

(8) Neste caso, t1 é satisfeita. Portanto, atualizar a lista de regras ativas, isto é, eliminar dos elementos da lista, as cláusulas correspondentes a t1. Se a premissa de algum elemento da lista de regras ativas tornar-se completamente verificada, então acrescentar suas conclusões ao conjunto de evidências a propagar e ir para o passo (1).

Caso contrário, ou seja, se todas as regras continuarem ativas, então ir para o passo (5).

(9) Se a solução da tarefa t1 deve ser perguntada ao usuário, fazer a pergunta e colocar a resposta fornecida no conjunto de evidências a propagar e ir para o passo (1).

Caso contrário:

(10) Seja C o conjunto de regras que concluem sobre t1. Aplicar o algoritmo de ativação de regras para o conjunto C e ir para o passo (4).

Com o emprego do Algoritmo 5.3, o encadeamento de regras é feito ora progressiva, ora retroativamente. Este algoritmo no entanto, poderá ser particularizado caso o usuário desejar desenvolver uma estratégia de raciocínio **puramente progressiva** ou **puramente retroativa**. Para o raciocínio progressivo, basta que não se permita a execução do passo (10) do algoritmo, o que pode ser conseguido de duas maneiras:

- descartando tarefas para as quais não é possível verificar, com base no banco atual de evidências, se estão satisfeitas ou não;

ou

- entrando com novos dados de forma que essas tarefas possam ser avaliadas (passo (9) do algoritmo).

Com relação ao raciocínio retroativo, basta manter o conjunto de evidências a propagar sempre vazio, isto é, tanto o conjunto inicial de evidências a propagar, quanto as conclusões de regras (passo (8) do algoritmo) são utilizadas somente para atualizar o banco de evidências.

Além disso, com estas estruturas de controle de raciocínio, principalmente a lista de regras ativas e a agenda de tarefas, é fácil incorporar heurísticas que agem sobre a lista de regras ativas e a agenda de tarefas, para implementar estratégias de resolução de conflito, como por exemplo:

- **princípio da generalidade**: dar maior prioridade, em qualquer instante da execução do mecanismo de inferência, às regras com as menores premissas, ou seja, regras que podem ser mais facilmente avaliadas. O contrário também é possível e corresponde ao **princípio da especificidade**: dar prioridade às regras mais específicas, ou seja, às regras que possuem premissas com o maior número de cláusulas.
- **princípio da recentidade**: dar maior prioridade às regras ativas que tiveram cláusulas de premissa avaliadas mais recentemente.

5.5 - SESSÃO DE CONSULTA AO SISTEMA MOTIN

Apresenta-se a seguir uma sessão de consulta ao sistema MOTIN, considerando o banco de regras estabelecido para o domínio de Meteorologia (Apêndice B). A exemplo da Seção 2.5, as linhas sublinhadas são comentários que não pertencem à sessão.

MODULO DE CONSULTA

Inicializa sessao
Sair do modulo de consulta

Banco de conhecimento:
> previsao

DEFINICAO INICIAL DO PROBLEMA

Deseja estabelecer instancias para objetos do problema?
> s

O especialista informa ao sistema os objetos conhecidos para o problema atual. Constrói-se então o grafo inicial de instâncias.

Objeto:
> sjc
Numero de instancias:
> 1
Objeto:
> sistema-sinotico
Numero de instancias:
> 1
Objeto:
> cavado
Numero de instancias:
> 1
Objeto:
> vortice
Numero de instancias:
> 1
Objeto:
> jato
Numero de instancias:
> 1
Objeto:
>

Deseja estabelecer relacoes entre estas instancias?
> s

AQUISICAO DE NOVA RELACAO

> presenca-de (jato-1, sjc-1)

Existe mais algum SJC?
> n

Existe mais algum JATO?
> n

AQUISICAO DE NOVA RELACAO

Alguma evidencia conhecida?
> s

São fornecidos os dados previamente conhecidos que constituirão o conjunto inicial de evidências a propagar.

Evidencia 1:
> carta-isalobarica (sjc-1) = -6
Evidencia 2:
> distancia (sistema-sinotico-1, tempo = 0) = 500
Evidencia 3:
> distancia (sistema-sinotico-1, tempo = -12) = 800
Evidencia 4:
> distancia (cavado-1, tempo = 0) = 450
Evidencia 5:
> distancia (cavado-1, tempo = -12) = 900
Evidencia 6:
> tamanho (cavado-1, tempo = 0) = 200

```
Evidencia 7:
> tamanho (cavado-1, tempo = -12) = 100
Evidencia 8:
> distancia (vortice-1, tempo = 0) = 400
Evidencia 9:
> distancia (vortice-1, tempo = -12) = 900
Evidencia 10:
>
```

OBJETIVOS DA CONSULTA

Definem-se os objetivos que se deseja alcançar.

```
Objetivo 1:
> precipitacao (sjc, tempo = 24)
Objetivo 2:
>
```

SESSAO DE CONSULTA

O sistema inicialmente, verifica a existência de relações entre as instâncias de objetos para estabelecer os contextos de aplicação das regras de decisão.

```
EXISTENCIA-DE (VORTICE-1, SJC-1)?
> s

Existe mais algum VORTICE
tal que:
    EXISTENCIA-DE (VORTICE,SJC-1)?
> n

EXISTENCIA-DE (CAVADO-1, SJC-1)?
> s

Existe mais algum CAVADO
```

```
tal que:
    EXISTENCIA-DE (CAVADO,SJC-1)?
> n

ASSOCIADO-A (SISTEMA-SINOTICO-1, SJC-1)?
> s

Existe mais algum SISTEMA-SINOTICO
tal que:
    ASSOCIADO-A (SISTEMA-SINOTICO,SJC-1)?
> n

EXISTENCIA-DE (JATO-1, SJC-1)?
> s

Existe mais algum JATO
tal que:
    EXISTENCIA-DE (JATO,SJC-1)
dado que:
    ASSOCIADO-A (SISTEMA-SINOTICO-1,SJC-1)?
> n

Existe algum UMIDADE
tal que:
    EXISTENCIA-DE (UMIDADE,SJC-1)
dado que:
    ASSOCIADO-A (SISTEMA-SINOTICO-1,SJC-1)
    AGE-SOBRE (JATO-1,SJC-1)?
> s
----- UMIDADE-1 -----

Existe mais algum UMIDADE
tal que:
    EXISTENCIA-DE (UMIDADE,SJC-1)
dado que:
    ASSOCIADO-A (SISTEMA-SINOTICO-1,SJC-1)
```

```
    AGE-SOBRE (JATO-1,SJC-1)?
> n
```

Neste ponto algumas regras já têm seus contextos estabelecidos e podem ser usadas.

```
DBU (SJC-1)?
> 4
>

INTENSIDADE (SISTEMA-SINOTICO-1)?
> pouca
>
```

Entretanto, para outras, ainda é preciso estabelecer os contextos de aplicação.

```
Existe algum ZONA-FRONTAL
tal que:
    PARTICULARIZACAO-DE (ZONA-FRONTAL,SISTEMA-SINOTICO-1)
dado que:
    ASSOCIADO-A (SISTEMA-SINOTICO-1,SJC-1)?
> s
----- ZONA-FRONTAL-1 -----

Existe mais algum ZONA-FRONTAL
tal que:
    PARTICULARIZACAO-DE (ZONA-FRONTAL,SISTEMA-SINOTICO-1)
dado que:
    ASSOCIADO-A (SISTEMA-SINOTICO-1,SJC-1)?
> n

Existe algum BLOQUEIO
tal que:
    EXISTENCIA-DE (BLOQUEIO,SJC-1)
dado que:
```

```
    ASSOCIADO-A (SISTEMA-SINOTICO-1,SJC-1)?
> s
----- BLOQUEIO-1 -----

Existe mais algum BLOQUEIO
tal que:
    EXISTENCIA-DE (BLOQUEIO,SJC-1)
dado que:
    ASSOCIADO-A (SISTEMA-SINOTICO-1,SJC-1)?
> n
```

Valores de predicados podem ser acompanhados por um fator de crença.

```
LOCAL (ZONA-FRONTAL-1)?
> parana .9
> sao-paulo .8
>

ATIVA (ZONA-FRONTAL-1)?
> ?
```

Neste caso o usuário deseja saber quais são as respostas possíveis.

```
Dominio de valores para ATIVA (ZONA-FRONTAL-1):
{SIM, NAO}

> sim
>

VELOCIDADE (ZONA-FRONTAL-1)?
> 10
>

ATIVIDADE (ZONA-FRONTAL-1)?
> desenvolvendo-se
>
```

```
LOCAL (SISTEMA-SINOTICO-1)?
> sao-paulo
>

MES (SJC-1)?
> 10
>
```

RESULTADOS OBTIDOS

```
Deseja ver os resultados obtidos para:
PRECIPITACAO (SJC)
dado que
   TEMPO = 24?
> s

Deseja ver crencas acumuladas?
> n


PRECIPITACAO (SJC-1)
dado que:
  TEMPO = 24

       CRENCA    VALOR
      ------------------------
        0.63     {SIM, NAO}
        0.36     SIM

Aperte uma tecla para continuar ...

  ...
```

A sessão ainda continua, caso existam na agenda tarefas a serem avaliadas.

FIM DE SESSAO DE CONSULTA

Ao final de uma sessão, o usuário pode salvar o banco atual de evidências para futuro processamento.

Deseja salvar os dados e resultados desta sessao?
> n

MODULO DE CONSULTA

Inicializa sessao
Sair do modulo de consulta

...

5.6 - RESUMO

Neste capítulo discutiu-se o processo de inferência empregado no sistema MOTIN e seu relacionamento com os esquemas de armazenagem de conhecimento e com as estruturas de controle de raciocínio.

Foram discutidas, em primeiro lugar, as estruturas para utilização de regras de decisão. Mostrou-se então a forma com que as regras estão armazenadas no banco de conhecimento do sistema e o processo de ativação de regras, do qual resultam a **lista de regras ativas** e a **agenda de tarefas.** Mostrou-se também como este processo está sujeito à ação de meta-regras que atribuem níveis de utilidade e de prioridade tanto às regras como às tarefas.

A seguir discutiu-se o mecanismo de inferência do sistema MOTIN, primeiro mostrando como o conhecimento armazenado em rede semântica pode ser atualizado no curso de uma sessão de consulta e por

último, o algoritmo de resolução de problema que, de forma inovativa, emprega as regras de decisão ativas tanto progressiva como retroativamente. Mostrou-se também uma sessão de consulta ao sistema, para ilustrar as capacidades de raciocínio do sistema.

Procurou-se também chamar a atenção neste capítulo para a flexibilidade dos processos de inferência. Mostrou-se que o sistema MOTIN dispõe de um mecanismo capaz de orientar o raciocínio de forma progressiva, retroativa ou mista, controlado por estratégias que se baseiam em meta-conhecimento. Esta característica de flexibilidade - como se enfatizou para os esquemas de representação de conhecimento - é importante em ferramentas para construção de sistemas especialistas uma vez que permite uma adequação do processo de inferência a domínios específicos, expandindo suas fronteiras de aplicação.

CAPÍTULO 6

CONCLUSÃO

6.1 - COMPARAÇÃO COM OUTRAS ABORDAGENS

Existem atualmente muitas ferramentas para a construção de sistemas especialistas e é comum a ocorrência de características semelhantes entre elas. O propósito desta seção é situar o sistema MOTIN dentre algumas ferramentas que têm recebido mais destaque na literatura - porque originaram-se de sistemas especialistas famosos, porque têm sido muito utilizadas para construir sistemas práticos ou porque constituem abordagens importantes para a organização de um sistema especialista - comparando suas características ao longo das seguintes dimensões: bases de conhecimento e de dados, estrutura de controle e tratamento ao raciocínio inexato. Waterman (1986) apresenta referências bibliográficas e uma visão geral dos sistemas citados nesta seção. Encontra-se também em Waterman e Hayes-Roth (1983) uma discussão sobre características de ferramentas para construção de sistemas especialistas.

Inicialmente deve-se ressaltar que a proposta do sistema MOTIN é que características de generalidade são indispensáveis numa ferramenta. Dessa forma pode-se imaginar o sistema DIAGNÓSTICO como uma de suas "instâncias", na qual particulariza-se a forma de expressar o conhecimento para uma linguagem de quádruplas, a organização dos objetos para uma hierarquia estrita e a estratégia de raciocínio para o encadeamento retroativo e exaustivo de regras. Mostrou-se também no Capítulo 4, que o tratamento de incerteza e a regra de Bernoulli para combinação de evidências utilizados no sistema DIAGNÓSTICO, são casos particulares do esquema proposto. Além disso, características do EMYCIN (van Melle et alii, 1985) não implementadas

no DIAGNÓSTICO, como o emprego de meta-regras para controle do raciocínio, ocorrem no sistema MOTIN.

Pode-se argumentar sobre a validade de se ter ferramentas muito flexíveis para a construção de sistemas especialistas, especialmente no que se refere à estrutura de controle, uma vez que isto compromete a eficiência de um sistema prático. Defende-se aqui, como Fickas e Novick (1985) o fazem, que a seleção de qual parcela do conhecimento de resolução de problema usar a cada instante é um processo complexo e que requer conhecimento específico do domínio para ser efetivo e que a definição do conhecimento de controle pode ser tão difícil quanto definir o próprio conhecimento do domínio. A arquitetura proposta, portanto, com sua estrutura de controle baseada em meta-conhecimento, leva em conta essas questões, uma vez que mantém "embutido" no sistema apenas um mecanismo para tratamento de meta-regras, transferindo para o especialista (e conseqüentemente, para a aplicação), a definição das decisões de controle. Além disso, o emprego de um esquema declarativo como o de meta-regra, facilita a construção, depuração e manutenção do conhecimento estratégico. Advoga-se que para uma determinada aplicação, pode-se conseguir eficiência transformando em procedimentos o conhecimento de controle estabelecido e depurado, uma vez entendidos os princípios de resolução de problema do domínio em questão.

Existem pontos em comum entre a abordagem proposta e o sistema KAS, uma ferramenta desenvolvida a partir do sistema PROSPECTOR (Duda et alii, 1977), tais como a utilização de rede semântica como uma generalização de triplas objeto-atributo-valor e a utilização de regras de decisão com o emprego de encadeamentos progressivo e retroativo. Existe também no sistema KAS, a utilização de heurísticas para a seleção de regras e cláusulas mais promissoras. Neste sistema contudo tais heurísticas são preestabelecidas, o que torna o sistema menos flexível, embora mais eficiente. O tratamento de incerteza no sistema KAS é mais rígido e baseia-se em probabilidades.

A separação entre o conhecimento melhor estruturado do domínio e o conhecimento empírico é um ponto em comum também com o sistema KEE (Fikes and Kehler, 1985) no qual empregam-se os esquemas de representação por "frames" (ou rede semântica com hierarquia de partes e particularizações) e regras. Outra semelhança com o sistema KEE é a possibilidade de encadear regras progressiva ou retroativamente. O sistema MOTIN contudo, inova quando permite que estas duas estratégias se combinem num único ciclo de execução. Em ambos os sistemas é possível utilizar representação procedimental escrevendo-se programas LISP; no sistema KEE entretanto, existe também a possibilidade de programação orientada por objetos.

O sistema MOTIN apresenta também alguns pontos em comum com o OPS5 (Brownston et alii, 1985), apesar deste último ser uma linguagem de representação (baseada em regras) de propósito geral e não uma ferramenta do tipo esqueleto. O esquema "aberto" de representação de cláusulas, por exemplo, ocorre também no OPS5, no qual pode-se representar uma cláusula com um número arbitrário de relações entre símbolos, que correspondem a pares atributo-valor associados a um objeto. Com relação ao mecanismo de controle, tem-se no OPS5 um esquema mais simples, do tipo progressivo. Não existe tratamento de incerteza no OPS5, embora como uma linguagem de propósito geral, seja possível programá-lo.

Existem também algumas semelhanças do banco de evidências do sistema MOTIN com a estrutura de "blackboard" do HEARSAY-III. Em ambos tem-se informação estruturada e acessível globalmente. Além disso, devido à indexação com atributos, as fontes de conhecimento do sistema MOTIN (regras de decisão) também reagem a alterações produzidas na base de dados globais, durante o processo de resolução de problema. É possível também, como no HEARSAY-III, ter-se várias linhas de raciocínio, devido ao mecanismo de seleção de tarefas. A estrutura de escalonamento do sistema MOTIN, no entanto, baseada na lista de regras ativas e na agenda de tarefas, é mais simples.

Com relação à abordagem de generalização proposta pelo sistema AGE (Nii and Aiello, 1979), uma ferramenta também do tipo esqueleto, existem poucas semelhanças. Com o sistema MOTIN pretende-se ter características de generalidade por meio de estruturas de armazenamento flexíveis e com um interpretador baseado em meta-conhecimento que possui poucas decisões preestabelecidas. O enfoque do sistema AGE, ao contrário, é dispor de muitos componentes (módulos predefinidos) que podem ser combinados de várias maneiras para construir sistemas com diferentes comportamentos de resolução de problema.

Existem também algumas características de sistemas já desenvolvidos que não estão reproduzidas atualmente no sistema MOTIN, por exemplo, a organização de atributos em hierarquias, como ocorre no sistema S1, apesar disso ser possível com o esquema de rede semântica empregado; a possibilidade de representar relações n-árias entre elementos da base de dados, o que ocorre também no sistema S1 e em outros sistemas como por exemplo, KAS e ROSIE (Fain et alii, 1981) e, evidentemente, em ferramentas baseadas em programação lógica como PROLOG e DUCK; o emprego de múltiplas bases de dados, como ocorre por exemplo, nos sistemas ROSIE e KEE, ou o uso de um sistema de banco de dados, como existem nos sistemas R1 e ACE (Vesonder et alii, 1983); o emprego de estruturas hierárquicas para organizar o banco de regras, como acontece em sistemas como os propostos por Ferrante et alii (1984) e por Braun e Goldberg (1985); a manutenção de registros de dependência para controlar o raciocínio não-monotônico (deKleer et alii, 1979) como ocorre, por exemplo, nos sistemas EL e DUCK. Além disso, o esquema de rede semântica empregado no sistema MOTIN é bastante simples, comparado com abordagens conhecidas na literatura (Findler, 1979).

Existem entretanto algumas características interessantes no sistema MOTIN que não ocorrem em muitas ferramentas já desenvolvidas. Por exemplo, o emprego de variáveis em cláusulas de regras de decisão, com a conseqüente sofisticação dos mecanismos de "pattern-matching". Nos sistemas mais simples as cláusulas são meramente proposicionais;

em outros, o emprego de variáveis é bastante restrito, como, por exemplo, nos sistemas EMYCIN e S1, onde apenas o objeto das cláusulas é considerado como variável ou no sistema KAS, que não permite usar o "binding" de variáveis de cláusulas de premissa em ações de conseqüentes. Existem, evidentemente, sistemas em que o uso de variáveis em regras de decisão é bem mais sofisticado do que no sistema MOTIN, como por exemplo, em Simoni (1986), embora isto não seja comum em sistemas especialistas. Outra característica é o emprego de um esquema para representar o conhecimento bem estabelecido do domínio, apesar disto ser uma preocupação atual (Hart, 1982), principalmente no que se refere à estrutura e à função dos componentes do sistema sob estudo, em aplicações de diagnóstico (Davis, 1984), e já existir sistema especialista que emprega rede semântica para isso (Xiang and Srihari, 1985).

Há contudo no sistema MOTIN, duas características que lhe são próprias e que não ocorrem em ferramentas conhecidas da literatura: o tratamento dado ao problema de estabelecimento de contextos de aplicação de regras de decisão e o modelo de combinação de evidências.

Com relação aos contextos, a solução empregada em sistemas já desenvolvidos é, em geral, muito simples. No sistema DEVISER (Vere, 1983) por exemplo, desenvolvido para planejamento de ações, a premissa de uma regra consiste de pré-condições que devem ser retiradas do plano que está sendo construído, quando a regra é aplicada. Neste sistema o contexto é simplesmente, como a premissa, um conjunto de pré-condições. No caso do contexto no entanto, estas condições devem ser mantidas no plano mesmo depois da aplicação da regra. Não existe portanto o problema de instâncias de objetos. Em sistemas já desenvolvidos para domínios hierarquizados nos quais é possível existir várias instâncias de um mesmo objeto, orienta-se a aplicação de regras por meio de uma estrutura de árvore, com as limitações comentadas anteriormente. A utilização de uma estrutura de grafo, resultante da instanciação de objetos organizados numa rede semântica,

generaliza o mecanismo que vem sendo empregado por estes sistemas e amplia suas fronteiras de aplicação.

Quanto ao modelo de combinação de evidências empregando a teoria de Dempster-Shafer, apesar de existirem alguns trabalhos que estabelecem algoritmos eficientes para situações particulares (Barnett, 1981; Shafer and Logan, 1985), segundo pode-se apreender da literatura, ele ainda não é utilizado em sistemas especialistas, embora existam evidências que isto deve ocorrer brevemente (Gordon and Shortliffe, 1985). Uma extensão, proposta por Oblow (1985), combina a teoria de Dempster-Shafer com a teoria de conjuntos nebulosos (Zadeh, 1979) e, mesmo aumentando os problemas com a eficiência computacional, parece ser uma abordagem interessante para utilização em futuros sistemas especialistas.

É interessante comparar a arquitetura do sistema MOTIN com a da Figura 6.1, apresentada por Hayes-Roth (1985b) como a de um sistema baseado em regras bem sofisticado.

Existem muitos pontos em comum entre estas duas arquiteturas. Em ambas existe uma lista de regras que aguardam execução (**lista de regras ativas**, no MOTIN; **agenda**, na Figura 6.1) organizada segundo níveis de prioridade estabelecidos por **meta-regras**. Em ambas, um escalonador considera qual ação deve ser avaliada pelo **interpretador** (módulo **INFERÊNCIA**, na Figura 5.2).

No sistema MOTIN o **escalonador** (módulo **PLANEJAMENTO**) é responsável também pelo estabelecimento de contextos de aplicação. Na arquitetura da Figura 6.1 entretanto, o escalonador pode considerar **procedimentos de controle** criados especialmente para uma aplicação, característica que não ocorre no sistema MOTIN e que constitui uma extensão importante.

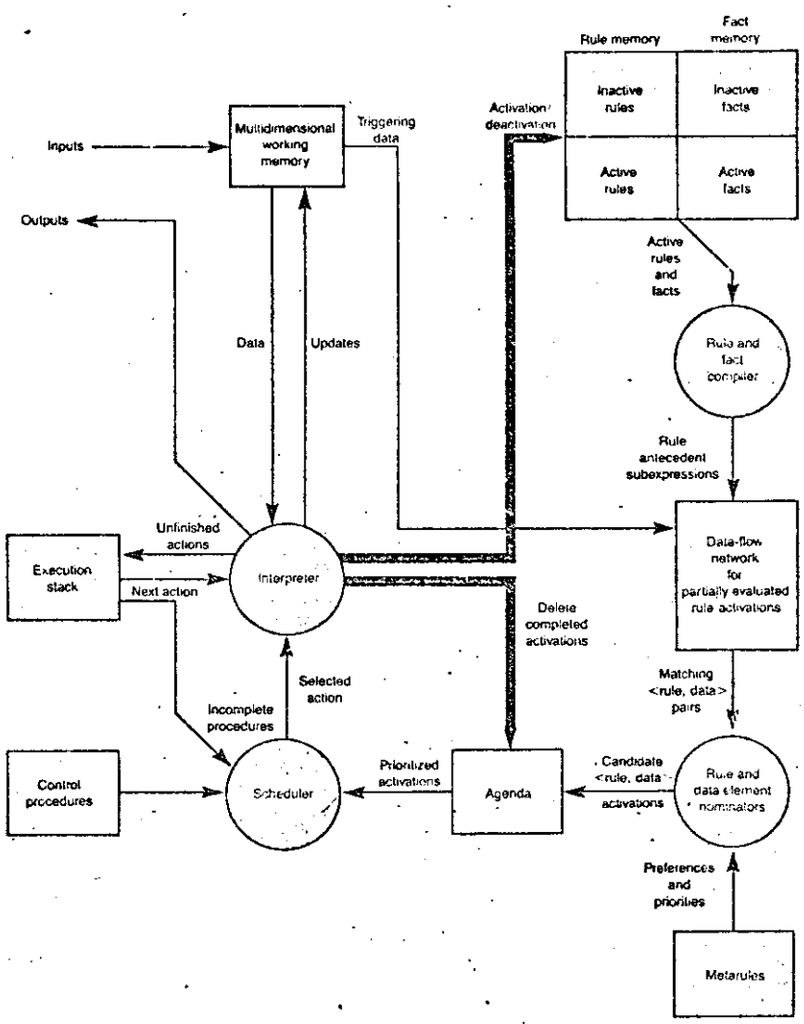

Fig. 6.1 - Um sofisticado sistema baseado em regras
FONTE: Hayes-Roth (1985b)

No sistema MOTIN as regras parcialmente avaliadas estão também armazenadas na lista de regras ativas; há contudo, a **agenda de tarefas** que armazena, em conjunto, todas as ações que aguardam execução. Na arquitetura apresentada por Hayes-Roth existe uma **pilha de execução** para armazenamento das regras parcialmente avaliadas. Esta estrutura sugere que as cláusulas de regras ativas são avaliadas na ordem em que elas aparecem nas premissas, sem a possibilidade de

alteração, o que não ocorre no sistema MOTIN, uma vez que a agenda de tarefas também está sujeita a ação de meta-regras.

Há também semelhança entre a **memória de trabalho multidimensional** da Figura 6.1 e o **banco de evidências** do sistema MOTIN. Hayes-Roth sugere que os dados obtidos durante o processo de inferência sejam armazenados numa estrutura que distingue dimensões como espaço, tempo ou nível de abstração. No sistema MOTIN é possível recuperar os elementos do banco de evidências considerando também estas dimensões - e outras - já que os novos primitivos conceituais são armazenados como propriedades das evidências. Existe no sistema MOTIN contudo, além desta estrutura, o **conjunto de evidências a propagar** que permite identificar claramente as evidências que devem ser levadas em conta quando é necessário combinar as estratégias de encadeamento progressivo e retroativo num único ciclo de execução.

A possibilidade de ativar e desativar classes de regras do banco de conhecimento é também uma semelhança entre as duas arquiteturas. No sistema MOTIN contudo, as regras inativas são armazenadas na **lista de regras desativadas** juntamente com o contexo para o qual a desativação ocorreu, havendo a possibilidade da mesma regra estar ativa para outros contextos. Entretanto, a possibilidade de desativar fatos, como sugere a Figura 6.1, não está presente no sistema MOTIN.

A preocupação com o desempenho motivou a inclusão de algumas estruturas na Figura 6.1 que não existem atualmente no sistema MOTIN: um **compilador de regras** que converte as condições de regras ativas numa **rede de fluxo de dados** que otimiza o esforço requerido para identificar regras executáveis. Embora exista no sistema MOTIN uma representação única para as cláusulas, o que facilita a identificação de regras executáveis, a inclusão destas estruturas é uma extensão importante para melhorar sua eficiência.

6.2 - ESTÁGIO ATUAL E EXTENSÕES FUTURAS

O sistema MOTIN está implementado na linguagem LISP para processamento em microcomputador do tipo IBM-PC (Apêndice C). Devido à flexibilidade de seus esquemas de representação de conhecimento e de seu mecanismo de inferência, espera-se poder utilizar este sistema em vários domínios de conhecimento.

O emprego da linguagem LISP é muito conveniente para desenvolver programas de Inteligência Artificial, dadas as suas capacidades para tratamento da informação simbólica. Entretanto, a facilidade no desenvolvimento tem seu preço: os sistemas escritos em LISP necessitam ser processados em máquinas especiais para alcançar eficiência computacional. Processados em máquinas convencionais, tais sistemas, em geral, são lentos. A implementação atual do sistema MOTIN não é uma exceção. Apesar de o sistema ter um desempenho razoável em sessões de aquisição de conhecimento, sente-se nitidamente a perda de eficiência computacional durante uma sessão de consulta, uma vez que o mecanismo de inferência envolve volume maior de processamento, sendo crítico o gerenciamento da memória. A sessão apresentada no Capítulo 5, por exemplo, apesar de simples e pequena - o banco de conhecimento envolvido possui apenas 45 regras e as intervenções do usuário podem ser feitas com rapidez - decorre em cerca de 5 minutos, 60% dos quais, aproximadamente, é consumido em operações de "garbage collection". Isto deve-se, em parte, à manipulação do banco de conhecimento que atualmente, é colocado todo em memória no início de uma sessão. Isto, obviamente, não é factível num sistema prático que envolve um grande número de regras e precisa ser melhorado. É preciso dizer portanto, que grande esforço necessita ser feito em busca de uma implementação eficiente do sistema MOTIN.

Existem também algumas extensões que podem ser realizadas, além das já comentadas na Seção 6.1. O módulo de aquisição de conhecimento, por exemplo, requer um **editor de regras** para facilitar a depuração e modificação do banco de conhecimento. Este editor deve

também levar em conta o conhecimento armazenado em rede semântica, atualizando-o caso necessário.

Outra extensão refere-se ao esquema de rede semântica que pode ser modificado de modo a representar **relações n-árias.** Apesar de ser possível representar uma relação n-ária como um conjunto de relações binárias, no sistema atual não há como indicar que **todo** o conjunto de relações binárias deve ser satisfeito para que se tenha a relação n-ária estabelecida.

Um grande volume de trabalho deve ser feito para incorporar ao sistema MOTIN um **módulo de explanação** capaz de fornecer a seus usuários boas justificativas para as ações e conclusões efetuadas durante uma sessão de consulta, o que poderá resultar em alterações na arquitetura do sistema. É importante neste módulo levar em conta, além das inferências empregando regras de decisão, as inferências realizadas na rede semântica, o que torna o processo de explanação mais complexo.

A questão da eficiência é importante para sistemas práticos e portanto, uma extensão possível é implementar mecanismos que propiciem eficiência, como por exemplo, as estratégias de resolução de conflito apontadas no Capítulo 5. Outra é a organização da base de conhecimento em múltiplas classes de regras, como existe no sistema KEE, com a possibilidade de considerar apenas um subconjunto do banco de regras numa determinada aplicação. Pode-se também identificar os procedimentos onerosos do programa, substituindo-os por implementações numa linguagem mais eficiente. Os algoritmos para tratamento de incerteza, por exemplo, podem ser facilmente traduzidos para uma linguagem de programação convencional.

Uma extensão interessante seria verificar a possibilidade de aumento de eficiência do sistema com a execução paralela das tarefas da agenda. Uma vez colocadas na agenda e estabelecidas as suas prioridades de execução, algumas tarefas, devido à existência de variáveis, necessitam ser executadas seqüencialmente. No entanto é

possível que algumas tarefas possam ser executadas de forma paralela, o que ocorre por exemplo, com as tarefas correspondentes a uma regra ativada para mais de um contexto de aplicação.

REFERÊNCIAS BIBLIOGRÁFICAS

BARNETT, J. A. Computational methods for a mathematical theory of evidence. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 7., Vancouver, BC, 1981. **Proceedings**. Los Altos, CA, Morgan Kaufmann, 1981, p. 868-875.

BARR, A.; FEIGENBAUM, E.A. **The handbook of artificial intelligence**, Los Altos, CA, William Kaufmann, 1981. V. 1.

BARSTOW, D.R.; AIELLO, N.; DUDA, R.O.; ERMAN, L.D.; FORGY, C.L.; GORLIN, D.; GREINER, R.D.; LENAT, D.B.; LONDON, P.E.; McDERMOTT, J.; NII, H.P.; POLITAKIS, P.; REBOH, R.; ROSENSCHEIN, S.; SCOTT, A.C.; VAN MELLE, W.; WEISS, S.M. Languages and tools for knowledge engineering. In: HAYES-ROTH, F.; WATERMAN, D.A.; LENAT, D.B., ed., **Building expert systems**, Reading, MA, Addison-Wesley, 1983. Cap. 9, p. 283-345

BITTENCOURT, G. **Intégration d'outils graphiques dans un logiciel de calcul formel**. Rapport de DEA, Grenoble, LIFIA, Juillet, 1986.

BOBROW, D.G.; RAPHAEL, B. New programming languages for artificial intelligence research. **Computing Surveys**, 6(3):153-174, Sep. 1974.

BRACHMAN, R. J.; SMITH, B. C., ed., Special issue on knowledge representation. **SIGART Newsletter**, 70, Feb. 1980.

BRAUN, F.; GOLDBERG, M. Un système général pour la construction d'une hiérarchie d'experts en télédétection. In: CONGRÈS RECONNAISSANCE DES FORMES ET INTELLIGENCE ARTIFICIELLE, 5., Grenoble, 1985. **Annales**. Grenoble, INRIA, 1985. p. 931-941.

BROWNSTON, L.; FARREL, R.; KANT, E.; MARTIN, N. **Programming expert systems in OPS5 - an introduction to rule-based programming**, Reading, MA, Addison-Wesley, 1985.

BUCHANAN, B.G. Expert systems. **Journal of Automated Reasoning**, 1(1):28-35, 1985.

BUCHANAN, B.G.; BARSTOW, D.; BECHTEL, R.; BENNET, J.; CLANCEY, W.; KULIKOWSKI, C.; MITCHELL, T.; WATERMAN, D.A. Constructing an expert system. In: HAYES-ROTH, F.; WATERMAN, D.A.; LENAT, D.B., ed., **Building expert systems**, Reading, MA, Addison-Wesley, 1983. Cap. 5, p. 127-167.

CHOURAQUI, E.; FARRENY, H.; KAYSER, D.; PRADE, H. Modélisation du raisonnement et de la connaissance. **Technique et Science Informatiques**, 4(4):391-399, 1985.

DAVIS, R. Diagnostic reasoning based on structure and behavior. In: BOBROW, D.G.; HAYES, P.J., ed., Special volume on qualitative reasoning about physical systems. **Artificial Intelligence**, 24(1-3):347-410, Dec. 1984.

DAVIS, R. Generalized procedure calling and content-directed invocation. **SIGART Newsletter**, 64:45-54, Aug. 1977.

DAVIS, R.; BUCHANAN, B.; SHORTLIFFE, E. Production rules as a representation for a knowledge-based consultation program. **Artificial Intelligence**, 8(1):15-45, Feb. 1977.

DAVIS, R.; KING, J. An overview of production systems. In: ELCOCK, E.W.; MICHIE, D., ed., **Machine Intelligence**, 8., New York, NY, Halsted, 1977. p. 300-332.

deKLEER, J.; DOYLE, J.; STEELE, G.L.; SUSSMAN, G.J. Explicit control of reasoning. In: WINSTON, P.J.; BROWN, R.H., ed., **Artificial intelligence: an MIT perspective**, Cambridge, MA, MIT, 1979. p. 93-116.

DOYLE, J. A truth maintenance system. **Artificial Intelligence**, 12(3):231-272, Nov. 1979.

DUDA, R.O.; HART, P.E.; NILSSON, N.J.; REBOH, R.; SLOCUM, J.; SUTHERLAND, G.L. **Development of a computer-based consultant for mineral exploration: annual report.** Menlo Park, CA, Stanford Research Institute, 1977.

ENGELMORE, R.; TERRY, A. Structure and function of the CRYSALIS system. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979. **Proceedings.** Los Altos, CA, Morgan Kaufmann, 1979. p. 250-256.

ERMAN, L.D.; HAYES-ROTH, F.; LESSER, V.R.; REDDY, D.R. The HEARSAY-II speech-understanding system: integrating knowledge to resolve incertainty, **Computing Surveys**, 12(2):213-253, June, 1980.

FAIN, J.; GORLIN, D.; HAYES-ROTH, F.; ROSENSCHEIN, S.J.; SOWIZRAL, H.; WATERMAN, D. **The ROSIE language reference manual**, Santa Monica, CA, RAND Corp., 1981. (Tech. Rep. N-1647-ARPA). Apud Fikes and Kehler (1985).

FERRANTE, R.D.; CARLOTTO, M.J.; POMAREDE, J.M.; BAIM, P.W. Multi-spectral image analysis system. In: CONFERENCE ON ARTIFICIAL INTELLIGENCE APPLICATIONS, 1., Denver, CO, 1984. **Proceedings.** Los Angeles, CA, IEEE Computer Society, 1984. p. 357-363.

FICKAS, S.; NOVICK, D. Control knowledge in expert systems: relaxing restrictive assumptions. In: INTERNATIONAL WORKSHOP: EXPERT SYSTEMS & THEIR APPLICATIONS, 5., Avignon, 1985. **Proceedings.** Paris, Agence de l'Informatique, 1985. p. 981-994.

FIKES, R.; KEHLER, T. The role of frame-based representation in reasoning. **Communications of the ACM**, 28(9):904-920, Sep. 1985

FINDLER, N.V., ed. **Associative networks: representation and use of knowledge by computers**, New York, NY, Academic, 1979.

FRIEDMAN, L. Extended plausible inference. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 7., Vancouver, BC, 1981. **Proceedings**. Los Altos, CA, Morgan Kaufmann, 1981, p. 487-495.

GARVEY, T.D.; LOWRENCE, J.D.; FISCHLER, M.A. An inference technique for integrating knowledge from disparate sources. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 7., Vancouver, BC, 1981. **Proceedings**. Los Altos, CA, Morgan Kaufmann, 1981, p. 319-325.

GEVARTER, W.B. **An overview of artificial intelligence and robotics**, Washington, DC, NASA, 1983. V. 1. (NASA Technical Memorandum 85838).

GORDON, J.; SHORTLIFFE, E.H. A method for managing evidential reasoning in a hierarchical hypothesis space. **Artificial Intelligence**, 26(3):323-357, July, 1985.

HART, P.E. Directions for AI in the eigthies. **SIGART Newsletter**, 79:11-16, Jan. 1982.

HAYES-ROTH, B. A blackboard architecture for control. **Artificial Intelligence**, 26(3):251-321, July, 1985.

HAYES-ROTH, F. Rule-based systems. **Communications of the ACM**, 28(9):921-932, Sep. 1985.

HAYES-ROTH, F. The knowledge-based expert system: a tutorial. **Computer**, 17(9):11-28, Sep. 1984.

HAYES-ROTH, F.; WATERMAN, D.A., ed., Proceedings of the workshop on pattern-directed inference sustems. **SIGART Newsletter**, 63, June, 1977.

HAYES-ROTH, F.; WATERMAN, D.A.; LENAT, D.B. An overview of expert systems. In: HAYES-ROTH, F.; WATERMAN, D.A.; LENAT, D.B., ed., **Building expert systems**, Reading, MA, Addison-Wesley, 1983. Cap. 1, p. 3-29.

KOWALSKI, R. **Logic for problem solving.** Amsterdam, North- Holland, 1979.

LENAT, D.B. AM: an artificial intelligence approach to discovery in mathematics as heuristic search. In: DAVIS, R.; LENAT, D.B., ed., **Knowledge-based systems in artificial intelligence**, New York, NY, McGraw-Hill, 1982. Cap. 1, p. 1-225.

McDERMOTT, J. R1: a rule-based configurer of computer systems, **Artificial Intelligence**, 19(1):39-88, Sep. 1982.

MINSKY, M. A framework for representing knowledge. In: WINSTON, P.H., ed., **The psychology of computer vision**, New York, NY, McGraw-Hill, 1975. p. 211-277.

NEGOITA, C.V. **Expert systems and fuzzy systems.** Menlo Park, CA, Benjamin/Cummings, 1985.

NEWELL, A.; SIMON, H.A. GPS: a program that simulates human thougth. In: FEIGENBAUM, E.A.; FELDMAN, J.A., ed., **Computers and thougth**, New York, NY, McGraw-Hill, 1963. p. 279-293.

NEWELL, A.; SIMON, H.A. **Human problem solving**, Englewood Cliffs, NJ, Prentice-Hall, 1972.

NII, H.P.; AIELLO, N. AGE (Attempt to GEneralize): a knowledge-based program for building knowledge-based programs. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979. **Proceedings**. Los Altos, CA, Morgan Kaufmann, 1979. p. 645-655.

NILSSON, N.J. **Problem-solving methods in artificial intelligence.** New York, NY, McGraw-Hill, 1971.

NORMAN, D.A.; RUMELHART, D.E., ed., **Explorations in cognition,** San Francisco, CA, W.H. Freeman, 1975.

OBLOW, E.W. A hybrid-uncertainty theory. In: INTERNATIONAL WORKSHOP: EXPERT SYSTEMS & THEIR APPLICATIONS, 5., Avignon, 1985. **Proceedings.** Paris, Agence de l'Informatique, 1985. p. 1193-1201.

OLIVEIRA, C.A. Linguagem natural: um sistema de relações?. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 6., Belo Horizonte, MG, 1986. **Anais.** Belo Horizonte, MG, UFMG, 1986. p. 215-219.

OLIVEIRA, P.P.B. **Geração de modelos de regras: uma abordagem centrada na aprendizagem indutiva.** Dissertação de Mestrado em Computação Aplicada, São José dos Campos, SP, INPE, Set. 1986. (INPE-4299-TDL/276)

PRADE, H. **Comment prendre en compte les aspects approximatifs du raisonnement humain dans les systèmes experts;** Notes de cours de l'Université Paul Sabatier, Toulouse, Oct. 1982.

QUILLIAN, M.R. Semantic memory. In: MINSKY, M., ed., **Semantic information processing.** Cambridge, MA, MIT, 1968. p. 227-270.

RICH, E. **Artificial intelligence,** New York, NY, McGraw- Hill, 1983.

ROBINSON, A. A machine-oriented logic based on the resolution principle. **Journal of the ACM,** 12(1):23-41, Jan. 1965.

SACERDOTI, E.D. **A structure for plans and behavior,** New York, NY, Elsevier, 1977.

SANDRI, S.A.. **Sistema DIAGNÓSTICO: uma ferramenta para a construção de sistemas especialistas.** Dissertação de Mestrado em Computação Aplicada, São José dos Campos, SP, INPE, Out. 1985. (INPE-3890-TDL/222)

SHAFER, G. **A mathematical theory of evidence.** Princeton, NJ, Princeton University, 1976.

SHAFER, G.; LOGAN, R. **Implementing Dempster's rule for hierarchical evidence.** Lawrence, KS, University of Kansas, 1985. (Working paper no. 174)

SHORTLIFFE, E.H. **Computer-based medical consultation: MYCIN**, New York, NY, Elsevier, 1976.

SHORTLIFFE, E.H.; BUCHANAN, B. A model of inexact reasoning in medicine. **Mathematical Biosciences**, 23:351-379, 1975.

SILVA, O.O. **Indução de regras de decisão nebulosas.** Tese de Doutorado em Computação Aplicada, São José dos Campos, SP, INPE, Jul. 1983. (INPE-2855-TDL/143)

SIMONI, P.O. **Uma abordagem para redes associativas com aplicação na análise de cenas.** Tese de Doutorado em Computação Aplicada, São José dos Campos, SP, INPE, Mar. 1986. (INPE-4036-TDL/245)

SMITH, D.E.; GENESERETH, M.R.; GINSBERG, M.L. Controlling recursive inference. **Artificial Intelligence**, 30(3):343- 389, Dec. 1986.

SMITH, R.G.; BAKER, J.D. The dipmeter advisor system - a case study in commercial expert system development. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 8., Karlsruhe, 1983. **Proceedings.** Los Altos, CA, Morgan Kaufmann, 1983. p. 122-129.

STALLMAN, R.M.; SUSSMAN, G.J. Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. **Artificial Intelligence**, 9(2):135-196, Oct. 1977.

STEFIK, M.; AIKINS, J.; BALZER, R.; BENOIT, J.; BIRNBAUM, L.; HAYES-ROTH, F.; SACERDOTI, E. The organization of expert systems, a tutorial. **Artificial Intelligence**, 18(2):135-173, Mar. 1982.

THOMPSON, T.F.; WOJCIK, R.M. MELD: an implementation of a meta-level architecture for process diagnosis. In: CONFERENCE ON ARTIFICIAL INTELLIGENCE APPLICATIONS, 1., Denver, CO, 1984. **Proceedings.** Los Angeles, CA, IEEE Computer Society, 1984. p. 321-330.

VAN MELLE, W. A domain-independent production rule sustem for consultation programs. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979. **Proceedings.** Los Altos, CA, Morgan Kaufmann, 1979. p. 923-925.

VAN MELLE, W.; SHORTLIFFE, E.H.; BUCHANAN, B.G. EMYCIN: a knowledge engineer's tool for constructing rule-based expert systems. In: BUCHANAN, B.G.; SHORTLIFFE, E.H., ed., **Rule-based expert systems**, Reading, MA, Addison- Wesley, 1985. Cap. 15, p. 302-313.

VERE, S.A. Planning in time: windows and durations for activities and goals. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 5(3):246-267, May 1983.

VESONDER, G.T.; STOLFO, S.J.; ZIELINSKI, J.E.; MILLER, F.D.; COPP, D.H. ACE: an expert system for telephone cable maintenance. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 8., Karlsruhe, 1983. **Proceedings.** Los Altos, CA, Morgan Kaufmann, 1983. p. 116-121.

WALLIS, J.W.; SHORTLIFFE, E.H. **Explanatory power for medical expert systems: studies in the representation of causal relationships for clinical consultations.** Stanford, CA, Stanford University, July, 1982 (Report no. STAN-CS-82-923).

WATERMAN, D.A. **A guide to expert systems**, Reading, MA, Addison-Wesley, 1986.

WATERMAN, D.A.; HAYES-ROTH, F. An investigation of tools for building expert systems. In: HAYES-ROTH, F.; WATERMAN, D.A.; LENAT, D.B., ed., **Building expert systems**, Reading, MA, Addison-Wesley, 1983. Cap. 6, p. 169-215.

WATERMAN, D.A.; HAYES-ROTH, F., ed., **Pattern-directed inference systems**, New York, NY, Academic, 1978.

WEISS, S.M.; KULIKOWSKI, C.A. **A practical guide to designing expert systems.** Totowa, NJ, Rowman & Allanheld, 1984.

WESLEY, L.P.; HANSON, A.R. The use of an evidential-based model for representing knowledge and reasoning about images in the VISIONS system. In: WORKSHOP ON COMPUTER VISION: REPRESENTATION AND CONTROL, Rindge, NH. **Proceedings.** IEEE, Computer Society, 1982, p. 14-25.

WINSTON, P.H.; HORN, B. **LISP.** Reading, MA, Addison-Wesley, 1981.

XIANG, Z.; SRIHARI, S.N. Spatial structure and function representation in diagnostic expert systems. In: INTERNATIONAL WORKSHOP: EXPERT SYSTEMS & THEIR APPLICATIONS, 5., Avignon, 1985. **Proceedings.** Paris, Agence de l'Informatique, 1985. p. 192-206.

ZADEH, L.A. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. **AI Magazine,** 7(2):85-90, 1986.

ZADEH, L.A. A theory of approximate reasoning. In: HAYES, J.E.; MICHIE, D.; MIKULICH, L.I., ed., **Machine Intelligence**, 9., New York, NY, Halsted, 1979. Cap. 7, p. 149-194.

APÊNDICE A

ALGUNS RESULTADOS ACERCA DO MODELO DE VERDADE

Neste apêndice demonstram-se alguns resultados sobre o modelo de verdade adotado no sistema MOTIN.

Teorema A.1

Seja θ o conjunto de valores de um predicado, $P(\theta)$, o conjunto dos subconjuntos de θ e V um subconjunto de $P(\theta)$ tal que $\theta \in V$. Seja $V' = \{v \mid -v \in V\}$, onde $-v$ corresponde ao complemento de v em relação a θ.

Seja $f : V \cup V' \to [0,1]$ uma função de atribuição de crença tal que:

$f(\phi) = 0$

$\Sigma\ f(x) = 1$, $x \in V \cup V'$.

Se

$Cr: V \cup V' \to [0,1]$ e $Pl: V \cup V' \to [0,1]$

são as funções de credibilidade e de plausibilidade, respectivamente, tais que:

$Cr(x) = \Sigma\ f(y)$, $y \subseteq x$

$Pl(x) = 1 - Cr(-x)$,

então, para todo $x \in V \cup V'$:

$$Cr(x) \leq Pl(x).$$

Prova:

Para todo $x \in V \cup V'$ tem-se:

$$Cr(x) = \sum_{y \subseteq x} f(y)$$

$$Pl(x) = 1 - Cr(-x) = \sum_{y \in V \cup V'} f(y) - \sum_{y \subseteq x'} f(y)$$

$$= \sum_{y \cap x \neq \phi} f(y)$$

Mas para quaisquer x e y pertencentes a $V \cup V'$, se $y \supseteq x$ então $y \cap x \neq \phi$. Logo:

$$Cr(x) \leq Pl(x).$$

Teorema A.2

Seja [c,p] um intervalo de crença e f o fator de crença da conclusão de uma regra de decisão. Então:

$$[f*c,\ 1 - f*(1 - p)]$$

é um intervalo de crença.

Prova:

Do enunciado do teorema tem-se:

$0 \leq c \leq 1$; $c \leq p \leq 1$; $0 \leq f \leq 1$

Deve-se mostrar que:

$$0 \leq f*c \leq 1 - f*(1 - p) \leq 1.$$

(a) Mostrar que: $0 \leq f*c$.

Trivial, pois $f \geq 0$ e $c \geq 0$.

(b) Mostrar que: $1 - f*(1 - p) \leq 1$.

$1 - f*(1 - p) \leq 1 \leftrightarrow 0 \leq f*(1 - p)$

$\leftrightarrow 0 \leq (1 - p)$, pois $f \geq 0$

$\leftrightarrow p \leq 1$.

(c) Mostrar que: $f*c \leq 1 - f*(1 - p)$.

$f*c \leq 1 - f*(1 - p) \leftrightarrow f*c + f*(1 - p) \leq 1$

$\leftrightarrow f*(1 - (p - c)) \leq 1$

Mas:

$$0 \leq f \leq 1$$

e

$1 - (p - c) \leq 1$, pois $p \geq c \geq 0$.

Portanto:

$$f*(1 - (p - c)) \leq 1.$$

Teorema A.3

Seja f um fator de crença de uma regra de decisão. Sejam [c,p] e [f*c, 1 - f*(1 - p)] dois intervalos de crença. Então:

$$1 - f*(1 - p) - f*c \geq p - c.$$

Prova:

$1 - f*(1 - p) - f*c \geq p - c$

$\leftrightarrow \quad 1 - f*(1 - p + c) \geq p - c$

$\leftrightarrow \quad 1 - f*(1 - p + c) - (p - c) \geq 0$

$\leftrightarrow \quad (1 - f) + f*(p - c) - (p - c) \geq 0$

$\leftrightarrow \quad (1 - f) - (1 - f)*(p - c) \geq 0$

$\leftrightarrow \quad (1 - f)*(1 - (p - c)) \geq 0$

$\leftrightarrow \quad (1 - (p - c)) \geq 0$,

pois como $0 \leq f \leq 1$, $(1 - f) \geq 0$

$\leftrightarrow \quad p - c \leq 1$

$\leftrightarrow \quad p \geq c.$

Portanto, pelo Teorema A.3, quando se emprega uma regra de decisão, à medida que o fator de crença f diminui, aumenta a largura do intervalo de crença, ou seja, aumenta a incerteza sobre a conclusão. Para os casos limites tem-se:

(a) f = 1

Neste caso: [f*c, 1 - f*(1 - p)] = [c,p],
ou seja, a crença na conclusão é igual à crença na premissa;

(b) f = 0

Neste caso: [f*c, 1 - f*(1 - p)] = [0,1],
ou seja, nada se pode dizer sobre a veracidade ou falsidade da conclusão.

APÊNDICE B

REGRAS DE DECISÃO PARA O DOMÍNIO DE METEOROLOGIA

Este apêndice apresenta as regras do sistema de precipitação pluviométrica descrito por Sandri (1986) e que foram reescritas na sintaxe adotada no sistema MOTIN. Como conseqüência desta nova forma das regras, algumas modificações foram introduzidas.

No banco original existem regras para precipitação em 24 horas e regras para precipitação em menos - genericamente - de 24 horas. Com a possibilidade de introduzir TEMPO como primitivo conceitual, considerou-se, apenas para ilustrar, que menos de 24 horas refere-se a 12 horas.

O domínio de valores de alguns atributos também foi modificado. Por exemplo, cláusulas como:

"a velocidade ... está aumentando"

"a intensidade ... está diminuindo"

...

foram reescritas utilizando-se variáveis quantificadas e assumindo domínios de atributos (velocidade, intensidade, ...) convertidos para intervalos.

Além disto, devido à possibilidade de estruturação de objetos, algumas regras tornaram-se mais simples. Por exemplo, BLOQUEIO, JATO e UMIDADE que originalmente foram considerados como atributos tornaram-se objetos e em conseqüência, cláusulas de premissa da forma EXISTE BLOQUEIO, EXISTE JATO, ..., foram eliminadas, incorporando estas condições nos contextos de aplicação das regras. Outras simplificações foram possíveis estabelecendo-se adequadamente a semântica de relações. A regra:

SE: 1) EXISTE JATO

ENTÃO: 1) CONDIÇÃO-ALTO-NÍVEL (SJC) = FAVORÁVEL , 0.70

por exemplo, foi retirada do banco de conhecimento e reescrita como meta-regra associada à relação PRESENÇA-DE (JATO, SJC).

O sistema MOTIN pode receber regras tanto interativamente como de modo "batch". Este apêndice mostra também como se deve montar um arquivo de regras. As linhas que começam com o símbolo "%" são consideradas comentários pelo módulo de aquisição de conhecimento.

```
% REGRA $R1
% SE:
    LOCAL (ZONA-FRONTAL) = PARANA    &
    VELOCIDADE (ZONA-FRONTAL) = 10    &
    ATIVIDADE (ZONA-FRONTAL) = "DESENVOLVENDO-SE"    &
    CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL

% ENTAO:
    C-G-E (SJC,TEMPO = 24) = FAVORAVEL , 0.80

%   Notar que é importante deixar uma linha em branco no final da
%   premissa e no final da conclusão de cada regra.
%
% REGRA $R2
% SE:
    LOCAL (ZONA-FRONTAL) = RIO-GRANDE-SUL    &
    VELOCIDADE (ZONA-FRONTAL) = 20    &
    ATIVIDADE (ZONA-FRONTAL) = "DESENVOLVENDO-SE"    &
    CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL

% ENTAO:
    C-G-E (SJC,TEMPO = 24) = FAVORAVEL , 0.60

% REGRA $R3
% SE:
    TODOS (*X, *Y), (*X > *Y)
    LOCAL (ZONA-FRONTAL) = RIO-GRANDE-SUL    &
    VELOCIDADE (ZONA-FRONTAL) = 20    &
```

```
    INTENSIDADE (ZONA-FRONTAL,TEMPO = *X)  <
         INTENSIDADE (ZONA-FRONTAL,TEMPO = *Y)    V
    VELOCIDADE (ZONA-FRONTAL,TEMPO = *X)   <
         VELOCIDADE (ZONA-FRONTAL,TEMPO = *Y)
    CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL

% ENTAO:
    C-G-E (SJC,TEMPO = 24) = FAVORAVEL , 0.20

% REGRA $R4
% SE:
    TODOS (*X, *Y), (*X > *Y)
    DISTANCIA (SISTEMA-SINOTICO,TEMPO = *X)  <
         DISTANCIA (SISTEMA-SINOTICO,TEMPO = *Y)    &
    INTENSIDADE (SISTEMA-SINOTICO) = POUCA

% ENTAO:
    C-G-E (SJC,TEMPO = 24) = FAVORAVEL , 0.70

% REGRA $R5
% SE:
    LOCAL (SISTEMA-SINOTICO) = SUL    &
    CONDICAO-ALTO-NIVEL (SJC) = DESFAVORAVEL

% ENTAO:
    C-G-E (SJC,TEMPO = 24) = DESFAVORAVEL , 0.90

% REGRA $R6
% SE:
    TODOS (*X), (*X PERTENCE [-24,0])
    CONDICAO-PRESSAO (SJC,TEMPO = *X) = FAVORAVEL    &
    DBU (SJC) < 3

% ENTAO:
    CONDICAO-LOCAL (SJC,TEMPO = 24) = FAVORAVEL , 0.80
```

```
% REGRA $R7
% SE:
    TODOS (*X), (*X PERTENCE [-24,0])
    CONDICAO-PRESSAO (SJC,TEMPO = *X) = FAVORAVEL    &
    DBU (SJC) > 3

% ENTAO:
    CONDICAO-LOCAL (SJC,TEMPO = 24) = FAVORAVEL , 0.60

% REGRA $R8
% SE:
    TODOS (*X), (*X PERTENCE [-24,0])
    CONDICAO-PRESSAO (SJC,TEMPO = *X) = DESFAVORAVEL    &
    DBU (SJC) < 3

% ENTAO:
    CONDICAO-LOCAL (SJC,TEMPO = 24) = FAVORAVEL , 0.40

% REGRA $R9
% SE:
    TODOS (*X), (*X PERTENCE [-24,0])
    CONDICAO-PRESSAO (SJC,TEMPO = *X) = DESFAVORAVEL    &
    DBU (SJC) > 3

% ENTAO:
    CONDICAO-LOCAL (SJC,TEMPO = 24) = DESFAVORAVEL , 0.70

% REGRA $R10
% SE:
    TODOS (*X), (*X PERTENCE [-3,0])
    CONDICAO-PRESSAO (SJC,TEMPO = *X) = FAVORAVEL    &
    DBU (SJC) < 3

% ENTAO:
    CONDICAO-LOCAL (SJC,TEMPO = 12) = FAVORAVEL , 0.80
```

```
% REGRA $R11
% SE:
    TODOS (*X), (*X PERTENCE [-3,0])
    CONDICAO-PRESSAO (SJC,TEMPO = *X) = FAVORAVEL    &
    DBU (SJC) > 3

% ENTAO:
    CONDICAO-LOCAL (SJC,TEMPO = 12) = FAVORAVEL , 0.40

% REGRA $R12
% SE:
    TODOS (*X), (*X PERTENCE [-3,0])
    CONDICAO-PRESSAO (SJC,TEMPO = *X) = DESFAVORAVEL    &
    DBU (SJC) < 3

% ENTAO:
    CONDICAO-LOCAL (SJC,TEMPO = 12) = FAVORAVEL , 0.40

% REGRA $R13
% SE:
    TODOS (*X), (*X PERTENCE [-3,0])
    CONDICAO-PRESSAO (SJC,TEMPO = *X) = DESFAVORAVEL    &
    DBU (SJC) > 3

% ENTAO:
    CONDICAO-LOCAL (SJC,TEMPO = 12) = DESFAVORAVEL , 0.70

% REGRA $R14
% SE:
    TODOS (*X, *Y), (*X > *Y)
    DISTANCIA (CAVADO, TEMPO = *X)   <
         DISTANCIA (CAVADO, TEMPO = *Y)    &
    TAMANHO (CAVADO, TEMPO = *X)     >
         TAMANHO (CAVADO, TEMPO = *Y)

% ENTAO:
```

CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL , 0.80

% REGRA $R15

% SE:

ATIVIDADE (NUVEM) = "DESENVOLVENDO-SE PELA MANHA"

% ENTAO:

CONDICAO-LOCAL (SJC,TEMPO = 12) = FAVORAVEL , 0.50

% REGRA $R16

% SE:

TODOS (*X, *Y), (*X > *Y)

DISTANCIA (VORTICE, TEMPO = *X) <

DISTANCIA (VORTICE, TEMPO = *Y)

% ENTAO:

CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL , 0.80

% REGRA $R17

% SE:

LOCAL (ZONA-FRONTAL) = SAO-PAULO &

ATIVA (ZONA-FRONTAL) = SIM

% ENTAO:

C-G-E (SJC,TEMPO = 12) = FAVORAVEL , 0.90

% REGRA $R18

% SE:

LOCAL (ZONA-FRONTAL) = PARANA &

ATIVA (ZONA-FRONTAL) = SIM &

CONDICAO-ALTO-NIVEL (SJC) = FAVORAVEL

% ENTAO:

C-G-E (SJC,TEMPO = 12) = FAVORAVEL , 0.70

% REGRA $R19

```
% SE:
    CONDICAO-LOCAL (SJC,TEMPO = 12) = FAVORAVEL    &
    ESTACAO (SJC) = VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 12) = SIM , 0.70

% REGRA $R20
% SE:
    MES (SJC) < 3    V
    MES (SJC) > 10

% ENTAO:
    ESTACAO (SJC) = VERAO , 1.00

% REGRA $R21
% SE:
    MES (SJC) PERTENCE [3,10]

% ENTAO:
    ESTACAO (SJC) <> VERAO , 1.00

% REGRA $R22
% SE:
    CONDICAO-LOCAL (SJC,TEMPO = 24) = FAVORAVEL    &
    ESTACAO (SJC) = VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 24) = SIM , 0.60

% REGRA $R23
% SE:
    CONDICAO-LOCAL (SJC,TEMPO = 12) = FAVORAVEL    &
    ESTACAO (SJC) <> VERAO

% ENTAO:
```

```
    PRECIPITACAO (SJC,TEMPO = 12) = SIM , 0.40

% REGRA $R24
% SE:
    CONDICAO-LOCAL (SJC,TEMPO = 24) = FAVORAVEL    &
    ESTACAO (SJC) <> VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 24) = SIM , 0.50

% REGRA $R25
% SE:
    C-G-E (SJC,TEMPO = 12) = FAVORAVEL    &
    ESTACAO (SJC) = VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 12) = SIM , 0.90

% REGRA $R26
% SE:
    C-G-E (SJC,TEMPO = 12) = FAVORAVEL    &
    ESTACAO (SJC) <> VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 12) = SIM , 0.80

% REGRA $R27
% SE:
    C-G-E (SJC,TEMPO = 24) = FAVORAVEL    &
    ESTACAO (SJC) = VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 24) = SIM , 0.80

% REGRA $R28
% SE:
```

```
    C-G-E (SJC,TEMPO = 24) = FAVORAVEL    &
    ESTACAO (SJC) <> VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 24) = SIM , 0.70

% REGRA $R29
% SE:
    HORA (SJC) PERTENCE [10,16] V
    HORA (SJC) PERTENCE [0,4]

% ENTAO:
    EVOLUCAO-PRESSAO (SJC) = "DEVERIA DIMINUIR" , 1.00

% REGRA $R30
% SE:
    HORA (SJC) PERTENCE [4,10] V
    HORA (SJC) PERTENCE [16,24]

% ENTAO:
    EVOLUCAO-PRESSAO (SJC) = "DEVERIA AUMENTAR" , 1.00

% REGRA $R31
% SE:
    CARTA-ISALOBARICA (SJC) PERTENCE [-10,-6)

% ENTAO:
    CONDICAO-PRESSAO (SJC,TEMPO = 24) = FAVORAVEL , 0.80

% REGRA $R32
% SE:
    CARTA-ISALOBARICA (SJC) PERTENCE [-6,-2)

% ENTAO:
    CONDICAO-PRESSAO (SJC,TEMPO = 24) = FAVORAVEL , 0.50
```

```
% REGRA $R33
% SE:
    CARTA-ISALOBARICA (SJC) PERTENCE [-2,0)

% ENTAO:
    CONDICAO-PRESSAO (SJC,TEMPO = 24) = FAVORAVEL , 0.20

% REGRA $R34
% SE:
    TODOS (*X, *Y), (*X > *Y)
    PRESSAO (SJC,TEMPO = *X) < PRESSAO (SJC,TEMPO = *Y)    &
    EVOLUCAO-PRESSAO (SJC) = "DEVERIA DIMINUIR"    &
    TENDENCIA-PRESSAO (SJC) < -3

% ENTAO:
    CONDICAO-PRESSAO (SJC,TEMPO ε ⌊-3,0]) = FAVORAVEL , 0.6

% REGRA $R35
% SE:
    TODOS (*X, *Y), (*X > *Y)
    PRESSAO (SJC,TEMPO = *X) > PRESSAO (SJC,TEMPO = *Y)    &
    EVOLUCAO-PRESSAO (SJC) = "DEVERIA AUMENTAR"    &
    TENDENCIA-PRESSAO (SJC) > -3

% ENTAO:
    CONDICAO-PRESSAO (SJC,TEMPO ε [-3,0]) = DESFAVORAVEL , 0.6

% REGRA $R36
% SE:
    TODOS (*X, *Y), (*X > *Y)
    PRESSAO (SJC,TEMPO = *X) = PRESSAO (SJC,TEMPO = *Y)    &
    EVOLUCAO-PRESSAO (SJC) = "DEVERIA DIMINUIR"

% ENTAO:
    CONDICAO-PRESSAO (SJC,TEMPO ε [-3,0]) = DESFAVORAVEL , 0.6
```

```
% REGRA $R37
% SE:
    TODOS (*X, *Y), (*X > *Y)
    PRESSAO (SJC,TEMPO = *X) = PRESSAO (SJC,TEMPO = *Y)    &
    EVOLUCAO-PRESSAO (SJC) = "DEVERIA AUMENTAR"

% ENTAO:
    CONDICAO-PRESSAO (SJC,TEMPO ε [-3,0]) = FAVORAVEL , 0.6

% REGRA $R38
% SE:
    CONDICAO-LOCAL (SJC,TEMPO = 24) <> FAVORAVEL    &
    ESTACAO (SJC) = VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 24) = NAO , 0.50

% REGRA $R39
% SE:
    CONDICAO-LOCAL (SJC,TEMPO = 24) <> FAVORAVEL    &
    ESTACAO (SJC) <> VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 24) = NAO , 1.00

% REGRA $R40
% SE:
    C-G-E (SJC,TEMPO = 24) <> FAVORAVEL    &
    ESTACAO (SJC) = VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 24) = NAO , 0.50

% REGRA $R41
% SE:
    C-G-E (SJC,TEMPO = 24) <> FAVORAVEL    &
```

```
    ESTACAO (SJC) <> VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 24) = NAO , 0.70

% REGRA $R42
% SE:
    CONDICAO-LOCAL (SJC,TEMPO = 12) <> FAVORAVEL   &
    ESTACAO (SJC) = VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 12) = NAO , 0.80

% REGRA $R43
% SE:
    CONDICAO-LOCAL (SJC,TEMPO = 12) <> FAVORAVEL   &
    ESTACAO (SJC) <> VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 12) = NAO , 1.00

% REGRA $R44
% SE:
    C-G-E (SJC,TEMPO = 12) <> FAVORAVEL   &
    ESTACAO (SJC) = VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 12) = NAO , 0.70

% REGRA $R45
% SE:
    C-G-E (SJC,TEMPO = 12) = DESFAVORAVEL   &
    ESTACAO (SJC) <> VERAO

% ENTAO:
    PRECIPITACAO (SJC,TEMPO = 12) = NAO , 0.90
```

```
%   O arquivo de regras deve terminar com 2 linhas NIL.
%
NIL
NIL
```

APÊNDICE C

IMPLEMENTAÇÃO DO SISTEMA MOTIN

Este apêndice mostra como o sistema MOTIN está organizado, seus módulos componentes e seus relacionamentos com a arquitetura mostrada na Figura 5.2.

O programa está implementado na linguagem LISP para processamento em microcomputador compatível com IBM-PC. Uma medida do tamanho do programa é o número de linhas, com indentação e comentários, de seu código fonte: aproximadamente 9000 (cerca de 250 "Kbytes"). A Figura C.1 mostra a organização do sistema.

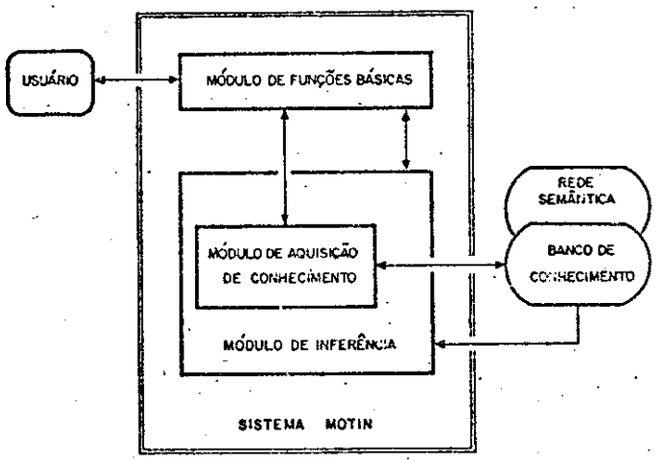

Fig. C.1 - Organização do sistema MOTIN

O usuário comunica-se com o sistema através do módulo de funções básicas. A partir deste módulo tem-se acesso aos módulos de aquisição de conhecimento e de inferência. No caso de a interação com o usuário ser apenas para atualizar o banco de conhecimento, com a aquisição de novas regras de decisão ou meta-regras, ou apenas para atualizar o conhecimento armazenado em rede semântica, carrega-se na memória somente o módulo de aquisição de conhecimento. Se, por outro

lado, a interação corresponder a uma sessão de consulta, além do módulo de inferência carrega-se na memória o módulo de aquisição de conhecimento, pois durante o processo de resolução de problema é possível estabelecer novas regras ou relações.

Mostra-se a seguir como estão constituídos os módulos do sistema MOTIN e quais são as funções de seus sub-módulos.

MÓDULO DE FUNÇÕES BÁSICAS

a) MFB-0.MFB:
   - Funções para armazenar, recuperar e exibir a base de conhecimento;
   - Funções do analisador sintático de entradas do usuário;
   - Funções para tratamento de opções:
     !AB - terminar prematuramente uma sessão;
     !BA - estabelecer a entrada de regras via "batch";
     !CR - alterar a credibilidade mínima;
     !PL - alterar a plausibilidade mínima;
     !MT - alterar o modo de trabalho do procedimento de inferência.

MÓDULO DE AQUISIÇÃO DE CONHECIMENTO

a) MAC-0.MAC:
   - Carrega o módulo de aquisição de conhecimento.

b) MAC-1.MAC:
   - Inicialização do banco de conhecimento;
   - Funções básicas para aquisição de conhecimento.

c) MAC-2.MAC:
   - Aquisição de nova regra ou meta-regra.

d) MAC-3.MAC:
   - Determinação de grafos de contexto de uma regra.

e) MAC-4.MAC:
   - Aquisição de novas relações.

MÓDULO DE INFERÊNCIA

a) MIN-0.MIN:

- Carrega o módulo de inferência.

b) MIN-1.MIN:

- Funções básicas do módulo de inferência.

c) MIN-2.MIN:

- Determinação de contexto de aplicação de regras;
- Tratamento de incerteza.

d) MIN-3.MIN:

- Inicialização e término de uma sessão de consulta.

e) MIN-4.MIN:

- Semântica de funções de cláusulas;
- Pré-análise de regras;
- Utilização de meta-regras.

f) MIN-5.MIN:

- Execução de tarefas.

g) MIN-6.MIN:

- Utilização de regras para resolução de problema.