

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

Dr. Múcio Roberto Dias



Presidente

Dr. Celso de Renna e Souza



Orientador

Dr. Fernando Curado



Membro da Banca
-convidado-

Dr. Emmanuel Piseces Lopes Passos



Membro da Banca
-convidado-

Dr. Nelson Delfino d'Ávila Mascarenhas



Membro da Banca

Candidato: Paulo Ouverá Simoni

São José dos Campos, 07 de março de 1986

A minha esposa e filhas.

AGRADECIMENTOS

Ao INPE pela oportunidade de desenvolver este trabalho.

Ao Dr. Celso de Renna e Souza, pela orientação e incentivo.

Ao Arry, Cristina Romeiro e operadores do B-6800 que, com suas informações e solicitude, permitiram que o programa pudesse ser testado.

A todas as demais pessoas que direta ou indiretamente, propiciaram condições, efetuaram críticas e deram seu estímulo para a realização deste trabalho.

ABSTRACT

An associative network knowledge representation model is presented, which uses concept structuring and a network structure both for formulae, that relate concepts and predicates, and for decision rules with a syntax that allows for quantified expressions in the premisses and conclusions. The possibilities of achieving knowledge compression with a particular type of part relation inference are discussed, as well as those of having conflicts among inferred relations. The stored knowledge is employed through a search process in which it is checked if activated hypothesis or predicates satisfy the formulae or rules, whenever they are accessed by the control in order to answer some question, to find solutions for a problem or to propagate instantiated elements. The control mechanism allows forward and backward chaining of decision rules and of formulae in general, by means of two control structures, one for alterations to be propagated and other for problems to be solved, and also allows nonmonotonic reasoning. A mixed initiative mechanism for scene analysis applications is proposed by means of which scene-related processes can be activated both from the use of formulae and rules or in a spontaneous fashion. Details about a computer program, which implemented the proposed model, and the developed applications are commented.

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	xi
<u>CAPÍTULO 1 - INTRODUÇÃO</u>	1
1.1 - O problema em estudo	1
1.2 - Comentários sobre processamento humano de informações	2
1.3 - Abordagens para o problema de análise de cenas	6
1.4 - Alguns modelos de representação de conhecimento	10
1.5 - A abordagem proposta	14
1.6 - Organização do trabalho	17
<u>CAPÍTULO 2 - ARMAZENAMENTO DE CONHECIMENTO</u>	19
2.1 - Introdução	19
2.2 - Descrição do modelo	20
2.2.1 - Propriedades e conceitos	20
2.2.2 - Hierarquia de partes	23
2.2.3 - Hierarquia de particularizações	24
2.2.4 - Instanciação de conceitos	24
2.2.5 - Lista de propriedades	25
2.3 - Condições de transferência e consistência de estruturas	27
2.4 - Inconsistência entre propriedades	33
2.5 - Um exemplo de estruturas de armazenamento	35
<u>CAPÍTULO 3 - MECANISMO DE CONTROLE</u>	39
3.1 - Introdução	39
3.2 - Estruturas do banco de hipóteses	41
3.3 - Acesso a propriedades	45
3.3.1 - Informações para acesso	45
3.3.2 - Mecanismo de acesso a propriedades	52
3.4 - Indicadores de condições de utilização de propriedades	58
3.4.1 - Atribuição de indicadores de direção de fluxo de execução.	58
3.4.2 - Mecanismo de atribuição de indicadores	62

	<u>Pág.</u>
3.5 - Utilização de propriedades	64
3.5.1 - Condições para utilização de propriedades	64
3.5.2 - Mecanismo para utilização de propriedades	67
3.6 - Resolução de problemas	74
3.6.1 - Obtenção de soluções e geração de subproblemas	74
3.6.2 - Encadeamento retroativo e progressivo de propriedades	76
3.7 - Controle de ações	80
3.7.1 - Características principais	80
3.7.2 - Detalhes sobre o mecanismo de controle	81
<u>CAPÍTULO 4 - PROCESSO DE ANÁLISE</u>	85
4.1 - Estruturas de descrição de uma cena	85
4.2 - Características do processo de análise	86
4.3 - Aplicação em segmentação controlada pela interpretação	89
4.4 - Aplicação em separação de regiões	95
4.5 - Comparação com outros modelos	100
4.5.1 - Modelos de representação	100
4.5.2 - Modelos de análise de cenas	105
<u>CAPÍTULO 5 - CONCLUSÕES</u>	109
5.1 - Resultados obtidos	109
5.2 - Extensões futuras	112
REFERÊNCIAS BIBLIOGRÁFICAS	115
APÊNDICE A - LINGUAGEM DE ACESSO À REDE ASSOCIATIVA	
APÊNDICE B - DETERMINAÇÃO DE CAMINHOS ENTRE NÓS	
APÊNDICE C - IMPLEMENTAÇÃO DO MODELO	

LISTA DE FIGURAS

	<u>Pág.</u>
1.1 - Esquema de "merge" controlado pela interpretação	7
1.2 - Esquema de análise por regras	8
1.3 - Esquema de análise por "schemata"	9
1.4 - Proposições e quantificadores	11
1.5 - Papéis e relações estruturais	12
1.6 - Transferência de papéis e relações	13
1.7 - Esquema proposto para processo de análise	16
2.1 - Estruturas de propriedade e de hierarquia de partes	26
2.2 - Combinações possíveis de relações	29
2.3 - Casos de transferência de relação de parte	30
2.4 - Transferência de relação e inversa	31
2.5 - Inibição de transferência de relação	32
4.1 - Cena para teste de fusão	93
4.2 - Cena para teste de separação	99
B.1 - Dígrafo auxiliar para o primeiro exemplo	B.2
B.2 - Dígrafo auxiliar para o segundo exemplo	B.2
B.3 - Dígrafo auxiliar para o terceiro exemplo	B.3
B.4 - Transferências possíveis em um caminho	B.8
B.5 - Obtenção de relações para $p(i)$ vinda de (b)	B.9
B.6 - Obtenção de relações para $p(i)$ vinda de (c)	B.10

CAPÍTULO 1

INTRODUÇÃO

1.1 - O PROBLEMA EM ESTUDO

Na pesquisa sobre o problema de análise de cenas por computadores, algumas tentativas já realizadas ou em andamento levam em conta um número bastante grande de fatores que influenciam tal análise e que se interagem. Estes fatores envolvem análises a partir de características devido a cores, de descritores de forma, de condições para segmentação, de informações sobre o tipo específico de tarefa a ser realizada, etc. É difícil estudar a influência de fatores específicos ou estabelecer relações de dependência entre eles.

A utilização de informações sobre um domínio de aplicação para direcionar o processo de análise de cenas já foi proposta anteriormente e abordada por outros autores. A complexidade do problema abordado não esgota, no entanto, as possibilidades de novos experimentos; os resultados até hoje obtidos podem ser bons para uma tarefa específica, mas as generalizações não têm sido bem sucedidas.

No presente trabalho, utiliza-se uma estrutura única de armazenamento de informações. Ela comporta desde informações de caráter geral sobre um universo de aplicação até condições para ativação de um processo genérico sobre uma cena. Através desta estrutura, e da linguagem que é utilizada para a introdução de novas informações, é possível acessar todas as demais informações descritas sobre um universo ou sobre um processo envolvido na análise. É utilizada, também, uma maneira de propagar informações que têm semelhança com aquelas supostas para a memória humana.

A estrutura de armazenamento e o processo de propagação podem ser utilizados para testar algoritmos que utilizem informações

sobre um universo; a introdução de novas características dos algoritmos, que envolvem parcelas diversas de conhecimento, é facilitada com o emprego de uma estrutura única.

Neste trabalho, a descrição pictórica da cena comporta apenas regiões e bordas; a introdução de outros tipos, no entanto, pode eventualmente ser feita, desde que conhecidos seus componentes.

1.2 - COMENTÁRIOS SOBRE PROCESSAMENTO HUMANO DE INFORMAÇÕES

Ao proceder à análise de diversos tipos de cenas pode ser conjecturado que, se existir uma maneira sistemática de introduzir conhecimentos sobre o tipo de cena e sobre possíveis elementos constituintes, então poder-se-ão escolher medidas mais adequadas para chegar a identificações mais confiáveis. Tal conhecimento poderia ser útil, além disso, em situações em que as medidas que podem ser obtidas não são suficientes (caso de oclusão, por exemplo) ou são em número bastante grande.

Diversos trabalhos na área de Inteligência Artificial já tentaram mostrar a viabilidade de empregar modelos de representação de conhecimento e de processamento no tratamento de vários tipos de problemas. Processamento de linguagem natural, de trechos contínuos de discurso, diagnóstico médico, simulação de jogos de tabuleiro são algumas tentativas feitas; em algumas pode ser notado um desempenho comparável ao de seres humanos (embora medidas de desempenho não necessariamente traduzam fidelidade).

Em muitos casos, o que uma pessoa faz conscientemente pode ser modelado; aquilo que faz inconscientemente, no entanto, constitui um campo de estudo que possibilitaria a explicação de muitos comportamentos. Ao tentar reproduzir o comportamento, deve-se atribuir à máquina a capacidade de executar os dois tipos de ação. Alguns resultados descritos por McKean (1985) indicam que as pessoas efetuam um armazenamento de informações que lhes foram apresentados por intervalos de

tempo suficientemente curtos para que possam ter consciência da apresentação ou em condições de cegueira de um dos campos visuais devido a um dano cerebral.

As atividades ligadas à percepção podem ser descritas em relação a seus resultados supondo, muitas vezes, a presença de padrões organizados (que podem ser conjecturados tanto ao nível de percepção visual como em casos que envolvem maiores abstrações, como configurações de tabuleiros de xadrez). A maneira pela qual se chega aos resultados não é bem compreendida ainda, mesmo usando métodos de análise; pode ser mencionado o comentário atribuído a Pascal, segundo o qual a mente humana funciona naturalmente, sem normas.

De interesse imediato para problemas que envolvam processamento visual seria um conhecimento de teorias relativas à percepção (visual) humana, que estabelecem leis de organização, sem estarem presas a processos. Tais teorias são conflitantes em muitos pontos básicos quanto interpretam o fenômeno perceptivo. A teoria "gestalt" supõe a percepção como uma "gestalt" irredutível às sensações elementares; percebe-se uma estrutura como um todo, existindo leis que regem a formação de agrupamentos (proximidade, igualdade, contorno, boa curva, etc.), que alguns resumem na lei da melhor forma e que se impõem ao homem sem que ele as crie. De acordo com esta teoria, a inteligência não influencia a percepção, sendo a forma função de condições objetivas do modelo, existindo antes de qualquer aprendizado (Doria, 1968).

Na concepção tomista, o conhecimento sensível é uma operação dos sentidos, distinto do conhecimento intelectual devido a sua particularidade; a forma não é aquilo que é conhecido, mas aquilo pelo que se conhece um objeto.

Na concepção associacionista, a percepção é um fenômeno de composição de elementos primitivos, as sensações; recebe-se um mosaico de estímulos que são recompostos de acordo com leis de associação. Ou

tras concepções postulam que a percepção é uma função da inteligência; perceber é entender o que é percebido, sendo a forma a representação intelectual do objeto.

Pela análise das teorias existentes e de resultados experimentais, podem ser inferidas algumas evidências de como se devem comportar os processos perceptivos. Não se pode afirmar, com certeza, até que ponto o conhecimento passado sobre objetos influencia sua identificação, nem se o todo é sempre percebido antes das partes. Tais incertezas por um lado tornam difícil a reprodução do comportamento humano e a descoberta de maneiras de evitar processos de busca exaustivos necessários para executar as mesmas atividades; por outro lado, abrem a possibilidade de estabelecer diversos modelos alternativos e estudar como se comportam.

A extensão de métodos de análise (de protocolos e de tempos de resposta) a situações que envolvem atividades perceptuais tem sido tentada procurando determinar como se chegam aos resultados. Um trabalho neste sentido foi o de Akin e Reddy (1977), em que indivíduos tentam descobrir o conteúdo de uma cena oculta, gerando protocolos verbais, podendo o indivíduo obter informações tais como cor, orientação de regiões, etc., de um examinador. Outro trabalho, o de Navon (1977), analisa tempos de resposta de indivíduos que tentam perceber o conteúdo, em diversos níveis de detalhe, de uma cena simples apresentada rapidamente. É proposto que o processamento vai do nível global para níveis locais, levando a uma organização espacial que pode ser suficiente para o reconhecimento; os detalhes seriam detectados à medida que forem essenciais para determinar o conteúdo e não se tenta explicar que partes recebem atenção ao longo do tempo. Além disso, o contexto permite ignorar inconsistências ou completar características não-observadas. Foi observado que não se pode tirar a atenção do todo para os detalhes; a percepção de um todo não seria obtida a partir da percepção dos elementos.

Por outro lado, as experiências de Biederman (1972) apresentam uma evidência de que o reconhecimento de objetos de uma cena é influenciado pela presença de outros objetos próprios do tipo de cena.

Zeidel (1984) comenta as diferenças funcionais entre os hemisférios esquerdo (produção de discurso, leitura, tratamento de informações novas) e direito (percepção espacial, memorização de elementos familiares, reconhecimento de faces) do cérebro e, em particular, as diferenças relativas à percepção visual, observadas em pessoas com problemas em apenas um dos lados ou que foram submetidas à separação das partes. Por exemplo, existe diferença quanto à observação de detalhes num contexto familiar.

Pode ser mencionada a observação de que pessoas eventualmente memorizam o que gostariam de ter visto e não o que realmente vêem. Em contraste, pessoas submetidas à hipnose podem se lembrar de detalhes que aparentemente não tinham consciência. A falta de capacidade de direcionar a atividade perceptiva pode indicar um tipo de processamento não-controlado conscientemente.

De alguma forma, parece ser plausível a conjectura que, quando se tenta reconhecer algo, algumas descrições (ou diversas delas, redundantes) são acessadas, procurando-se obter um "casamento" da descrição como uma dentre as várias que podem ser obtidas. Neste processo, alguns problemas que podem ocorrer são os de partes que faltam; partes que se supõem pertencer à descrição e não estão nas descrições conhecidas; descrições que não se "casam" bem com as conhecidas, mas podem ser consideradas como um caso particular de alguma delas; ordem de acesso a diversas descrições; verificação de identificação através de descrições alternativas. Pode ocorrer, também, a necessidade de observar outros detalhes daquilo que está sendo objeto da verificação.

1.3 - ABORDAGENS PARA O PROBLEMA DE ANÁLISE DE CENAS

Esta área de pesquisa tem sido objeto de diferentes abordagens que tentam levar em conta aspectos observáveis ou hipóteses sobre o comportamento do sistema visual humano. Procurou-se limitar aquelas mais diretamente relacionadas com este trabalho, nos comentários a seguir.

Para o problema de analisar cenas de blocos, tenta-se chegar a uma rotulação final (independente da interpretação dos componentes), a partir de um conhecimento sobre tipos possíveis de rótulos, por eliminação de impossibilidades (Clowes, 1971; Waltz, 1975; Mackworth, 1977). Alguns problemas observados são os de supor a figura pré-segmentada e as superfícies planas (a extensão a superfícies curvas vem sendo tentada), os de rotulações impossíveis e os de figuras com ruído (que levam a bordas e vértices espúrios); existem programas que tentam contornar o problema de bordas não-detectadas.

Outro tipo de abordagem foi a de Preparata e Ray (1972) em que uma cena pré-segmentada é analisada com o auxílio de um mapa semântico, no qual se usa uma relação de contexto entre objetos (ou de talhes sucessivos, pelo englobamento) e uma medida global de confiança. A cada elemento do mapa estão associados atributos que o caracterizam (cor, textura, propriedades morfológicas) e existe uma relação de plausibilidade entre cada par de nós, a qual expressa a possibilidade de um objeto estar acima de outro. Emprega-se um algoritmo "top down", em que o contexto tem papel predominante: uma vez interpretado um domínio, os seus detalhes serão procurados entre os detalhes do nó correspondente do mapa semântico, tentando-se evitar ambigüidades (em vez de tentar resolvê-las); reduz-se, também, a extração de atributos. Um inconveniente é a rigidez que um mapa semântico relacionado com englobamento entre domínios impõe; a busca baseada somente num contexto pode levar à rejeição prematura de interpretações.

Um inconveniente de supor fornecida a segmentação é que uma segmentação aceitável pode depender da interpretação e, por sua vez, a interpretação depender da segmentação fornecida. Uma tentativa de efetuar o "merge" de acordo com possíveis interpretações (Tenenbaum and Barrow, 1976; Barrow and Tenenbaum, 1976) é usar conhecimentos gerais, específicos de um domínio e específicos de uma figura para restringir as possíveis interpretações (Figura 1.1). Tal tentativa baseia-se no fato que uma eventual similaridade pictórica não é sempre um bom critério para formar regiões que correspondam a objetos. O processo segue a idéia da propagação de rótulos de Waltz (1975), eliminando interpretações que não sejam consistentes com nenhuma interpretação de uma região adjacente. Uma eventual fonte de erros é a falta de "backtracking" no caso de "merges" não-seguros.

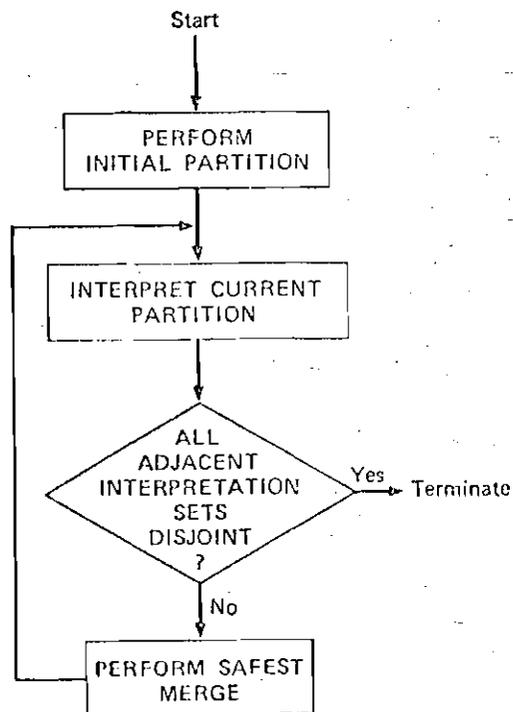


Fig. 1.1 - Esquema de "merge" controlado pela interpretação.

FONTE: Tenenbaum and Barrow (1976).

Uma abordagem tentada para considerar o problema de "merge" orientado (Ohta et alii, 1979; Ohta, 1980) faz uso de regras de decisão tipo conseqüente para executar "merge" de segmentos num esquema "top-down" (colocando ações numa agenda) e regras tipo antecedente, ativadas sempre que uma ação é executada (Figura 1.2). O processo de análise é dividido em fases de cena e de objeto (em que a análise é feita no contexto da cena).

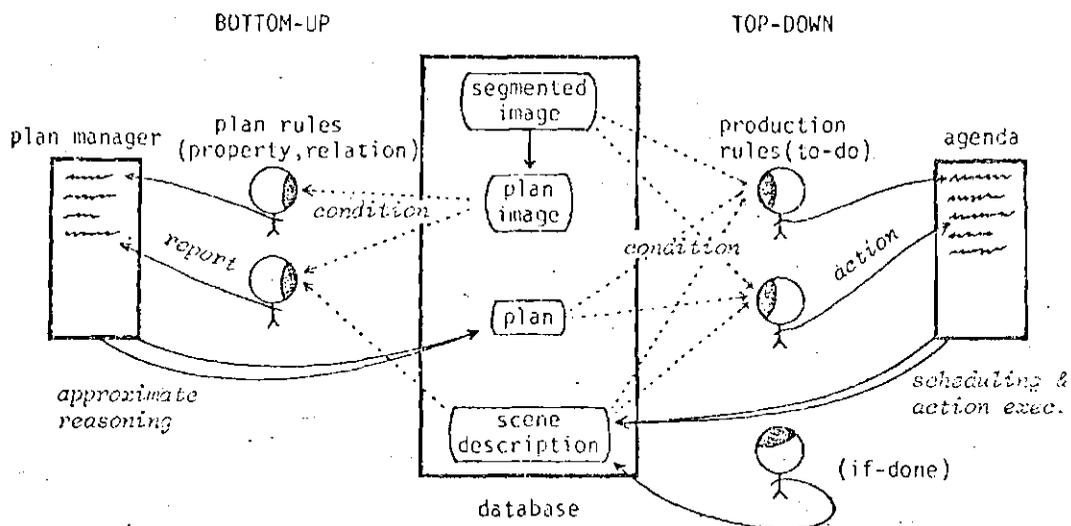


Fig. 1.2 - Esquema de análise por regras.

FONTE: Ohta (1980, p. 9).

Um sistema para o entendimento de mapas desenhados a mão (Havens and Mackworth, 1980, 1983) emprega um ciclo recursivo de percepção (Figura 1.3) para que os modelos em nível mais baixo na hierarquia ("schemata") sejam invocados por evidências em nível mais alto ou da imagem; os "schemata" possuem também conhecimento armazenado na forma procedural. Em um trabalho anterior (Mulder and Mackworth, 1978), empregava-se uma hierarquia que ocorria em cada nível o ciclo: descoberta de evidência, invocação de modelo, teste e elaboração; o controle foi alterado posteriormente (Havens, 1978) de modo a comportar os

"schemata". Existe uma integração entre buscas direcionadas por hipóteses ("top-down") com aquelas direcionadas por dados ("bottom-up") num modelo procedural. O ciclo de percepção leva em conta as hipóteses de que a percepção utiliza descrições armazenadas na formação da descrição do que é observado, de que é um processo não-determinístico que deve utilizar o contexto e permitir diversas interpretações parciais, de que é um processo ativo (gerado por hipóteses e expectativas) e passivo (pelas observações) e de que é um processo recursivo, em que descrições de conceitos instanciados são evidências para a percepção de conceitos em níveis mais altos.

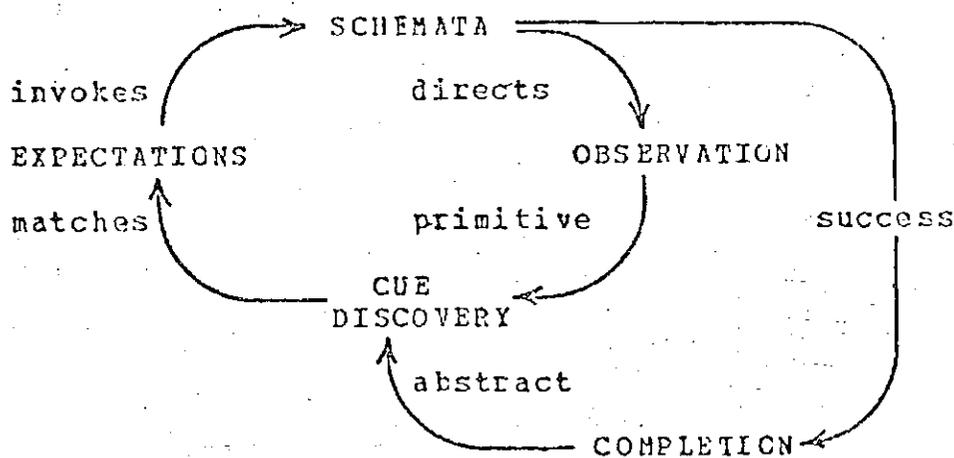


Fig. 1.3 - Esquema de análise por "schemata".

FONTE: Havens (1978, p. 148).

Segundo Fischler (1978), uma segmentação completa não é executada pelo sistema visual humano, nem é necessária para muitas aplicações, desde que não se deseje uma representação "lingüística" da cena. Propõe que a análise de uma cena, a fim de obter seu "gestalt", seja efetuada por um processo global de otimização a partir de uma representação isomórfica, que substituiria a simbólica. Deve ser observado, no entanto, que uma cena não é necessariamente semelhante

a um texto escrito, em que o todo deve ter um certo sentido e as partes ambíguas podem ser elucidadas a partir de um sentido hipotético para o todo.

Existem autores que propõem que os processos de visão em nível mais baixo ("low-level") são executados, por seres humanos, sem o envolvimento de processos cognitivos. Seriam geradas formas intermediárias de armazenamento, que não seriam simplesmente aquelas de tons cinza e de gradientes, a partir das quais haveria a atuação dos processos cognitivos.

Outras pesquisas recentes dedicam-se ao estudo de módulos do sistema visual humano, que podem operar não sobre a imagem diretamente, mas sobre representações próprias da informação (Brady, 1981). Tais pesquisas afastam-se da tentativa de construir sistemas completos para a visão computacional e podem levar ao conhecimento de quais são e como operam módulos específicos em tarefas determinadas, embora pouco se saiba sobre a maneira como eles interagem. Não se conhece bem, também, as maneiras pelas quais eles operam, quais são os processos envolvidos e se é necessário que estes processos não possuam semelhança com aqueles conjecturados para a memória.

1.4 - ALGUNS MODELOS DE REPRESENTAÇÃO DE CONHECIMENTO

Quillian (1968) desenvolveu um modelo de armazenamento do significado de palavras que consiste em nós, rotulados por uma palavra, ligados entre si por ramos associativos, podendo ser de dois tipos: símbolo (um para cada significado) e "token" (usados para elaborar o significado de outras palavras). O modelo foi usado para comparar o significado de palavras e permitiu que fosse testado como modelo de representação interna da memória humana através dos tempos de resposta a perguntas simples.

A adição de conectivos lógicos, quantificadores, descrições definidas e indefinidas, tratamento para tempo, quantificadores

vagos e operadores modais foi feita posteriormente (Schubert, 1976; Schubert et alii, 1979). As características gerais de redes semânticas seriam a representação de conceitos gerais e particulares, proposições e a possibilidade de evitar a duplicação de nós que denotam o mesmo conceito; além disto, é possível efetuar inferências segundo procedimentos heurísticos ou plausíveis. As proposições podem ser organizadas segundo tópicos (cor, tamanho, etc.) de modo a evitar o acesso direto a elas a partir de conceitos, com a finalidade de restringir processos de busca. Schubert et alii (1979) criticam também o uso da relação IS-A, que pode significar pertinência a um conjunto ("Rufus is a dog"), mas gera problemas no caso de conceitos genéricos ("dog is a mammal" - neste caso, a instância do primeiro conceito seria a instância do segundo) e predicação de segunda ordem ("blue is a color"). Na Figura 1.4a, tem-se a representação completa da proposição "John gives the book to Mary". Na Figura 1.4b, tem-se a forma simplificada para "all dogs chase same cat", traduzida para

$$\forall x [\text{dog}(x) \rightarrow \exists y [\text{cat}(y) \wedge \text{chase}(x,y)]]$$

(omitem-se os nós de proposição).

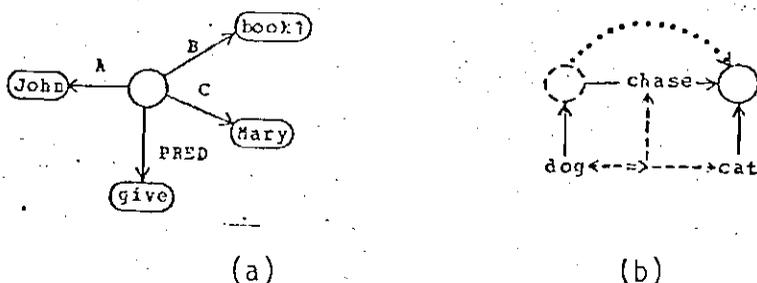


Fig. 1.4 - Proposições e quantificadores.

FONTE: Schubert (1976).

O formalismo das SI-Nets - "Structured Inheritance Networks" (Brachman, 1979; Brachman and Schmolze, 1985) permite que se expressem relações do chamado nível epistemológico como ramos numa rede semântica: estruturação de conceitos, transferências ("inheritance") e abstrações, sendo fixo o número de tipos de nós e ramos. Os conceitos repre

sentam objetos intensionais, os quais podem ser genéricos ou individuais; os termos "individualização", "instanciação" e "denotação" são usados para indicar respectivamente relações entre conceito genérico e individual, entre conceito genérico e objeto, e entre objeto e conceito individual. Cada conceito possui associado a ele os "papéis" R ("role/filler descriptions") que podem ser partes, argumentos de funções e casos de verbo, com entidade que é o valor e o papel funcional da entidade; possui associado também as relações estruturais SD, que são relações entre entidades correspondentes aos papéis. Na Figura 1.5, uma parte de ARCH é LINTEL, que deve ser WEDGE-BRICK (restrição é o predicado que uma entidade, que corresponda ao papel, deve satisfazer; o nome dá a relação conceitual entre o todo e a entidade). Existem primitivas epistemológicas para a transferência estruturada ("structured inheritance") de descrições de papéis e relações estruturais, através das quais são explicitadas as estruturas do conceito individualizado, assim como as modificações efetuadas (Figura 1.6). A linguagem KLONE permite a construção, acesso e remoção de estruturas, e suas funções trabalham somente com conceitos e relações do nível epistemológico.

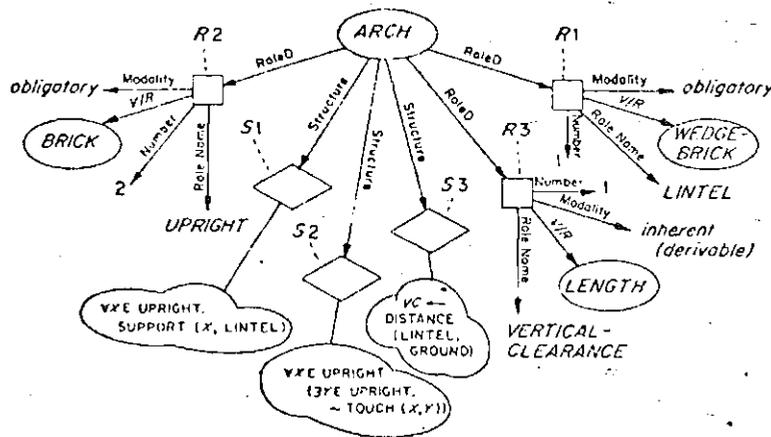


Fig. 1.5 - Papéis e relações estruturais.

FONTE: Brachman (1979).

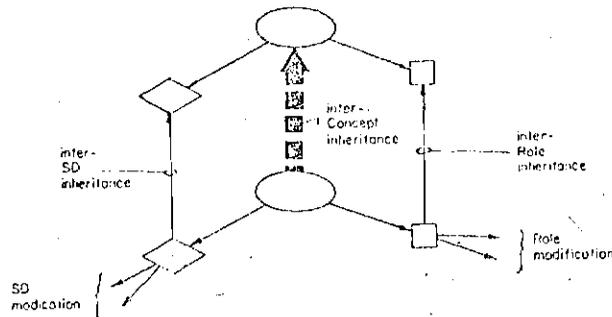


Fig. 1.6 - Transferência de papéis e relações.

FONTE: Brachman (1979).

Hendrix (1976, 1979) propõe um modelo em que são criados espaços que englobam conjuntos de nós e ramos, supernós e vistas a partir de espaços embutidos em outros; através destes, podem ser representados conectivos, quantificadores, estruturas de ordem superior e relacionados conhecimentos novos e antigos. Os ramos são relações binárias de pertinência ou de casos semânticos; a quantificação existencial é implícita e a universal obtida supondo toda variável como representante de um conjunto. Os mecanismos de dedução podem obter a transferência de propriedades a partir dos ramos de pertinência e de subconjuntos.

Uma aplicação deste modelo foi estabelecida para um sistema de prospecção geológica - PROSPECTOR (Duda et alii, 1977, 1978; Gaschnig, 1980), em que o antecedente e o conseqüente de regras de decisão são codificados como espaços. São introduzidos graus de necessidade e suficiência para que cada elemento do antecedente implique o conseqüente. As informações existentes numa aplicação propagam-se pelo uso de regras ou pelas relações de pertinência ou de subconjunto com heurísticas apropriadas.

1.5 - A ABORDAGEM PROPOSTA

Uma hipótese que é feita neste trabalho, a qual foi empregada na estruturação básica do modelo a ser exposto, é que o conhecimento armazenado a respeito de entidades conhecidas em um domínio de aplicação pode ser empregado para direcionar o processo de reconhecimento. Têm-se partes distintas do modelo para tratar de problemas de representação de conhecimento e de problemas mais voltados a medidas sobre uma imagem que representa uma cena, porém existe uma interligação dinâmica entre processos diferentes provenientes destas partes.

O modelo explora a representação de conhecimento do tipo declarativa e um ciclo de percepção já programado previamente em suas partes principais. O ciclo pode sofrer alterações pela introdução de heurísticas em diversas situações, pela utilização de comandos que dão opções de controle e pela ativação de elementos de hierarquias de conceitos.

O mecanismo de percepção a ser mostrado é bastante completo, no sentido de poder simular outros já descritos na literatura (vide Seção 4.5.2). Existe, também, facilidade para a especificação do tipo de ação que deve ser realizada.

Para o armazenamento do conhecimento disponível a respeito de um universo, existe uma linguagem através da qual as informações iniciais são introduzidas em uma estrutura tipo rede associativa. São empregados, como partes elementares das estruturas da rede, conceitos, relações de parte entre conceitos, atributos e predicados; esses dois últimos podem exigir, para o cálculo de seus valores, processos que envolvam medidas sobre uma imagem ou, então, podem ser inferidos a partir de outras entidades já existentes.

A utilização das informações fornecidas é feita através de: propagação de alterações introduzidas na rede instanciada, busca de resposta a perguntas ou busca de solução para problemas.

Através do emprego da linguagem de acesso à rede e dos comandos mencionados, é possível descrever condições que, quando ativadas, executam ações descritas por algoritmos (já propostos para tipos específicos de análise ou não). Podem ser introduzidas heurísticas para diversas finalidades, tais como: ativar partes específicas das estruturas de relação de parte, escolher que ação tomar em um dado ponto do processo, determinar se alguma ação de inicialização deve ser realizada quando não há ação possível prevista, etc. A execução de processos de medida pode ser controlada através de condições explícitas pela linguagem de acesso ou através de condições embutidas em heurísticas.

O modelo de representação reflete, parcialmente, as necessidades de uma representação declarativa associada a um processo de análise de cenas. Com este intuito, emprega-se a inferência de relações de parte, a forma generalizada de regras de decisão e predicados calculáveis. Algumas características do mecanismo de controle, como a reutilização de propriedades (quando as condições que levaram a uma primeira utilização não são mais válidas) e o encadeamento retroativo e progressivo de propriedades, também são provenientes deste tipo de necessidade.

Deve ser facilitado, com esta abordagem, o teste de novos algoritmos ou variantes de outros já existentes, embora seja possível que não se consiga uma eficiência computacional comparável à de um programa desenvolvido para um fim específico. O exemplo da Seção 4.3 mostra como simular um algoritmo para um tipo específico de operação sobre cenas.

A abordagem comporta os casos de executar processos de segmentação sobre uma imagem ou de fornecer uma segmentação inicial.

Esquemáticamente, os blocos de processamento e de armazenamento no modelo proposto são os da Figura 1.7, juntamente com os fluxos de dados envolvidos.

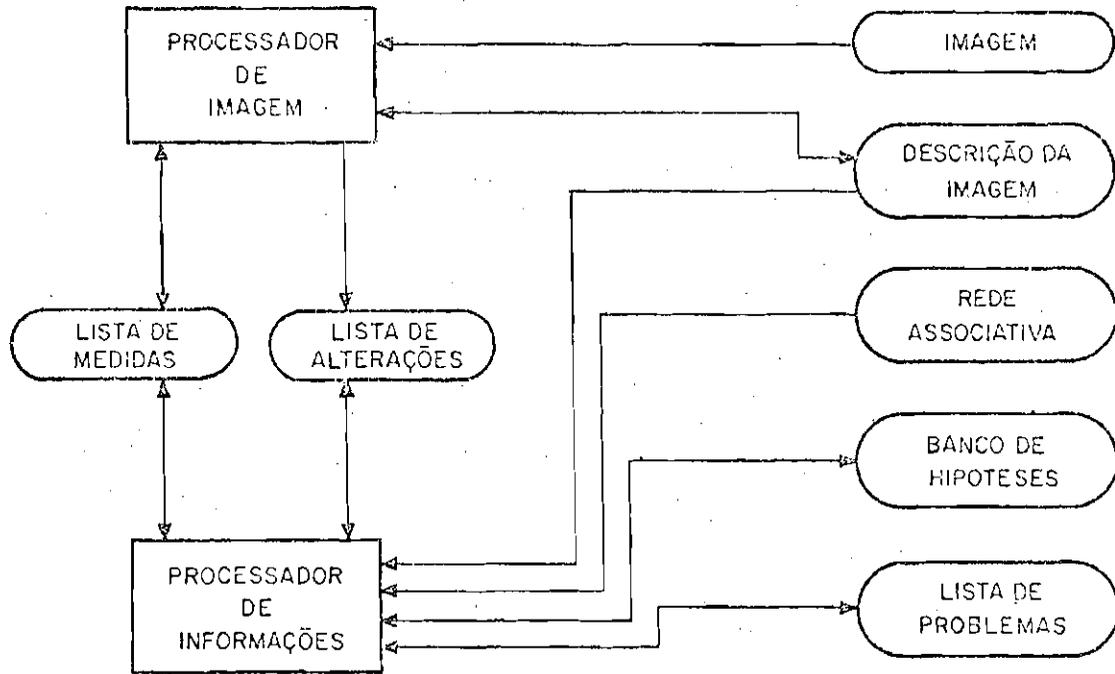


Fig. 1.7 - Esquema proposto para processo de análise.

A fim de simular a capacidade de atuação independente do processamento simbólico (realizado pelo processador de informações) e o diretamente voltado para imagens (realizado pelo processador de imagens), existe, no fluxo de controle, uma alternância de atuação de ambos; eles se comunicam através das estruturas de armazenamento de medidas a serem realizadas e da lista de alterações a propagar.

A imagem pode ser constituída por até três canais, com graus de cinza entre 0 e 225 e com tamanho 64x64. A descrição da imagem é constituída por um conjunto de áreas que indicam regiões, seus pontos extremos, bordas, interpretações, processos de formação, etc.

A rede associativa é constituída por um conjunto de áreas que indicam descrições de conceitos, hierarquias associadas a eles, proposições associadas, predicados, atributos, etc. O banco de hipóteses contém as entidades instanciadas da rede associativa.

A lista de alterações contém alterações de valores-verdade de hipóteses e predicados, introduzidas quando o banco de hipóteses é alterado pelo processador de informações, além de valores de medidas efetuadas pelo processador de imagem. Estas alterações comandam a ativação da rede associativa através de sua propagação.

A lista de medidas contém pedidos não-executados de realização de medidas sobre a imagem que podem ser gerados por um dos dois processadores.

O processador de informações é responsável pela busca de interpretações para partes da imagem, podendo acessar a descrição da imagem e rede associativa, efetuando alterações no banco de hipóteses ou pedidos de medidas, a fim de testar hipóteses feitas; efetua uma série de inferências a partir da rede associativa.

O processador de imagem pode realizar medidas ou gerar pedidos de medidas futuras a serem realizadas e é responsável pela manutencão da descrição da imagem. Emprega uma memória sobre a área onde efetuou as últimas medidas.

1.6 - ORGANIZAÇÃO DO TRABALHO

O modelo de armazenamento de conhecimento é apresentado no Capítulo 2. Ele inclui as hipóteses sobre a estruturação de conceituos, a nomenclatura adotada, as características principais das estruturaras, as condições de inferência entre estruturas, uma discussão sobre conflitos a partir de propriedades e um exemplo para uma aplicação simples.

No Capítulo 3 discute-se, em geral, o problema da utilização do conhecimento disponível sobre um universo de aplicação. Ele inclui uma descrição das estruturas que são geradas, como se acessam as propriedades para responder perguntas ou propagar dados, como se utilizam as propriedades acessadas, como são tratados os problemas (que po

dem acessar propriedades para resolvê-los e gerar outros subproblemas) e como se comporta o mecanismo geral de controle. Os algoritmos que descrevem o processo de controle são fornecidos, de forma resumida, em cada uma das seções que abordam os diversos aspectos envolvidos.

A descrição do processo de análise de cenas é feita no Capítulo 4, onde são mostradas as estruturas empregadas para a descrição de cenas, os mecanismos de geração de medidas não-controladas pela utilização de propriedades e aplicações em que se discutem as diversas características envolvidas do processo. Inclui também uma comparação do modelo de representação e do modelo de análise de cenas com outras abordagens existentes na literatura.

Os resultados obtidos são discutidos no Capítulo 5, assim como as extensões futuras ao trabalho realizado.

O Apêndice A contém uma descrição da linguagem empregada para a modificação da rede associativa, o Apêndice B contém algoritmos para a inferência de caminhos e sua prova de convergência e o Apêndice C descreve os módulos constituintes do programa que implementa o modelo proposto.

CAPÍTULO 2

ARMAZENAMENTO DE CONHECIMENTO

2.1 - INTRODUÇÃO

Serão descritas neste capítulo as características do modelo de armazenamento que foram estabelecidas de acordo com a tarefa a que o modelo se destina.

Existem modelos propostos para outras áreas de aplicação, por exemplo a da lingüística computacional. Nestes casos, é interessante prover o modelo de maneira a acomodar elementos que possam surgir na análise de uma sentença ou texto (casos semânticos, associação com redes de transição, etc.). Estes elementos não foram levados em conta no modelo sendo apresentado, embora possam ser eventualmente adicionados.

Um problema diferente daquele encontrado em aplicações puramente lingüísticas é o de determinar referentes de variáveis (universais ou existenciais), o qual envolve processos de busca sobre um banco de hipóteses relacionado com a descrição de uma cena e está relacionado com processos de medida a serem ou já realizados sobre a imagem.

Foi introduzida neste modelo a possibilidade de utilizar regras de decisão com uma sintaxe mais geral em relação àquela encontrada na literatura. Outro aspecto levantado aqui é quanto à consistência da estrutura e, relacionado com este, o aspecto da transferência (inferência de outras relações hierárquicas a partir das existentes), que pode levar a uma representação menos extensa. O problema de inferência é tratado diretamente a partir de processos de busca associados a variáveis sem utilizar processos de prova automática.

2.2 - DESCRIÇÃO DO MODELO

2.2.1 - PROPRIEDADES E CONCEITOS

A forma de representação de conhecimento que é empregada neste trabalho é do tipo rede associativa, tendo por base a estruturação e referências de conceitos. A noção de conceito é de natureza vaga, podendo designar atributos, propriedades, etc., desde que com o conteúdo cognitivo. Supõe-se que o conhecimento disponível a respeito de conceitos está associado à representação pelas proposições nas quais eles participam.

Supõe-se, também, que é possível estruturar os conceitos envolvidos em um universo de aplicação de acordo com hierarquias de partes e de particularizações. Existem algumas condições de consistência a serem impostas pela natureza das hierarquias e pela natureza das aplicações.

Parte da formulação a ser apresentada leva em conta os trabalhos de Schubert et alii (1979), Brachman (1979) e Rieger (1975).

Basicamente, o modelo de representação emprega:

- a) Lista de conceitos, com acesso a hierarquias de partes e de particularizações, e lista de referências de usos em propriedades; pode possuir também indicação de ações a tomar quando o conceito é ativado para algum referente (por exemplo, ativar conceitos das hierarquias de partes e de particularizações).
- b) Lista de predicados, com acesso à lista de referências de usos, seu tipo (não-calculável, calculável geral, relação geométrica), número de código que permite calcular seu valor-verdade a partir de argumentos conhecidos, indicação se o cálculo do valor envolve a realização de medida sobre a imagem e ordem do

primeiro argumento que pode ser desconhecido, no caso de ser calculável, o qual será também calculado como resultado colateral.

- c) Lista de atributos, com acesso à lista de referências de usos, tipo de valor (valor numérico, intervalo ou lista), número de código que permite calcular seu valor, indicação se o cálculo envolve a realização de medidas, indicação de deve ser calculado automaticamente para cada nova região e ponteiro para eventual conceito correspondente.
- d) Lista de propriedades, montadas na forma de fórmulas quantificadas, com acesso a listas de referências de uso.

Os detalhes do modelo referem-se ao conhecimento que é fornecido a priori, antes de uma aplicação. Durante uma aplicação, é montada uma outra estrutura, com muitos aspectos semelhantes ao desta, porém com os elementos identificados ou sobre os quais se fazem hipóteses (vide Seção 3.2). Pode haver mais de uma instanciação de um mesmo conceito, mais de uma hipótese sobre um mesmo elemento ou mais de uma aplicação de uma mesma propriedade, com referentes diferentes para as variáveis ou elementos diferentes mapeados aos nós da propriedade.

O termo propriedade empregado neste trabalho tem um significado diferente daquele que aparece em muitos textos de lógica (Carnap, 1970). Por exemplo, encontram-se nestes textos termos como propriedades qualitativas ("azul"), quantitativas ("pesa 3 kilos") ou relacionais ("tio de alguém"). Neste trabalho ele é empregado para fórmulas sem variáveis livres ou regras de decisão.

Não se permite, na implementação do modelo de representação, o compartilhamento de parcelas de uma propriedade com outras propriedades, devido a dificuldades de atualização da lista de propriedades e do uso comum de variáveis. O efeito desejável do compartilhamento, de introduzir alterações nas hipóteses formuladas, sempre que uma

das propriedades é utilizada, é obtido pela propagação de valores alterados.

As propriedades são tais que a seus nós podem ser associados valores-verdade pelo uso de predicados (vindos ou não de conceitos) e operadores.

Pode-se considerar que cada propriedade (e não cada nó isoladamente) representa um conceito. Por exemplo, pode-se ter o conceito "se existirem nuvens escuras no céu, então vai chover". O fato de dizer que existem propriedades associadas a conceitos vem da possibilidade de os nós de cada propriedade se referirem a conceitos gerais, com seu lugar na lista de conceitos e nas hierarquias e, também, com nomes que permitam seu uso em outras partes da rede.

A introdução de regras de decisão como propriedades torna-se, então, de certa forma natural. A cada regra está associado um conceito, que pode ser oriundo de conhecimento especializado sobre uma área.

O poder de inferência da representação depende da capacidade de expressar os conceitos objetivamente e de forma não conflitante. Algumas regras de consistência podem ser usadas para evitar a aceitação de certas estruturas. Não se realiza nenhuma tentativa para verificar se as propriedades são conflitantes; mecanismos de prova automática de teoremas poderiam ser empregados desde que não se permitisse a presença proposital de propriedades conflitantes na rede (uma maneira de tratar casos de fragmentação excessiva da hierarquia de particularizações consiste em permitir que as propriedades que valham para um conceito num nível mais alto na hierarquia não valham necessariamente para os que estão num nível inferior).

Outro aspecto da representação adotada é que ela deve permitir a existência de dois conceitos para representar o mesmo objeto do universo, quando se têm duas intensões para o mesmo objeto; o fato de a extensão ser a mesma é representado por uma propriedade.

2.2.2 - HIERARQUIA DE PARTES

Esta hierarquia reflete a capacidade de expressar um conceito em função de outros; utiliza-se a relação estrutural PARTE-DE para a introdução de partes.

Dado um conceito, ele pode ser parte de mais de um conceito. Por exemplo, CORAÇÃO é parte de TÓRAX e também de SISTEMA-CIRCULATORIO. A hierarquia de partes possibilita a associação de diversos tipos de entidades a conceitos; a noção de parte, aqui empregada, tem relação com a de papel ("role"), mostrada por Brachman (1979), podendo designar partes conceituais, atributos (cor, tamanho, etc.), casos verbais, etc.

A estrutura de partes é considerada estática, a menos das condições de transferência (vide Seção 2.3). Verifica-se, a partir da entrada de uma relação, se os conceitos envolvidos estão definidos e se são satisfeitas algumas condições de consistência.

Considera-se, também, que as relações de parte não são afetadas, na sua definição, por graus subjetivos de crença. Desta forma, a relação "contexto para", que surge no trabalho de Preparata e Ray (1972) e que tem interpretação semelhante à da relação de partes no presente modelo, não pode ser expressa. Outras restrições, tais como o número de instâncias (ou sua não-existência), são colocadas como propriedades.

Além disso, quando os conceitos são especificados como partes de outros, propriedades adicionais podem ser adicionadas para o uso particular como parte; alternativamente, podem ser criadas particularizações com as novas propriedades, sendo estas particularizações colocadas como parte. Por exemplo, se um conceito JANELA tem algumas propriedades genéricas associadas (permite passagem de luz, etc.) e ele for colocado como parte de CASA, outras propriedades serão aplicáveis (por exemplo, a forma obedece a alguns padrões usuais); alternativamente,

pode ser criado JANELA-DE-CASA, particularização de JANELA com estas propriedades adicionais (algumas das quais podem até ser conflitantes com as de JANELA).

2.2.3 - HIERARQUIA DE PARTICULARIZAÇÕES

Esta hierarquia reflete a capacidade de dizer que um conceito é caso particular de outro; utiliza-se a relação estrutural PARTICULARIZAÇÃO-DE. Faz-se a restrição que um conceito pode ser particularização de somente um outro (esta restrição facilita a transferência de partes entre conceitos e a análise de compatibilidade entre valores); segundo esta restrição, não se pode representar explicitamente na hierarquia a transitividade da relação de particularização.

Pode-se ter conceitos genéricos ou individuais neste modelo; o fato de um conceito não possuir particularizações não é suficiente para considerá-lo como individual. A própria interpretação da relação de particularização depende dos tipos de conceitos envolvidos; Schubert et alii (1979) observam que se os dois conceitos envolvidos são genéricos, então toda instância do mais específico é também uma instância do mais geral ("instância" utilizada no sentido de conceito individual).

Esta estrutura é considerada como estática; após a entrada de uma relação, verifica-se se os conceitos envolvidos estão definidos e se são satisfeitas algumas condições de consistência.

2.2.4 - INSTANCIACÃO DE CONCEITOS

O domínio de aplicação que serviu de motivação para o desenvolvimento do presente modelo foi o de análise de cenas. Nesse caso, existe a necessidade de efetuar um mapeamento entre elementos específicos de uma cena e conceitos, de modo a permitir que se estabeleçam hipóteses sobre as entidades representadas pictoricamente pelos elementos. Estas entidades serão consideradas como instâncias de conceitos

e referidas pelos seus descritores na cena. É possível atribuir um referente fictício a uma instância, para o caso de conceitos que não possuam ocorrência, ou aqueles para os quais somente suas partes sejam identificadas.

Em outras aplicações, principalmente lingüísticas, o termo instância muitas vezes se refere a conceitos individuais ("João", "Rio de Janeiro", etc.).

2.2.5 - LISTA DE PROPRIEDADES

A lista de propriedades abrange a caracterização de partes, de particularizações ou de predicados em geral, fatos associados a conceitos, conhecimentos empíricos, etc.

Para a introdução de propriedades, emprega-se uma linguagem cuja sintaxe é semelhante à do cálculo de predicados de primeira ordem (vide Apêndice A); todas as variáveis devem ser quantificadas. As propriedades são formadas por operadores lógicos, operador que indica regra de decisão, atributos (associados a conceitos ou não), valores, intervalos numéricos e predicados (conceitos ou relações de parte usados predicativamente, predicado não-calculável, predicado calculável geral ou de relação geométrica). O operador unário NÃO sõ pode ser aplicado diretamente a um predicado (o que não altera o poder de expressão e simplifica alguns algoritmos). Os quantificadores disponíveis são o universal e o existencial; o numérico não foi completamente implementado (vide Seção 5.2).

No exemplo da Figura 2.1, tem-se a forma como se estrutura internamente a propriedade e as relações dadas. O conceito CASA é usado como predicado unário; os conceitos JANELA e TELHADO são usados no sentido de existir um encadeamento de relações de parte entre instâncias de CASA E JANELA (ou TELHADO) e de existir instância de JANELA (ou TELHADO); neste exemplo existe tal encadeamento, se existir instân

cia de PAVIMENTO. O predicado ACIMA pode ser considerado como do tipo relação geométrica, no caso de uma aplicação em análise de cenas.

PAVIMENTO PARTE-DE CASA

JANELA PARTE-DE PAVIMENTO

TELHADO PARTE-DE CASA

$\forall x \forall y \forall z [[CASA(x) \text{ E } JANELA(x,y) \text{ E } TELHADO(x,z)] \text{ ----> } ACIMA(z,y)]$

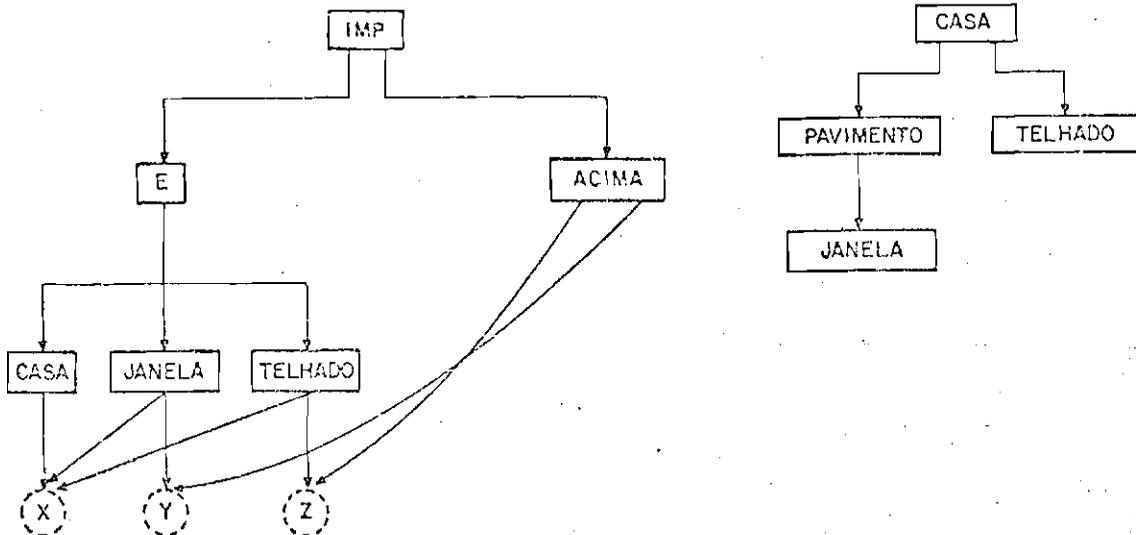


Fig. 2.1 - Estruturas de propriedade e de hierarquia de partes.

Regras de decisão têm, neste modelo, a forma geral a seguir, onde Q_x designa um conjunto de quantificadores, sendo o primeiro universal, o real c (grau de certeza) pertence ao intervalo $(0,1]$ e <antecedente> e <conseqüente> designam fórmulas, cada qual com um conjunto próprio de quantificadores e com os quantificadores do conjunto Q_x compartilhados entre ambas:

(c)
Qx [<antecedente> ----> <conseqüente>]

Não existe análise de consistência associada às propriedades; cada uma delas é uma parcela independente de conhecimento na rede. Eventualmente poderão existir, propositalmente, propriedades conflitantes na rede.

Não existe, também, maneira de explicitar hierarquias de propriedades, ou seja, de dizer que algumas são mais gerais que outras; um esquema proposto é o de Schubert et alii (1979), em que as propriedades são classificadas segundo a hierarquia de tópicos e conceitos envolvidos. Reder e Anderson (1979) comentam as evidências de que um tipo de organização, em que os fatos relacionados a um conceito estão aglutinados, está presente na memória humana.

Para propriedades em que ocorrem variáveis quantificadas existencialmente, monta-se a função de Skolem correspondente a cada uma delas (com as variáveis universais que são argumento da função). Durante uma aplicação, são guardadas as instâncias observadas da função, as quais servem como indicação inicial no processo de busca realizado para a determinação do referente.

2.3 - CONDIÇÕES DE TRANSFERÊNCIA E CONSISTÊNCIA DE ESTRUTURAS

Existe um aspecto de compatibilidade entre estruturas novas e já existentes que deve ser observado. Para isso, seria necessário uma especificação completa das estruturas de conceitos envolvidas, o que nem sempre é possível; pode não ser praticável, também, a análise de todas as situações que envolvam incompatibilidade.

Os procedimentos adotados neste modelo pressupõem que foram realizados diversos testes relevantes ao se entrar com uma informação nova. As estruturas são armazenadas da maneira como são fornecidas. Durante uma aplicação, pode ocorrer a transferência de partes a

elementos instanciados das hierarquias segundo algumas condições que se supõe apropriadas para a transferência. Deve ser observado, no entanto, que não se dispõe de uma teoria mais geral sobre a natureza dos conhecimentos envolvidos, que poderia ser usada para saber em que circunstâncias tais transferências deveriam ou poderiam ser realizadas.

Dados os conceitos A, B, C, as combinações possíveis de relações de parte e de particularização são as da Figura 2.2 (tomando uma só relação entre cada par de conceitos).

Os casos permitidos são (a)-(f); nos casos (h) e (l) existe mais de uma generalização para um conceito; nos casos (g), (i), (j) e (k) ocorre um ciclo (na versão atual do programa desenvolvido, a verificação de ciclos que envolvam mais de um tipo de relação não foi implementada para a fase de entrada de elementos das hierarquias, sendo, no entanto, prevista em extensão posterior).

De uma maneira geral, existem casos em que pode haver a transferência de relações de parte (independentemente do que é especificado pelas propriedades relacionadas) e outros em que não pode haver. Alguns casos são os descritos a seguir (Figura 2.3).

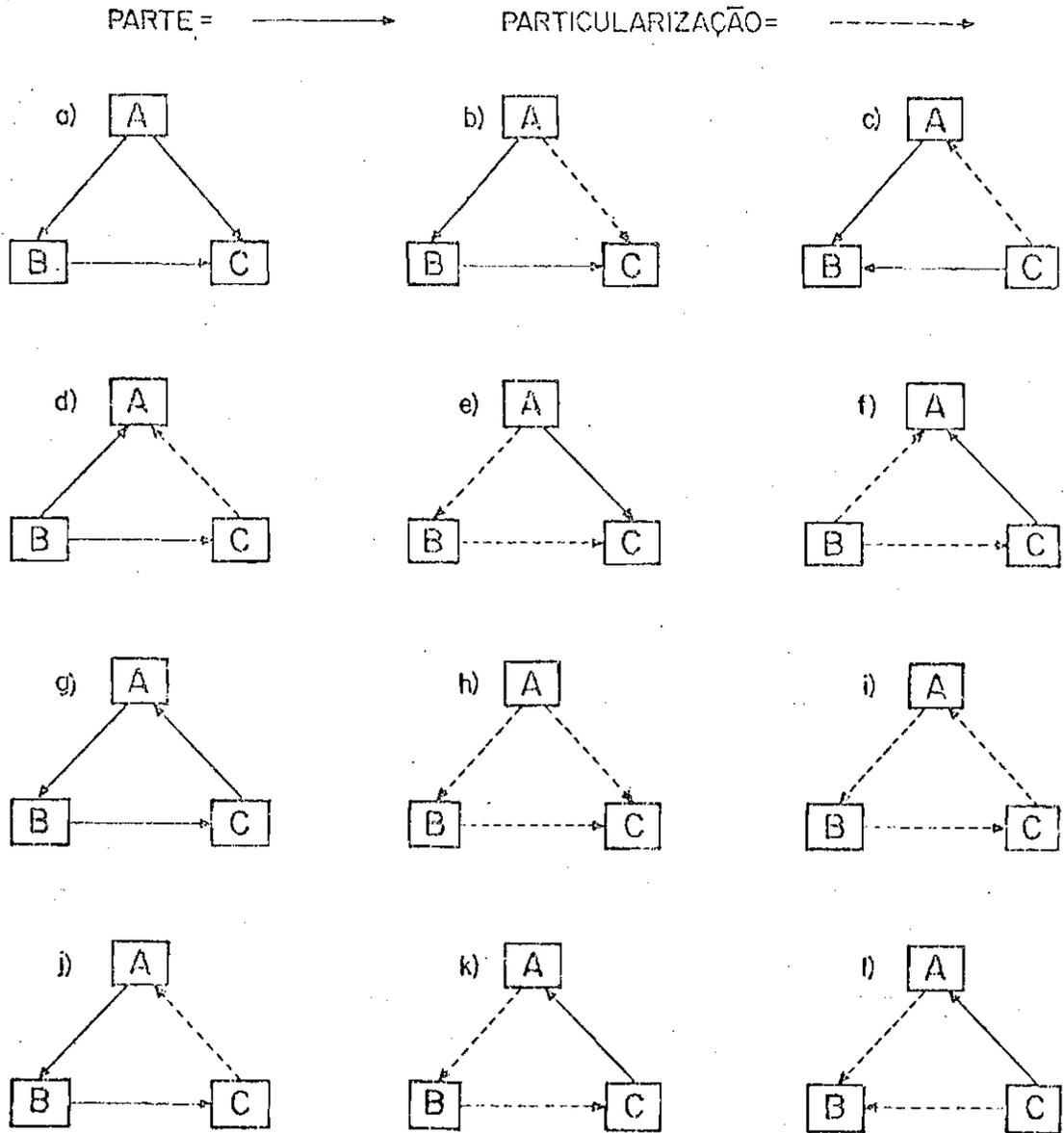


Fig. 2.2 - Combinações possíveis de relações.

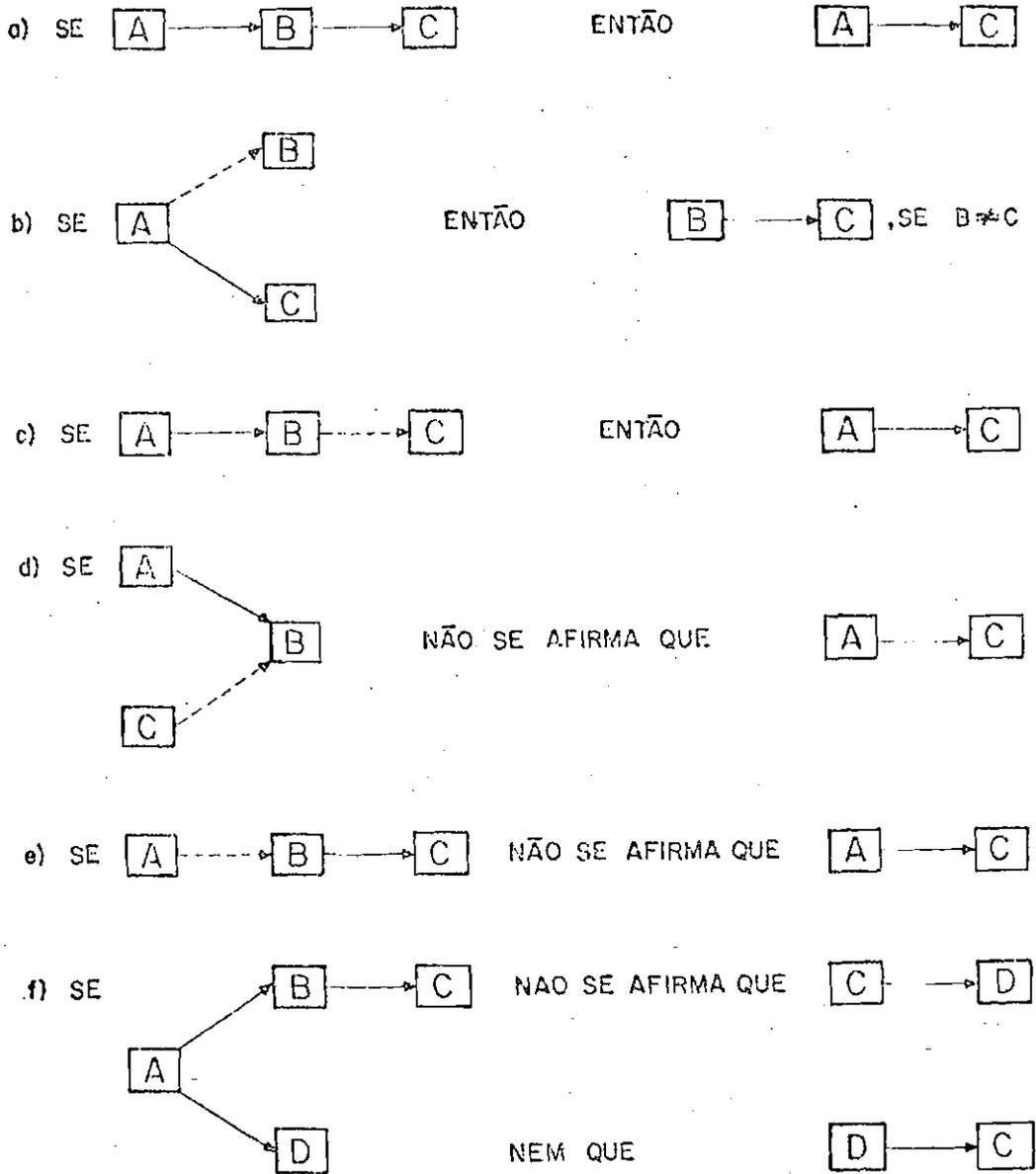


Fig. 2.3 - Casos de transferência de relação de parte.

Alguns exemplos são dados a seguir, para os casos em se efetua a transferência.

- a) se CABEÇA PARTE-DE CORPO e
 BOCA PARTE-DE CABEÇA ,

então BOCA PARTE-DE CORPO

b) se MAMÍFERO PARTICULARIZAÇÃO-DE ANIMAL e
CORPO PARTE-DE ANIMAL ,

então CORPO PARTE-DE MAMÍFERO

c) se DEDO PARTE-DE CORPO e
DEDO-PÉ PARTICULARIZAÇÃO-DE DEDO ,

então DEDO-PÉ PARTE-DE CORPO

É possível que, de acordo com as relações de parte e de particularização armazenadas nas hierarquias, possa ocorrer a transferência de uma relação de parte e de sua inversa também (este problema poderia ser evitado se, na entrada de uma relação, já fosse verificada a existência de ciclos que envolvessem todas as relações possíveis de ser inferidas). Isto pode ocorrer, por exemplo, pela aplicação do caso (b) de transferência por duas vezes, nas situações da Figura 2.4

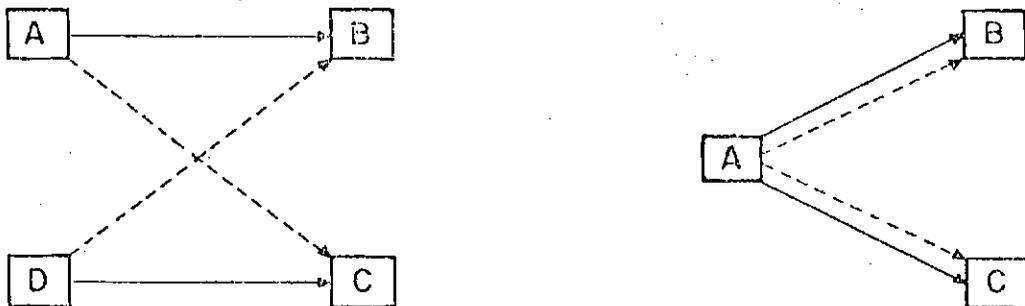


Fig. 2.4 - Transferência de relação e inversa.

A aplicação das situações (c) por duas vezes, ou de (b) juntamente com (c) uma vez não resulta em conflito, desde que não haja nenhum ciclo formado por relações explicitamente fornecidas de ambas hierarquias. Optou-se por não permitir a transferência na situação (d) para haver a possibilidade de transferir o máximo possível de relações de parte, através da situação (b). Além disso, é possível a ocorrência de conflitos empregando (d) duas vezes, (b) e (d) com pelo menos quatro nós e (c) e (d) com pelo menos quatro nós; existe, assim, potencialmente, mais casos de conflito com o uso de (d). A situação (e) foi evitada devido a diversos exemplos em que a transferência não é satisfatória para diversos universos de aplicação.

Nos casos em que se pode gerar uma nova relação de partes, deve-se evitar a formação de ciclos que envolvam as novas relações. Em princípio, deve prevalecer a relação de parte originalmente fornecida, que inibe, em todos os casos, a adição de relações inversas. A Figura 2.5 exibe um caso em que ocorre tal inibição; não se pode inferir a relação B PARTE-DE C a partir da situação (b) (a ausência de C PARTE-DE B possibilitaria esta adição).

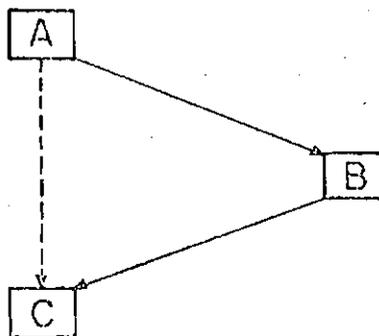


Fig. 2.5 - Inibição de transferência de relação.

Na implementação atual, as condições de transferência não podem ser aplicadas diretamente às hierarquias; elas somente são usadas quando se tem uma propriedade instanciada e se quer determinar a existência de caminhos, através de relações, de parte e de conceitos,

entre um conceito referenciado por um \bar{n} da propriedade (ou um \bar{n} que representa uma relação de parte) e outro referenciado através do \bar{n} que representa uma relação (o algoritmo empregado para a transferência é mostrado no Apêndice B).

2.4 - INCONSISTÊNCIA ENTRE PROPRIEDADES

Existem situações em que as propriedades, juntamente com a hierarquia de particularizações, não são livres de incompatibilidades. Em geral, se A e B são conceitos, sendo B uma particularização de A, podem existir propriedades associadas a A que não valem no caso particular de B; no presente modelo, este fato é assinalado por uma propriedade que se refere explicitamente a B.

Esta maneira de tratar o problema de representar generalizações, quando alguns de seus casos particulares representam uma exceção, evita uma pulverização de estruturas. Além disso, possibilita representar conceitos que, de outra forma, seriam particularizações de mais de um conceito mais geral, devido a uma eventual separação do geral. Da maneira empregada, coexistem as propriedades que podem levar a conflitos; durante a fase de aplicação, devem ser observados critérios para a análise de compatibilidade.

A área de ciências naturais é uma em que freqüentemente surgem tais problemas: na organização de animais ou plantas segundo classes, uma propriedade que é associada a todos os elementos de uma classe encontra casos particulares em que não vale. À parte do problema de gerar classificações, uma maneira de tratar o problema de representação é de forma que não dependa dos motivos que levam à classificação; de outra forma, deveria haver um mecanismo que tratasse os tipos possíveis de exceção, para os tipos possíveis de regras de classificação.

Seja o seguinte exemplo:

1) PINGÜIM PARTICULARIZAÇÃO-DE PÁSSARO

2) $\forall x$ [PÁSSARO(x) \rightarrow VOA(x)]

3) $\forall x$ [PINGÜIM(x) \rightarrow NÃO VOA(x)]

Se para o referente X tem-se que PINGÜIM(X) é verdadeiro, então: PÁSSARO(X) é verdadeiro (por (1)), VOA(X) é falso (por (3)) e VOA(X) é verdadeiro (por (2)).

Na fase de utilização de propriedades para uma aplicação específica, sempre que ocorrer um conflito de valores, deve-se verificar se um dos valores permitiu a atribuição de outro que gerou o valor conflitante por meio de uma relação da hierarquia de particularizações (no exemplo, através de PINGÜIM(X) verdadeiro concluiu-se PÁSSARO(X) verdadeiro). Faz-se a hipótese que o conflito pode ser resolvido supondo válido o resultado associado à particularização. Eventualmente, podem existir nós intermediários entre um nó de um conceito e um de uma particularização. Uma outra hipótese que é feita é que, ao usar o valor-verdade associado a uma hipótese como dado, deve ser observado se, para o mesmo referente, já existe alguma particularização já ativada como o valor-verdade oposto.

Estas considerações mostram, para a forma proposta para o problema de representação, parte das razões pelas quais é necessário associar aos valores-verdade obtidos a sua origem.

Outra maneira de contornar o problema do conflito é acrescentar, nas propriedades, condições que evitem a existência de conflito. Isto traz, no entanto, o inconveniente de as propriedades poderem se tornar muito extensas para acomodar todas as combinações possíveis de exceção. Além disso, a cada alteração das propriedades ou hierar

quias deve-se verificar todas as relações de consistência já estabelecidas. No exemplo visto, poder-se-ia ter

2-a) $\forall x [[PÁSSARO(x) \text{ E NÃO PINGÜIM}(x)] \rightarrow VOA(x)]$.

Ainda com relação à análise de compatibilidade de valores, se não for detectada a existência de propriedades conflitantes, pode ser feita a combinação dos diversos valores obtidos segundo algum critério ou teoria ou, então, tomar o valor como desconhecido.

2.5 - UM EXEMPLO DE ESTRUTURAS DE ARMAZENAMENTO

Para o caso específico de conceitos e propriedades relacionadas com figuras geométricas, serão tomados, neste exemplo, aqueles que dizem respeito às estruturas mencionadas anteriormente. A representação destas entidades envolve problemas do que tornar explícito e como. Alguns conceitos serão tomados como primitivos e outros serão obtidos através destes, de acordo com convenções que serão estabelecidas.

Será convencionado, inicialmente, que instâncias de figuras geométricas são conjuntos de pontos nos quais o contorno não está incluído; no entanto, o conceito FIGURA-GEOMÉTRICA tem como parte conceitual CONTORNO. O conceito FIGURA-GEOMÉTRICA engloba uma série de outros conceitos, como polígonos regulares, polígonos convexos, etc, sendo definido em termos de outros mais elementares (pode ser de maneira não-única, dependendo dos primitivos escolhidos e suas propriedades). Supõe-se implícito um sistema de coordenadas, que é utilizado em medidas como ÁREA e COMPACTAÇÃO. As medidas estão associadas, implicitamente, às particularizações de FIGURA-GEOMÉTRICA. Suponha-se, inicialmente, que a hierarquia fornecida como entrada é dada pelas relações 1-22 a seguir:

1) NÃO-POLÍGONO	PARTICULARIZAÇÃO-DE FIGURA-GEOMÉTRICA
2) CÍRCULO	PARTICULARIZAÇÃO-DE NÃO-POLÍGONO
3) POLÍGONO	PARTICULARIZAÇÃO-DE FIGURA-GEOMÉTRICA
4) POLÍGONO-NÃO-RETILÍNEO	PARTICULARIZAÇÃO-DE POLÍGONO
5) POLÍGONO-RETILÍNEO	PARTICULARIZAÇÃO-DE POLÍGONO
6) TRIÂNGULO	PARTICULARIZAÇÃO-DE POLÍGONO-RETILÍNEO
7) SEGMENTO-DE-CURVA	PARTICULARIZAÇÃO-DE CURVA
8) CURVA FECHADA	PARTICULARIZAÇÃO-DE CURVA
9) CONTORNO	PARTICULARIZAÇÃO-DE CURVA-FECHADA
10) SEGMENTO	PARTICULARIZAÇÃO-DE SEGMENTO-CURVA
11) PONTO-DE-CURVA	PARTICULARIZAÇÃO-DE PONTO
12) PONTO-EXTREMO	PARTICULARIZAÇÃO-DE PONTO-DE-CURVA
13) ÁREA	PARTE-DE FIGURA-GEOMÉTRICA
14) COMPACTAÇÃO	PARTE-DE FIGURA-GEOMÉTRICA
15) CONTORNO	PARTE-DE FIGURA-GEOMÉTRICA
16) COMPRIMENTO	PARTE-DE CURVA
17) VÉRTICE	PARTE-DE CURVA
18) PONTO-DE-CURVA	PARTE-DE CURVA
19) COMPRIMENTO	PARTE-DE CURVA-FECHADA
20) ÂNGULO	PARTE-DE VÉRTICE
21) SEGMENTO-DE CURVA	PARTE-DE VÉRTICE
22) RAI0-DE-CURVATURA	PARTE-DE PONTO-DE-CURVA

Antes de comentar os resultados da aplicação das condições definidas para a transferência de estruturas, serão dadas algumas justificativas da forma como a hierarquia foi elaborada, neste domínio em particular.

Quanto à hierarquia de particularizações, pode-se ter uma divisão entre POLÍGONO e NÃO-POLÍGONO ou, alternativamente, entre FIGURA-CONVEXA e FIGURA-NÃO-CONVEXA, conforme o critério (convexidade ou existência de vértice). Não seria interessante manter as duas divisões, pois se for necessário definir POLÍGONO-REGULAR, este seria uma particularização tanto de POLÍGONO como de FIGURA-CONVEXA. Se for necessária uma referência a todas as suas divisões, pode-se primeiro in

troduzir uma e, para cada particularização resultante, a outra (POLÍGONO teria como particularizações POLÍGONO-CONVEXO e POLÍGONO-NÃO-CONVEXO; a CONVEXIDADE seria aplicável tanto a POLÍGONO quanto a NÃO-POLÍGONO).

Outro conceito envolvido, o de CONTORNO, pode ser explicitado em termos de um mais geral, o de CURVA. O conceito CURVA possui um tipo particular de ponto como parte: PONTO-DE-CURVA é o nome dado a pontos que formem um feixe de "espessura" infinitesimal. A PONTO-DE-CURVA pode ser associado RAIOS-DE-CURVATURA, mas não a PONTO-EXTREMO, uma de suas particularizações (esta restrição deve ser fornecida através de uma propriedade). Implicitamente, PONTO-EXTREMO é parte de CURVA.

CURVA pode ter a particularização CURVA-FECHADA (uma curva com diversos laços é considerada como aberta, se existir percurso sem volta ao ponto inicial), a qual não possui PONTO-EXTREMO como parte. Além disso, pode ser definida explicitamente na rede ou, então, ter um predicado associado que informe, para cada instância de CURVA, se é fechada ou não.

No caso de CONTORNO, deve haver uma restrição adicional, como deixar à direita (por exemplo) a instância correspondente de FIGURA-GEOMÉTRICA, além de formar somente um laço.

SEGMENTO-CURVA é uma particularização de CURVA, sendo aberta, sem laços nem vértices; compõe o CONTORNO de POLÍGONO.

Outra restrição que pode ser levantada é que CONTORNO possui a parte VÉRTICE sô quando CONTORNO é parte de POLÍGONO; uma forma de especificá-la seria:

*x *y [[NÃO-POLÍGONO(x) E CONTORNO(x,y)] ---->

*z [NÃO VÉRTICE(y,z)]]

De forma semelhante, outras propriedades são especificadas: quando VÉRTICE é parte de POLÍGONO-RETILÍNEO, em vez de SEGMENTO-DE-CURVA tem-se SEGMENTO; no caso de POLÍGONO-RETILÍNEO, SEGMENTO-DE-CURVA é SEGMENTO; no caso de CÍRCULO, uma propriedade de definição é a que afirma ser um NÃO-POLÍGONO em que o RAI0-DE-CURVATURA do contorno é constante.

Voltando agora ao problema da transferência de estruturas, se as hierarquias forem exatamente aquelas exibidas no início desta seção, irá ocorrer um conflito (uma propriedade que permite sua descoberta é, por exemplo, aquela que diz que CONTORNO só possui a parte VÉRTICE quando CONTORNO é parte de POLÍGONO):

- 23) SEGMENTO PARTICULARIZAÇÃO-DE CURVA (de (7) e (10))
- 24) VÉRTICE PARTE-DE SEGMENTO (de (17), (23) e condição (b))
- 25) SEGMENTO PARTE-DE VÉRTICE (de (10), (21) e condição (c)).

Este conflito pode ser evitado adicionando a relação a seguir que inibe a transferência que ocasiona (25):

- 26) SEGMENTO PARTE-DE VÉRTICE.

Uma alternativa à forma como a hierarquia deste exemplo foi dada seria indicar, para cada conceito, todas as partes possíveis. A forma apresentada tem a vantagem de supor implícita uma série de relações, de diminuir a área estática de armazenamento e de só expandir o grafo à medida do necessário. Por outro lado, a forma exposta torna mais complexo o processamento das hierarquias.

CAPÍTULO 3

MECANISMO DE CONTROLE

3.1 - INTRODUÇÃO

O mecanismo de controle aqui descrito reflete a motivação da aplicação do modelo ao domínio da análise de cenas. A aplicação a outros domínios é também viável, como será comentado na Seção 4.5.1.

Existe um conjunto de comandos de controle, destinados a oferecer opções para as ações a serem seguidas, em diversos pontos; alguns destes terão suas ações comentadas ao longo do trabalho. Dentre o conjunto de opções, existem aquelas para a consideração de todas as alterações a serem propagadas de uma só vez, para o armazenamento de valores encontrados de funções de Skolem, para a criação de elementos das hierarquias de um conceito instanciado, para a especificação da fração do tempo total de processamento destinado à realização de medidas solicitadas ou espontâneas, para a definição dos locais onde deve haver interrupção, para a geração de subproblemas, para as ações pré-programadas que podem ser escolhidas quando não há ação possível a executar, etc.

O mecanismo de controle atua basicamente na forma de propagar alterações introduzidas na parte instanciada da rede, resolver problemas estabelecidos na forma de perguntas e realizar medidas. Utiliza-se um mecanismo de acesso a propriedades, que efetua um mapeamento entre elementos especificados para o acesso e nós de propriedade; os mapeamentos obtidos são utilizados na ação especificada e os eventuais resultados obtidos são inseridos na parte instanciada da rede.

Para a utilização de uma propriedade, uma série de passos deve ser executada, sendo descritos nas próximas seções. Logo após a entrada de uma propriedade, por um comando da linguagem de acesso à rede, determina-se, para cada nó, quais variáveis precisam ter seu refe

rente determinado, para que a expressão que o \bar{n} representa seja satisfeita, procurando minimizar o número de variáveis em cada \bar{n} . Para a utilização da propriedade, efetua-se o mapeamento entre elementos especificados, associando, se possível, um referente a variáveis universais ou estabelecendo que os referentes serão determinados durante a utilização. Quando uma propriedade possui \bar{n} s de relação de parte, é necessário determinar caminhos entre \bar{n} s de conceito e de parte expandindo a hierarquia de partes. As transferências possíveis entre valores-verdade de \bar{n} s de cada propriedade mapeada são determinados, podendo-se rejeitar uma aplicação se não for possível, por exemplo, responder a uma pergunta que gerou o acesso. Durante o percorrimento para a utilização, são efetuadas buscas na parte instanciada das hierarquias e da lista de predicados de modo a determinar referentes para as variáveis sem referente associado.

Podem ser necessários a criação de elementos das hierarquias associadas a um conceito que foi instanciado, a análise da compatibilidade de valores novos com os já existentes previamente, a análise de valores-verdade intermediários entre V e F (graus de certeza), a análise de variáveis existenciais cujo referente não foi encontrado e a propagação de solução de problemas ou geração de subproblemas entre as ações que devem ser executadas após a utilização de propriedades.

Uma observação quanto à linguagem de acesso à rede é que ela não permite que se especifique atribuição de valor-verdade a um predicado ou uma hipótese. Eventuais atribuições de valores irão depender da maneira como as propriedades são acessadas. Uma exceção é dada pelas regras de decisão que permitem a atribuição de valores aos \bar{n} s da conclusão que não sejam \bar{n} s de predicado calculável; é possível ter regras de decisão em que na conclusão inverte-se o valor-verdade de um elemento da premissa (vide Seção 4.3).

A utilização de propriedades pode resultar tanto na verificação se as informações disponíveis são consistentes de acordo com uma propriedade específica, como na atribuição de valores-verdade ou

criação de referentes fictícios para as variáveis. As buscas efetuadas sobre elementos já ativados, ocasionadas pelas variáveis, podem ter seus resultados dependentes do instante em que são realizadas; em certos casos, pode ser necessária uma reutilização de uma propriedade quando são alterados elementos da parte instanciada da rede.

Deve ser observado também que, num modelo de reconhecimento, podem ser estabelecidas hipóteses tentativamente a partir de evidências parciais observadas e, mesmo que o conjunto de propriedades seja destituído de inconsistências, há possibilidade de conflitos entre valores obtidos.

Além da verificação de conseqüências de informações fornecidas, existem situações em que se quer responder a alguma pergunta (também na forma de fórmula atômica, em que os referentes associados ao elemento de acesso podem ser conhecidos ou não; podem existir, também, dados associados à pergunta), fazendo uso de uma propriedade e outras informações conhecidas. Se o elemento sobre o qual se faz a pergunta já está ativado com valor-verdade conhecido, deve ser comparada a resposta com o valor antigo, desde que este valor antigo não seja proveniente da aplicação da mesma propriedade.

3.2 - ESTRUTURAS DO BANCO DE HIPÓTESES

Durante o processo em que são utilizados os conhecimentos armazenados, é necessário manter explícito que partes das estruturas foram efetivamente instanciadas, quais os valores e valores-verdade associados e as razões pelas quais foram atribuídos os valores. Dentre as principais estruturas, que serão parcialmente descritas a seguir, algumas delas têm semelhança com aquelas utilizadas no armazenamento de conhecimentos genéricos sobre um domínio. A esse conjunto de estruturas dá-se o nome de banco de hipóteses.

Ele consta basicamente de uma lista de conceitos, predicados e atributos instanciados, de uma lista de propriedades instanciadas.

dos, de uma lista alterações a serem propagadas e de uma lista de pro
blemas a resolver. Outras estruturas são empregadas para aplicações es
pecíficas no domínio da análise de cenas (vide Seção 4.1). Estas estru
turas, descritas a seguir, foram estabelecidas de modo a permitir que
se tenha acesso às instâncias associadas a cada conceito, às partes e
particularizações associadas e que se tenha possibilidade de efetuar a
análise de conflitos e evitar o uso repetido de propriedades em condi
ções idênticas.

- a) Lista de instâncias de conceitos (hipóteses): Inclui ponteiro para conceito correspondente, referente associado, grau de certeza atual, ponteiros para as hierarquias de parte e particularizações já instanciadas e para uma lista de história de valores, indicação se o valor foi atribuído tentativamente, além de outras indicações auxiliares para percorrimento.
- b) Lista de partes associadas a hipóteses: Inclui ponteiro para hipóteses relativas à parte e ao conceito que possui a parte, grau de certeza e indicação se o valor foi atribuído tentativamente; existe também uma lista inversa, associada a esta, com lista de hipóteses que possuem uma determinada hipótese como parte.
- c) Lista de particularizações associados a uma hipótese: inclui endereço da hipótese que é particularização de outra.
- d) Lista de valores de atributos: Inclui o valor atual e o referente para a instância.
- e) Lista de instâncias de predicados: Inclui o valor atual, indicação se o valor foi atribuído tentativamente e referentes para a instância.
- f) Lista de propriedades instanciadas: Sempre que se efetua o acesso a uma propriedade, a fim de responder a uma pergunta ou pro

pagar algum dado, gera-se uma cópia da propriedade original, com diversas informações sobre a situação específica da utilização. Inclui indicação se a utilização envolve resposta a pergunta com referente a determinar, o estado em relação a subproblemas (ou seja, se a propriedade depende da resolução de algum subproblema ou se já pode ser utilizada), ponteiro para instância associada de caminho, indicação se a instância corresponde a uma reutilização ou se ela deve ser reutilizada, ponteiro para lista de referentes de variáveis, indicação se ocorreu conflito de valores durante a utilização ou se foi suspensa devido à medida ainda não realizada, além de cópia dos diversos nós, com indicação de direção de fluxo de propagação, tipo de elemento que foi associado durante o acesso à propriedade e diversos tipos de valores associados durante a utilização.

- g) Lista de valores provisórios gerados durante a utilização de uma propriedade: Destina-se a armazenar valores provisórios que são serão incorporados ao banco de hipóteses se a utilização de uma propriedade não gerar erros ou interrupções. Inclui eventual endereço de elemento correspondente no banco de hipóteses, valor associado e eventual endereço de nó correspondente na instância da propriedade ou na instância de caminho.
- h) Lista provisória de referentes de variáveis para utilização de propriedades: Durante a utilização de uma propriedade, é necessário controlar que referentes já foram atribuídos a variáveis que exigem processo de busca para a determinação de referentes. Inclui valor provisório associado à busca, se o referente está fixado e, para cada referente de cada variável, seu estado e o tipo de referente.
- i) Lista de referentes de variáveis: Associada a cada propriedade que é utilizada, armazena-se uma lista com os referentes finais associados; inclui tipo do referente e endereço da variável.

- j) Lista de alterações: Para o controle de alterações introduzidas no banco de hipóteses, utiliza-se uma lista, ordenada segundo uma medida de instante da alteração, podendo existir mais de uma alteração para o mesmo elemento do banco de hipótese. Inclui tipo e endereço do elemento alterado, tipo do valor associado à alteração, indicação se é proveniente de medida (solicitada por propriedade que ficou suspensa ou já foi reutilizada com a medida, não-solicitada ou não-proveniente de medida), ponteiro para instância de propriedade suspensa devido à medida solicitada, indicação se é proveniente de utilização de propriedade, indicação se já propagou a alteração pelas hierarquias e se é proveniente da solução de problema.
- l) Lista de instâncias de caminhos: Durante a utilização de uma propriedade, se ela possui nós que representam relação de parte, são gerados caminhos entre nós de conceito e de relação de parte e, para cada mapeamento gerado, inclui-se uma instância de um dos caminhos. Inclui ponteiro para nó no caminho, tipo do nó e referente associado ao nó ou elemento de acesso correspondente.
- m) Lista de mapeamentos de propriedades: Durante o acesso a propriedades, gera-se uma lista em que são armazenadas as propriedades mapeadas e os mapeamentos possíveis para cada uma delas. Inclui, para cada mapeamento, as associações realizadas entre nós da propriedade e elementos especificados para acesso e os referentes determinados para variáveis ou a indicação de que um processo de busca deve ser realizado.
- n) Lista de problemas: Contém perguntas geradas por perguntas com referentes a determinar ou por nós de propriedade para os quais não se detectou, na sua utilização, algum elemento com valor conhecido; os problemas podem ser relacionados entre si, no sentido de que um deles pode depender da solução de outros pa

ra poder ser solucionado. Esta lista inclui indicação do tipo do problema e elemento associado nas listas de definição, ponteiro para o problema dominante, ponteiro para o primeiro subproblema ou solução do atual, ponteiro para o próximo subproblema do mesmo problema dominante, indicação de onde o problema é gerado (inicial, gerado no acesso ou na utilização de propriedade), ponteiro para a propriedade onde foi gerado, seu estado e seus referentes.

Quanto à lista de alterações, sempre que um novo elemento é introduzido, pode-se eliminar a alteração mais antiga ou a mais recente referente ao mesmo elemento do banco de hipóteses. Pode-se, também, a fim de evitar um crescimento excessivo desta lista, utilizar mecanismos de apagamento de elementos já existentes, segundo critérios que levam em conta se o elemento é proveniente da utilização de uma propriedade, se ela corresponde a uma hipótese introduzida tentativamente ou a um dado inicial (tais elementos apagados podem vir a ser colocados, em instante posterior, de volta na lista, pois eles mantêm a condição de não-propagados nas suas listas de instâncias).

3.3 - ACESSO A PROPRIEDADES

3.3.1 - INFORMAÇÕES PARA ACESSO

As informações que podem ser utilizadas para efetuar o acesso a propriedades aplicáveis podem ser dos seguintes tipos:

a) Pergunta

- i) $C(X)$: pergunta sobre o valor-verdade que pode ser concluído para um conceito C , através das propriedades conhecidas, para um referente X ;
- ii) $RP(X, C2(Y))$: pergunta sobre relações de parte (uma pergunta que envolve conceitos ou relações de parte implícitas

por n̄s de uma propriedade n̄o gera acesso a proprie-
dade);

iii) PNC(X,Y,...): pergunta sobre um predicado n̄o-calculável
PNC, para os referentes X,Y,etc. (n̄o se permite pergunta
sobre um predicado calculável);

iv) C(): pergunta sobre a existência de referente, fazendo uso
de propriedades conhecidas, que satisfaça ao conceito C
usado predicativamente;

v) RP(X,C2()), RP(, C2(Y)): idem, para relaçao de parte.

vi) PNC(,Y,Z,...): idem, para predicado n̄o-calculável.

b) Dado:

i) C(x): dado sobre a hipotese C, que pode ter valor-verdade
conhecido ou n̄o;

ii) RP(X,C2(Y)): dado sobre uma relaçao de parte;

iii) PNC(X,Y,...): dado sobre um predicado n̄o-calculável;

iv) PC(X,Y,...): dado sobre um predicado calculável PC;

v) A(X): dado sobre um atributo;

onde se supoe que existe apenas uma pergunta e que os dados podem ser
originais ou adicionados (segundo a necessidade de encontrar elementos
correspondentes a n̄s de propriedade).

Sempre que numa propriedade for utilizada uma variável quanti-
ficada existencialmente (ou numericamente), a procura de um referente

para ela envolverá um processo de busca dentre os elementos que possam ser mapeados aos nós onde a variável é um argumento. Por este motivo, na fase de obtenção de mapeamentos estabelecem-se elementos fictícios que são mapeados a nós em que este tipo de variável ocorre.

As variáveis universais, por outro lado, podem ter seu referente fixado a partir do referente de algum argumento de um elemento de especificação mapeado a um nó onde a variável é argumento ou, então, ter o referente dependente de um processo de busca.

O exemplo a seguir mostra casos diferentes de mapeamentos, para especificações diferentes de acesso.

propriedade existente: $\forall x [H(x) \rightarrow M(x)]$

especificações para acesso:

a) - pergunta : $H(A)$

- dado : $M(A)$

Inicialmente, para $M(A)$ procuram-se, através da lista de referência de conceitos, os nós de todas as propriedades em que se tem um conceito igual empregado como predicado. Para a propriedade única dada, seus nós são percorridos procurando elementos fornecidos originalmente que possam corresponder ao nó; gera-se, então, o mapeamento:

$$\left\{ \begin{array}{l} \text{elemento } H(A) - \text{nó } H(x) \\ \text{elemento } M(A) - \text{nó } M(x) \end{array} \right.$$

Para cada variável universal, procura-se determinar um referente assinalando a sua origem; no caso, o referente determinado (A), vindo de $H(A)$, é o mesmo vindo de $M(A)$.

b) - pergunta : $H(A)$,

- dado : $M(B)$,

- elemento já conhecido no banco de hipóteses, mas não especificado para o acesso : $M(A)$.

Para a mesma propriedade acessada que no caso (a), gera-se o mapeamento:

$$\left\{ \begin{array}{l} \text{elemento } H(A) - \bar{n} \bar{H}(x), \\ \text{elemento } M(B) - \bar{n} \bar{M}(x). \end{array} \right.$$

Quando se efetua a busca de referente para a variável x , obtêm-se dois diferentes provenientes de elementos originais; elimina-se, então, o mapeamento obtido.

c) - pergunta : $H(A)$

- elementos já conhecidos, mas não especificados : $M(A)$, $M(B)$.

Para a mesma propriedade, gera-se o mapeamento

$$\left\{ \begin{array}{l} \text{elemento } H(A) - \bar{n} \bar{H}(x), \\ \text{elemento fictício } M(-) - \bar{n} \bar{M}(x). \end{array} \right.$$

onde o elemento fictício $M(-)$ é introduzido por não se ter especificado o elemento que possa corresponder ao $\bar{n} \bar{M}(x)$. Posteriormente, procura-se obter um elemento já conhecido que possa substituir o fictício, obtendo-se o mapeamento:

$\left\{ \begin{array}{l} \text{elemento} \quad H(A) - \bar{n} \bar{H}(x), \\ \text{elemento adicionado} \quad M(A) - \bar{n} \bar{M}(x). \end{array} \right.$

Poder-se-ia ter outros fatores que tornariam mais complexa a obtenção dos mapeamentos, em relação ao exemplo mostrado:

- Existência de mais de uma propriedade acessada; cada uma é analisada separadamente.
- Existência de nós que representam relações de parte na propriedade; deve-se obter todos os caminhos possíveis entre nós de conceito e nós de parte (vide Apêndice B) e, para cada um, gerar um mapeamento diferente. Para cada mapeamento, existe o problema de determinar os referentes dos elementos correspondentes aos nós intermediários dos caminhos e encontrar elementos (dentre os especificados para o acesso ou não) que correspondam a estes nós.
- Possibilidade de considerar elementos da hierarquia de particularizações, para cada conceito referenciado.
- Existência de pergunta sem referente especificado.
- Opção de como considerar variáveis universais que não tiveram referente associado: eliminar o mapeamento ou indicar que a variável irá exigir a realização de um processo de busca para a determinação de possíveis referentes.

Para o caso da existência de nós que representam relações de parte, o exemplo seguinte mostra o tipo de mapeamento efetuado:

propriedade existente: $\forall x \forall y [[A(x) \text{ E } D(x,y)] \text{ IMP } \bar{N} \bar{A} \bar{O} P(x)]$

hierarquia de partes : C PARTE-DE B

D PARTE-DE C

hierarquia de particularizações: A PARTICULARIZACAO-DE B

especificações para acesso:

- pergunta : $A(a)$
- dados : $RP(A(a),C(c)), C(c), RP(C(c), D(d)), D(d)$
- banco de hipóteses: $P(d)$

Inicialmente, associa-se a pergunta ao nō $A(x)$, criam-se elementos fictícios $RP(-(-),D(-))$ e $P(-)$, que são associados aos nōs $D(x,y)$ e $P(x)$. Gera-se um mapeamento da propriedade, com estas associações. Determina-se, a seguir, um caminho que envolve nōs de conceito e de relações de parte, a partir do nō de conceito $A(x)$ e do nō de relação de parte $D(x,y)$; neste caminho, tem-se um nō de conceito para A , um de relação para a relação entre C e A , um de conceito para C , um de relação para D e C e um de conceito para D . Gera-se, então, uma instância de caminho que é associada ao mapeamento da propriedade. Efetua-se nova associação de elementos a nōs da propriedade, associando o elemento $RP(C(c),D(d))$ ao nō $D(x,y)$. Percorrem-se os nōs da instância de caminho, para a associação de elementos a esses nōs; ao nō de conceito D é associado o elemento $D(d)$, ao nō de relação de parte $RP(C,D)$ é associado o elemento $RP(C(c),D(d))$, ao nō de conceito C o elemento $C(c)$ e ao nō de relação de parte $RP(A,C)$ o elemento $RP(A(a),C(c))$. A seguir, testa-se a consistência dos referentes dos elementos associados aos nōs da instância de caminho. Para a determinação dos referentes das variáveis universais, percorrem-se os diversos elementos especificados para o acesso: a pergunta $A(a)$ fornece o referente para a variável x , o dado $D(d)$ o referente para a variável y e, a partir des

te referente, o elemento fictício $P(-)$ tem atribuído este mesmo referente. Efetua-se, então, uma busca no banco de hipóteses a fim de substituir elementos fictícios; encontra-se um que tem descrição idêntica ao do elemento fictício $P(d)$, que passa de fictício para adicionado.

O fato de se ter perguntas e dados e, também, elementos originais ou adicionados, estabelece uma preferência que é dada quando é efetuada a determinação de referentes de variáveis (vide Seção 3.3.2). Além disso, se são especificados vários elementos para o acesso, estes elementos em conjunto podem restringir os referentes que poderão ser associados a variáveis universais.

A determinação de referentes de variáveis pode depender, também, do comando de controle TODASALT: se o valor for S, todas as alterações não-propagadas na lista de alterações geram elementos correspondentes na lista de acesso, como dados adicionados; no caso de existir uma variável universal cujo referente não tenha sido determinado, ele só pode ser determinado a partir dos elementos adicionados; por outro lado, se o valor for N, não são adicionados tais elementos e pode haver um processo de busca para a determinação dos referentes.

As perguntas em que pelo menos um dos referentes é desconhecido não são respondidas pela simples verificação se já existe de finida alguma entidade no banco de hipóteses que as respondam. Através de uma analogia com o processo de extração de respostas, quando se empregam mecanismos de dedução automática (Nilsson, 1980), elas se assemelham a perguntas do tipo " $\exists x A(x) ?$ ", aceitando-se respostas positivas ou negativas para A.

Por exemplo, para uma pergunta " $\exists x A(x) ?$ ", se existe uma propriedade

$$\forall x [B(x) \rightarrow A(x)],$$

haverá uma resposta se existe uma maneira de deduzir $B(x)$, ou se $B(b)$ for conhecido (verdadeiro, no caso); neste exemplo, pode-se dizer que o problema original foi reduzido a um outro pela utilização da propriedade dada, de saber se " $\exists x B(x)$?". Assim, uma pergunta deste tipo pode gerar outras perguntas associadas a ela, tratadas como subproblemas.

Para a geração de subproblemas, não se supõe que propriedades sejam colocadas em alguma forma normalizada; são gerados tantos subproblemas quantos são os nós de propriedade em que aparece uma variável cujo referente deve ser determinado e, somente após a resolução de todos os subproblemas é que se verifica se realmente se chega a uma solução do problema original.

Não se aceitam, para a busca de um referente que responda à pergunta, mapeamentos em que, na propriedade acessada, existe uma variável existencial que corresponda ao referente procurado; por exemplo, se a pergunta é da forma " $C()$?" e uma propriedade candidata é

$$\forall x [P(x) \rightarrow \exists y [C(y)]],$$

esta propriedade não é utilizada.

Numa situação mais geral, poderia haver um processo de busca associado, no sentido de efetuar a procura até se encontrarem todas as partes de um conceito $C()$ ou até se ter certeza que nenhum conceito está presente; até se encontrarem todos os conceitos que podem ter Y como parte, todas as partes de X ou pelo menos um referente que satisfaça ao predicado não calculável (conforme os casos possíveis de pergunta).

3.3.2 - MECANISMO DE ACESSO A PROPRIEDADES

O algoritmo que efetua o acesso a propriedades é descrito a seguir; são referenciados alguns outros algoritmos auxiliares que se não comentados posteriormente.

- i) Escolha a lista de referência de conceitos ou de predicados de acordo com uma eventual pergunta ou dados originais;
se não existe elemento na lista de acesso que seja do mesmo tipo que o da lista de referência:
 termine
se existe : vá para (ii).

- ii) Se existe nō de propriedade referenciado : vá para (iii)
se não existe : termine.

- iii) Execute o algoritmo para associação de elementos de acesso a nōs da propriedade;
se não existe nō de parte na propriedade : vá para (v)
se existe nō de parte :
 se ainda não gerou caminhos correspondentes aos nōs de conceito e de parte (vide Apêndice B) :
 execute o algoritmo para geração de caminhos
 se encontrou algum caminho :
 execute o algoritmo para introdução de instâncias de caminhos em cada mapeamento da propriedade;
 vá para (iii)
 se não encontrou nenhum caminho :
 elimine a propriedade e os mapeamentos gerados
 vá para (ii)
 se já gerou caminhos :
 se já gerou instâncias de caminhos : vá para (iv)
 se ainda não gerou :
 execute o algoritmo para introdução de instâncias de caminhos
 vá para (iii).

- iv) Execute o algoritmo para associação a nōs de caminho;
se foram eliminados todos mapeamentos da propriedade :
 vá para (ii)

- se permanece algum mapeamento :
- execute o algoritmo para a determinação de referentes de elementos a partir de referentes de instâncias de caminhos;
 - se algum referente foi alterado pelo algoritmo : v̄ para (iv)
 - se nenhum foi alterado : v̄ para (v)
- v) Se permanece algum mapeamento:
- execute o algoritmo para determinação de referentes de variáveis;
 - se algum elemento especificado para o acesso teve seu referente alterado :
 - se foram eliminados todos os mapeamentos : v̄ para (ii)
 - se permanece algum mapeamento :
 - refigual <-- true; adicref <-- false;
 - enquanto refigual e não adicref :
 - execute o algoritmo para a determinação de referentes de elementos a partir de referentes de instâncias;
 - se não modificou nenhum referente :
 - se adicionou algum referente : adicref <-- true
 - se não adicionou : refigual <-- false; - se adicref : v̄ para (v).
- vi) Se existe alguma associação entre um elemento da lista de acesso e um n̄ de propriedade ou um n̄ de instância de caminho que foi eliminada :
- execute o algoritmo para substituição de elemento na lista de acesso;
 - se deve voltar a associar n̄s da propriedade : v̄ para (iii).
- vii) Se todos os mapeamentos foram eliminados : v̄ para (ii)
- se permanece algum mapeamento :
 - se existe variável universal com referente não determinado :
 - execute o algoritmo para a determinação de referentes de variáveis e geração de subproblemas;
 - se a propriedade foi rejeitada : v̄ para (ii)

se deve voltar a associar os n̄s da propriedade :
v̄ para (iii).

- viii) Se todos os mapeamentos foram eliminados : v̄ para (ii)
se permanece algum mapeamento :
se existe algum n̄ da propriedade, que não seja n̄ de parte
ou de predicado calculável, tal que seu elemento
correspondente seja fictício e que o n̄ dependa apenas de
variáveis universais e que todas elas possuem referente
conhecido :
execute o algoritmo para a determinação de elemento
não fictício para o n̄ da propriedade;
se deve voltar a associar a n̄s da propriedade :
v̄ para (iii).

- ix) Se todos os mapeamentos foram eliminados : v̄ para (ii)
se permanece algum mapeamento e a propriedade possui n̄ de
parte:
execute o algoritmo para a determinação de elemento
não-fictício para o n̄ de caminho ;
se deve voltar a associar a n̄s da propriedade :
v̄ para (iii).
se não deve :
v̄ para (ii).

Os algoritmos auxiliares que foram mencionados ao longo do fluxo principal executam diversas funções de percorrimento das estruturas de dados que armazenam propriedades e caminhos, gerando, eliminando ou modificando os mapeamentos a n̄s de propriedade e de caminho. Estas funções são dadas, resumidamente, a seguir:

- a) Algoritmo para associação de elementos de acesso a n̄s de propriedade: Geram-se mapeamentos iniciais, se nenhum ainda foi determinado, ou outros adicionais, se já existe algum e após a sua obtenção foram adicionados outros elementos à lista de

especificações para acesso. A busca de elementos é realizada dentre os elementos originais ou adicionados e geram-se tantos mapeamentos quantas forem as combinações possíveis de associações entre nós e elementos. Aos nós de relação de parte e aos que não possuem correspondente, associa-se, no primeiro percurso, elementos fictícios.

- b) Algoritmo para introdução de instâncias de caminhos: Efetua-se um percurso, para cada caminho e cada mapeamento da propriedade, de modo a associar um novo elemento fictício aos nós do caminho que não possuem nó correspondente na propriedade (nó de relação de parte ou de conceito sem correspondente) e para as quais não existe nenhum elemento original ou adicionado que possa corresponder a eles.
- c) Algoritmo para associação de elementos de acesso a nós de instância de caminho: Efetua-se a associação de elementos de acesso a nós de instâncias, de acordo com o tipo do nó (conceito ou relação de parte) e com a existência de nó correspondente na propriedade.
- d) Algoritmo para determinação de referentes de elementos de acesso a partir de referentes de nós de instâncias de caminhos: Elementos fictícios associados a nós de instâncias de caminhos podem ter alguns de seus referentes determinados a partir de referentes de alguns dos elementos que sejam conhecidos e do fato de que nós vizinhos num caminho devem ter referentes iguais.
- e) Algoritmo para determinação de referentes de variáveis universais: Procura-se determinar, para cada mapeamento da propriedade, referentes para variáveis universais, a partir dos elementos da lista de acesso associados a nós da propriedade, de modo que sejam consistentes entre si. Em caso de conflito, há preferência por

aqueles provenientes de pergunta e, após estes, pelos provenientes de dados originais; pode haver a eliminação do mapeamento da propriedade, se não existir ordem de preferência que resolva a inconsistência. Pode haver o cancelamento da associação de um elemento a um nó quando existe conflito de referentes, o referente da variável é proveniente de pergunta ou dado original e o elemento sob consideração é um dado adicionado. Pode ser determinado, também, que uma variável é do tipo que é proveniente de pergunta com referente a determinar, quando existe uma variável que possui referente desconhecido e um elemento correspondente a um nó onde ela ocorre que também não possui referente determinado. Para elementos de relação de parte, é necessário verificar se possuem nó correspondente na propriedade ou na instância de caminho; quando possuem na instância, ainda podem ocorrer os casos de os nós de conceito, correspondentes à parte e ao conceito que possui a parte, possuem nó correspondente na propriedade, devendo-se analisar a variável do nó contra um dos referentes do elemento. Por outro lado, quando possuem na propriedade, pode ocorrer uma situação em que um elemento correspondente a um nó que representa o conceito que possui a parte receber um referente a partir do referente da variável.

- f) Algoritmo para substituição de elemento na lista de acesso: Quando ocorre a eliminação de uma associação nó-elemento na determinação de referentes de variáveis, gera-se uma descrição do elemento desejável a ser adicionado à lista de acesso, a partir dos referentes conhecidos para as variáveis universais do nó da propriedade. Através de uma busca no banco de hipóteses, pode ser adicionado algum elemento ou eliminado o mapeamento da propriedade.
- g) Algoritmo para determinação de referentes de variáveis e geração de subproblemas: Quando variáveis universais permanecem sem

ter seu referente determinado, a ação a tomar depende de existir ou não pergunta com referente a determinar. Se não existe este tipo de pergunta, pode ser eliminado o mapeamento da propriedade quando existe algum nó onde ela ocorre que possui elemento correspondente fictício e todas as alterações foram adicionadas à lista de acesso (TODASALT : S); caso isto não ocorra, a variável depende de processo de busca. Para as situações onde existe pergunta com referente a determinar, podem ser gerados outros subproblemas, eliminado o mapeamento da propriedade ou gerada a indicação que deve voltar a associar nós da propriedade.

- h) Algoritmo para determinação de elemento não-fictício para nó da propriedade: Procura determinar, a partir dos referentes conhecidos para elementos fictícios, se existem elementos no banco de hipóteses que possam ser correspondentes; pode ocorrer de não existir nenhum elemento e não existir pergunta, caso em que o elemento muda de fictício para adicionado.
- i) Algoritmo para determinação de elemento não-fictício para nó de instância de caminho: Procura encontrar elementos que possam substituir fictícios, tal como no algoritmo anterior, sendo a busca efetuada entre nós das instâncias de caminho e da propriedade que possam corresponder ao conceito da parte e ao que possui a parte.

3.4 - INDICADORES DE CONDIÇÕES DE UTILIZAÇÃO DE PROPRIEDADES

3.4.1 - ATRIBUIÇÃO DE INDICADORES DE DIREÇÃO DE FLUXO DE EXECUÇÃO

A partir das condições especificadas para o acesso a propriedades, tais como a existência de pergunta ou dados a propagar e constatação de que algumas variáveis podem exigir um processo de busca

para a determinação de seus referentes, e do fato de considerar cada propriedade, no caso geral, como uma fórmula quantificada fechada que se supõe tomar o valor-verdade verdadeiro em todos os casos, pode-se, em muitos casos, determinar, de maneira unívoca, segundo alguns critérios que serão estabelecidos, quais nōs da propriedade poderão ter seu valor utilizado e quais poderão ter o valor atribuído, durante a utilização de uma propriedade. Utiliza-se um marcador, denominado indicador de antecessor, que carrega a informação conhecida sobre a transferência de valor-verdade. Consta de 3 campos: de definição (que pode tomar valores D, P ou I, conforme o valor seja definitivo, provisório ou indeterminado), de direção (que pode tomar os valores T, R ou I, conforme se utilize ou atribua o valor do nō ou este esteja indeterminado) e de valor (que pode tomar os valores V, F ou I, conforme o valor seja verdadeiro, falso ou indeterminado).

Algumas restrições sobre os indicadores podem ser impostas a priori:

- o nō de raiz de instância da propriedade tem indicador (DRV), pois o conhecimento expresso pela propriedade é válido;
- o antecedente de uma regra de decisão tem indicador (DTV) e a conclusão (DRV), pois sō é possível utilizar o conhecimento expresso pelas condições do antecedente se este for verdadeiro e a conclusão consiste em pelo menos uma atribuição de valor;
- um nō de predicado do tipo calculável possui indicador (DTI), pois o valor deste tipo de predicado não pode ser inferido a partir do valor de outros nōs e, sim, deve ser obtido pela execução de uma rotina de cálculo associada ao predicado; se for conhecido o valor do predicado quando os referentes necessários são conhecidos, então pode ser levantada a indeterminação no campo de valor;

- um \bar{n} que corresponda a uma pergunta a ser respondida tem indicador (DRI), pois necessariamente a propriedade \bar{e} utilizada na obtenção de uma resposta para a pergunta; a indeterminação no campo de valor também pode ser levantada a partir da utilização do algoritmo que determina os indicadores, levando em conta os indicadores de outros tipos de \bar{n} s de propriedade.

No exemplo a seguir, será visto quais são os indicadores atribuídos:

- propriedade instanciada:

$$\forall x [C1(x) \text{ ----} \rightarrow (r) [PC(x) \text{ ----} \rightarrow \exists y [C2(y) \text{ E } C3(y)]]]$$

- dado: $C1(X1)$, com valor-verdade V

- variável y exige busca.

O \bar{n} correspondente à raiz recebe indicador (DRV), a seguir percorre-se o \bar{n} de antecedente da regra procurando por pergunta ou elemento com valor desconhecido, não os encontrando. Percorre-se, posteriormente, os \bar{n} s da conclusão, detectando a existência de um \bar{n} de predicado calculável PC e \bar{n} s com variável que exige busca; o \bar{n} de predicado calculável recebe indicador (DTI) e, após isto, propaga-se o indicador (DRV) da conclusão para os argumentos do \bar{n} E, o que resulta no indicador (DRV) para os \bar{n} s C2 e C3. Finalmente, recupera-se o valor conhecido do predicado, resultando num indicador (DTV) para seu \bar{n} .

Seja agora o seguinte exemplo:

- propriedade instanciada:

$$\forall x [C(x) \text{ ----} \rightarrow \forall y \forall z [PNC1(x,y,z) \text{ ----} \rightarrow PNC2(x,y,z)]]$$

- pergunta : $C(X1)$

- variáveis y e z exigem busca.

O nōs de raiz (a primeira implicação) recebe indicador (DRV), a seguir efetua-se um percorrimto dos nōs terminais da propriedade em que o nō da pergunta recebe o indicador (DRI), descobre-se que existem nōs afetados por variáveis que exigem busca (nōs de predicado não-calculável PNC1 e PNC2, dominados pela segunda implicação), que não existem nōs de predicado calculável nem nōs sem variável que exige busca e com valor do elemento associado desconhecido e que não existem dados originais associados a nōs da propriedade, atribuindo-se, então, os indicadores (DTI) ao conseqüente da raiz e aos seus argumentos. A seguir, propaga-se o indicador da raiz a seus argumentos, gerando-se indicadores (DRF) para o nō de C e (DTF) para a segunda implicação (esta é a única possibilidade de concluir algum valor para o nō de C a partir do valor do nō do conseqüente da raiz). Aos nōs de predicado são atribuídos os indicadores (DTV) e (DTF). Não havendo valores conhecidos de elementos associados a nōs, encerra-se a atribuição.

Para a mesma propriedade do exemplo anterior, mas com uma pergunta " $PNC1(X1, Y1, Z1) ?$ " e dados correspondentes aos nōs C e $PNC2$, com valor-verdade V , atribui-se (DTI) aos nōs de C e $PNC2$ pelo fato de os elementos associados a esses nōs serem originais e atribui-se (DRI) ao nō $PNC1$. Propagando o indicador associado à raiz, o nō do conseqüente da raiz recebe (DRV), o nō de $PNC1$ recebe (DRF) e o de $PNC2$ recebe (DTF). A partir dos valores conhecidos para elementos associados aos nōs, o nō de C recebe (DTV); porém, como $PNC2$ possui valor-verdade V , seu nō não pode cumprir a exigência de transmitir valor F e elimina-se a propriedade instanciada, pois ela não pode responder à pergunta.

De uma maneira geral, através do uso de indicadores é possível saber se uma pergunta pode ser respondida, a partir dos dados

conhecidos; saber se certos argumentos de um n \bar{o} podem ter seu valor atribuído ou se existe uma situação em que n \bar{a} o se tem como fazer a atribuição; minimizar a utilização das estruturas do banco de hipóteses e a atribuição de valores a elementos do banco e deixar de efetuar buscas, comandadas por quantificadores, e medidas associadas a predicados calculáveis; terminar a utilização de uma propriedade t \bar{a} o logo se conclua que n \bar{a} o se pode obter um valor desejado para um n \bar{o} .

3.4.2 - MECANISMO DE ATRIBUIÇÃO DE INDICADORES

O algoritmo que \bar{e} empregado para a atribuição de indicadores de direção de fluxo est \bar{a} descrito resumidamente a seguir:

- i) se propriedade \bar{e} regra de decis \bar{a} o :
 - n \bar{o} \leftarrow premissa;
 - execute algoritmo para a atribuição ascendente na premissa;
 - execute (ii);
 - n \bar{o} \leftarrow consequente;
 - execute o algoritmo para a escolha de indicadores de consequente;
 - execute (ii)
- se n \bar{a} o \bar{e} regra de decis \bar{a} o :
 - execute o algoritmo para a escolha de indicadores de terminais de propriedade;
 - execute (ii);
 - execute (iii);
- ii) execute o algoritmo para a atribuição de indicadores a argumentos
- iii) execute o algoritmo para a atribuição de indicadores atrav \bar{e} s de valores conhecidos para elementos.

Os algoritmos auxiliares mencionados tem suas funções resumidas a seguir:

- a) Algoritmo para atribuição ascendente em premissa: N \bar{o} s que n \bar{a} o dependem de vari \bar{a} vel que exija processo de busca e cujos elementos correspondentes durante o acesso possuem valor desconhecido recebem indicador (DRI), que s \bar{a} o propagados a seus antecessores; se algum n \bar{o} possui como elemento correspondente uma pergunta, a inst \bar{a} ncia da propriedade \bar{e} eliminada.
- b) Algoritmo para escolha de indicadores de conseq \bar{u} ente: Diversos n \bar{o} s da conclus \bar{a} o de uma regra t \bar{e} m atribu \bar{i} do um indicador a eles, conforme o tipo de elementos que foram mapeados a esses n \bar{o} s; para pergunta e elemento com valor desconhecido, sem depender de vari \bar{a} vel que exige busca, o indicador \bar{e} (DRI); para predicados calcul \bar{a} veis, desde que existam outros tipos de n \bar{o} s, o indicador \bar{e} (DTI) e, se n \bar{a} o existem, elimina-se a inst \bar{a} ncia da propriedade; para o caso em que existe pergunta ou elemento com valor desconhecido e n \bar{o} s sem indicador atribu \bar{i} do, esses recebem (DTI); para os demais casos, o tipo de indicador depende da exist \bar{e} ncia de elemento original ou adicionado e de n \bar{o} com vari \bar{a} vel que exige busca. Os indicadores atribu \bar{i} dos s \bar{a} o propagados aos antecessores dos n \bar{o} s.
- c) Algoritmo para escolha de indicadores de terminais de propriedades: De acordo com a exist \bar{e} ncia de pergunta, elemento com valor desconhecido correspondente a n \bar{o} que n \bar{a} o depende de vari \bar{a} vel que exige busca, predicado calcul \bar{a} vel, dados originais ou adicionados e n \bar{o} s com vari \bar{a} vel que exige busca, atribuem-se indicadores aos n \bar{o} s terminais que s \bar{a} o posteriormente propagados a seus antecessores.
- d) Algoritmo para atribui \bar{c} o de indicadores a argumentos: A partir do indicador atribu \bar{i} do a cada n \bar{o} de operador, efetua-se,

se possível, a atribuição de indicadores a seus argumentos; nós com campo de valor indeterminado podem vir a ter este valor determinado; existem casos em que não é possível haver a transferência de valores, sendo eliminada a instância da propriedade. Quando o indicador de um nó tem seu campo de valor determinado, é necessário verificar se seu antecessor também é alterado, interrompendo-se um processo descendente de atribuição; nestes casos, pode ainda ocorrer situações em que a alteração no antecessor introduza inconsistências.

- e) Algoritmo para atribuição de indicadores através de valores conhecidos para elementos: Valores conhecidos de dados especificados para o acesso são utilizados para realizar atribuições aos campos de valor, devendo-se verificar se são introduzidas inconsistências.

3.5 - UTILIZAÇÃO DE PROPRIEDADES

3.5.1 - CONDIÇÕES PARA UTILIZAÇÃO DE PROPRIEDADES

Após uma propriedade ter sido acessada a partir da lista de elementos especificados para acesso e ter sido efetuada a atribuição de indicadores a seus nós, para um determinado mapeamento de propriedade escolhido para utilização, pode ser iniciada a utilização propriamente dita. A utilização de uma propriedade, num caso geral em que existem variáveis que exigem busca para a determinação de referentes, é semelhante à verificação se os elementos que existem previamente à utilização são tais que as condições de transferência de valores-verdade podem ser satisfeitas. Efetua-se um percorrimto dos nós da propriedade em que se procuram referentes para as variáveis e em que os elementos correspondentes a nós que têm seu valor atribuído são colocados em uma lista provisória (que serve para atualizar o banco de hipóteses se não forem detetadas condições de falha durante a utilização).

Existem diversas condições de conflito, de falhas e de situações em que decisões sobre a continuação ou não do processo de utilização devem ser tomadas. Pode ser especificado externamente se pode ser criado um referente fictício para variáveis existenciais, quando se afirma que existe algum referente e este não é encontrado (controle CREFFIC); pode também ser especificado se referentes encontrados para variáveis existenciais devem ser armazenados como valores de funções de Skolem, para uso em aplicações posteriores em que os argumentos da função sejam os mesmos (controle ARMSKL); além disso, o tipo de ação a ser tomada quando se percorrem nós com indicador indeterminado também pode ser especificado (controle USODADO).

Durante o percorrimento para utilização, pode-se ter o caso de um certo nó ter associado a ele uma mensagem sobre condições diversas que podem ser detectadas. Sempre que se termina o percorrimento de um nó, verifica-se se existem mensagens associadas ao nó que podem determinar o reinício de um processo de busca de referentes, término de percorrimento devido a condições não satisfatórias, suspensão da utilização devido à não-existência de valor de predicado calculável, à observação de condição de conflito entre um valor já conhecido de um elemento do banco de hipóteses e um valor novo atribuído, etc.

No exemplo a seguir, são indicadas as ações executadas durante o percorrimento. A primeira implicação é referenciada como raiz e a segunda como --->. Associa-se a variáveis que exigem busca um valor-verdade provisório (de início, V para universais e F para existenciais), que controlam os valores associados às variáveis que as dominam e aos nós onde elas primeiro ocorrem. Os nós possuem diversos valores associados, que são empregados para controlar o processo de busca associado a variáveis do nó e o percorrimento de seus argumentos.

- propriedade instanciada :

$\forall x [C(x) \text{ --->} \forall y \exists z [PNC1(x,y,z) \text{ --->} PNC2(x,y,z)]]$

- pergunta : $C(X1)$?
- variáveis y e z exigem processo de busca
- elementos do banco de hipóteses :

$PNC1(X1, Y1, Z1)$, $PNC1(X1, Y2, Z1)$, $PNC2(X1, Y1, Z1)$, $PNC2(X1, Y2, Z1)$,
com valor-verdade V

y <--- V

z <--- F

raiz : L <--- vazio; campo de antecessor <--- V

----> : y exige busca e o \bar{n} é o primeiro \bar{n} a depender da variável

$PNC1$: acha $PNC1(X1, Y1, Z1)$ e $PNC1(X1, Y2, Z1)$; $ref(y)$ <--- $Y2$;
 $ref(z)$ <--- $Z1$; valor provisório <--- campo de antecessor
<--- V ;
 \bar{n} avaliado

----> : valor provisório <--- desconhecido

$PNC2$: acha $PNC2(X1, Y2, Z1)$; valor provisório <--- V ;
campo de antecessor <--- V ; \bar{n} avaliado;
emissão de mensagem de que valor-verdade não é compatível
vel com o esperado

----> : valor provisório <--- V ; armazenamento de valor de função
de Skolem; eliminação dos referentes de y e z ; indicação
que continua busca; \bar{n} s argumento assinalados como \bar{n}
-avaliados e seus valores e campos de antecessor com va
lor desconhecido

PNC1 : repete-se o processo para o elemento PNC1(X1,Y1,Z1);
repete-se, também, para os nōs ---->, PNC2 e ---->

PNC1 : não encontra mais nenhum elemento no banco de hipóteses;
emissão de mensagem de que esgotou busca de elementos e
não é o primeiro nō a depender de alguma das variáveis
que exigem busca

----> : valor do nō <--- V; campo de antecessor <--- V; nō
avaliado; emissão de mensagem de que valor-verdade não
é compatível com o esperado

raiz : valor provisório <--- valor do nō <--- V;
aviso de que não é possível, a partir do valor-verdade
do nō e de seus argumentos que têm seu valor utilizado,
calcular o valor de outros argumentos que devem ter seu
valor atribuído.

Para os dados específicos conhecidos no banco de hipóteses, não é possível responder à pergunta. Se estes dados fossem tais que o conseqüente tivesse valor F, a situação seria idêntica à do modus tollens.

3.5.2 - MECANISMO PARA UTILIZAÇÃO DE PROPRIEDADES

Para a utilização de uma propriedade emprega-se o algoritmo que tem seus passos principais mostrados a seguir; inicialmente, nō <--- raiz.

- i) se o nō é terminal:
 - se deve ter seu valor utilizado : execute (ii)
 - se deve ter seu valor atribuído : execute (iv)
 - se é não terminal : execute (v)

- ii) se o \bar{n} depende de variável existencial com referente desconhecido, existe valor de função de Skolem para os argumentos conhecidos:
 - associe \bar{a} variável existencial o referente;
 - se o \bar{n} é de predicado calculável :
 - se o \bar{n} depende de variável que exige busca e o referente de alguma variável, o qual é necessário, é desconhecido :
 - gere mensagens de que deve procurar os referentes
 - se não depende, ou todos referentes são conhecidos :
 - se o valor do predicado já foi calculado : vá para (ii)
 - se ainda não foi :
 - se o cálculo envolve medidas e estas não foram realizadas:
 - empilhe pedido de medida;
 - gere mensagem de que a utilização está suspensa
 - se não envolve ou já foram realizadas :
 - calcule o valor do predicado;
 - vá para (iii)
 - se o \bar{n} não é de predicado calculável :
 - se o \bar{n} depende de variáveis que exigem busca :
 - se existem elementos no banco de hipóteses, para os referentes conhecidos :
 - valor provisório <--- valor do elemento;
 - vá para (iii)
 - se não existem:
 - execute o algoritmo para teste de \bar{n} inicial;
 - se não terminou avaliação do \bar{n} : vá para (ii)
 - se o \bar{n} não depende de variáveis :
 - valor provisório <--- valor do \bar{n} <--- campo de antecessor <--- valor de elemento associado no acesso;
 - compare o valor com o que era esperado, pelo indicador;
 - retorne

- iii) armazene os referentes de variáveis com referente desconhecido;
se o valor provisório é conhecido e alguma variável existencial inicia dependência no \bar{n} :
 - execute o algoritmo para o armazenamento da função de Skolem;
 - se nenhuma variável inicia dependência no \bar{n} :
 - compare o valor com o que era esperado pelo indicador;
 - retorne
 - se alguma variável inicia :
 - execute algoritmo para análise de referente;
 - se continua busca de referente : vá para (ii)
 - se encerrou : retorne

- iv) se o \bar{n} não depende de variável que exige busca :
 - se o elemento do banco de hipóteses tem valor conflitante com o do elemento associado ao \bar{n} :
 - gere mensagem de conflito entre valores novos e antigos;
 - retorne
 - se o \bar{n} depende de variáveis que exigem busca :
 - se o \bar{n} depende de variável existencial que tem referente desconhecido e existe valor da função de Skolem :
 - assinale o referente da variável como sendo valor da função;
 - se existe elemento correspondente no banco de hipóteses :
 - se existe variável com referente desconhecido :
 - assinale o referente da variável;
 - se alguma variável inicia dependência no \bar{n} :
 - valor recebido <--- campo de antecessor;
 - execute o algoritmo para análise de referente encontrado;
 - se continua busca de referentes : vá para (iv)
 - se terminou : retorne
 - se não existe elemento correspondente :
 - valor auxiliar <--- desconhecido;
 - execute algoritmo para teste de \bar{n} inicial;
 - se continua busca de referente : vá para (iv)
 - se terminou : retorne;

- v) se existe argumento não-avaliado, com indicador (-T-) e ainda não encontrou argumento com valor não-compatível com o esperado :
- se algum destes argumentos não foi avaliado devido a variável que ainda não teve referente determinado e é argumento de predicado calculável :
 - se são existem argumentos deste tipo e aqueles que não podem determinar os referentes :
 - se o indicador é do tipo (-T-):
 - gere mensagem de que o valor é desconhecido
 - se o indicador é do tipo (-R-) :
 - gere mensagem de que não é possível calcular o valor a ser atribuído a argumentos de um nō
 - retorne
 - se existem outros tipos de nō :
 - escolha um deles;
 - vā para (vi)
 - se nenhum argumento deixou de ser avaliado :
 - escolha um deles;
 - vā para (vi)
 - se não existe argumento não-avaliado com (-T-) ou o último avaliado encontrou valor não-desejável :
 - se o indicador do nō é do tipo (-T-) :
 - se existe variável que exige busca para a qual é o primeiro nō :
 - se continua busca de referente : vā para (v)
 - se não continua :
 - campo de antecessor <--- valor do nō;
 - se o valor é desconhecido ou não-compatível :
 - gere mensagem
 - se não existe variável :
 - valor do nō <--- valor provisório;
 - se o valor é desconhecido ou não-compatível :
 - gere mensagem
 - retorne

se o indicador do \bar{n} é do tipo (-R-) :
valor do \bar{n} <--- campo de antecessor;
execute algoritmo para o cálculo de campo de antecessor de
argumentos com indicador (-R-);
se algum desses argumentos não teve seu campo de antecessor
atribuído ou se o valor não é compatível com o esperado :
gere mensagem;
retorne
se todos os argumentos tiveram campo atribuído :
se existe argumento não avaliado com indicador (-R-) :
escolha um deles;
 \bar{v} para (vi)
se não existe :
se não houve fim de busca de referente para o último
argumento :
valor auxiliar <--- valor provisório;
retorne;

vi) \bar{n} <--- argumento escolhido;
execute (v);
 \bar{n} <--- antecessor;
se houve falha no percorrimto : retorne
se o argumento não foi avaliado devido ao fato de não haver
referente de variável necessário :
se o valor provisório é desconhecido : \bar{v} para (v)
se é conhecido e existe análise pendente de referente :
 \bar{v} para (vii);
se o \bar{n} não depende de variável que exige busca : \bar{v} para (vii)
se depende :
se encerrou busca para alguma delas no argumento avaliado:
execute o algoritmo para teste de \bar{n} inicial;
se deve continuar a busca : \bar{v} para (v)
se deve encerrar : retorne
se não encerrou : \bar{v} para (vii);

- vii) execute o algoritmo para cálculo de valor provisório do \bar{n} ;
se o valor provisório \bar{n} é conhecido, existe variável existencial
que inicia dependência no \bar{n} e o indicador do \bar{n} é (-T-) :
 execute o algoritmo para armazenamento de função de Skolem;
 se existe análise pendente de referente encontrado e não existe
 argumento não avaliado que dependa de variáveis do \bar{n} :
 vã para (viii);
 se o \bar{n} depende de variável que exige busca, e \bar{n} é o primeiro \bar{n}
 para alguma delas, o valor provisório \bar{n} é conhecido :
 se existe argumento com indicador (-T-) que exige referente
 de variável e todos são conhecidos :
 gere mensagem de que há análise pendente de referente;
 vã para (v)
 se não existe este tipo de argumento :
 se o indicador do \bar{n} é (-R-) e o do último argumento não é:
 vã para (v)
 se não é : vã para (viii)
 caso contrário : vã para (v);
- viii) se o indicador do \bar{n} é do tipo (-R-) :
 valor recebido <--- campo de antecessor;
 execute o algoritmo para análise de referente encontrado;
 se deve continuar busca:
 assinale argumentos como não avaliados;
 vã para (v).

Os algoritmos auxiliares mencionados têm a seguinte fun
ção:

- a) Algoritmo para teste de \bar{n} inicial: Efetua um teste se o \bar{n}
atual é o primeiro \bar{n} para alguma variável que exige busca,
quando não encontra referente para alguma variável. Pode assi
nalar que continua, encerrou ou deve reiniciar a busca de refe
rentes, assinalar argumentos como não-avaliados, assinalar va

lor do \bar{n} , valor provisório e valores associados a variáveis e, também, alterar o conjunto L de variáveis que mantêm o referente fixado. Pode, também, para o caso de variáveis para as quais foi afirmada a existência de referente e ele não foi encontrado, criar referente fictício (se for permitido pelo mecanismo de controle).

- b) Algoritmo para armazenamento de função de Skolem: Dependendo do comando de controle ARMSKL e do valor provisório de um \bar{n} , pode armazenar o referente atual como valor da função ou apagar um referente já existente.
- c) Algoritmo para análise de referente: Quando é encontrado um referente para uma variável, determina o tipo de ação a ser tomada, dependendo do valor provisório de \bar{n} s com indicador (-T-) ou de valor recebido de antecessor, valor provisório do \bar{n} e de indicação se atribui valor a terminais, para \bar{n} s com indicador (-R-). A ação a ser tomada pode ser guardar referentes de variáveis, eliminar referentes, modificar conjunto de variáveis que tem referente fixado, assinalar que termina ou continua a busca de referentes e alterar valores associados às variáveis, valor do \bar{n} , valor recebido e atribuir valor a elementos da lista provisória associados a \bar{n} s (em que pode ser detectado algum conflito de valores).
- d) Algoritmo para cálculo de campo de antecessor de argumentos com indicador (-R-): A partir de valor do \bar{n} e de campo de antecessor de argumentos com indicador (-T-) que podem ser utilizados, calcula o valor de campo de antecessor de argumentos com (-R-), dependendo do tipo do \bar{n} (operador ou raiz de regra de decisão).

e) Algoritmo para cálculo de valor provisório de um \bar{n} : A partir de valores de campo de antecessor de argumentos com indicador (-T-) ou de valores auxiliares de argumentos com (-R-), calcula o valor provisório associado ao \bar{n} , dependendo do tipo de \bar{n} .

3.6 - RESOLUÇÃO DE PROBLEMAS

3.6.1 - OBTENÇÃO DE SOLUÇÕES E GERAÇÃO DE SUBPROBLEMAS

Será descrito, a seguir, de maneira mais detalhada, o tratamento de problemas e, em especial, de perguntas em que um referente é desconhecido.

Pelo fato de não se utilizarem mecanismos de prova automática de teoremas, não é tentada a resolução de problemas pelo processo de extração de respostas associado a estes mecanismos. Se desejável o emprego destes mecanismos, devem-se efetuar as necessárias interações entre este processo e o que aqui é empregado (em particular, como tratar regras de decisão).

Por este processo, através da utilização de uma propriedade, pode-se determinar algum referente ou gerar outras perguntas. Seja o seguinte exemplo:

- pergunta : $A()$?

- propriedade acessada : $\forall x [B(x) \rightarrow A(x)]$

No acesso a propriedades, se o banco de hipóteses possui uma ocorrência do conceito B (seja $B(b)$, para algum b), então determina-se o referente de x como sendo b; a propriedade é utilizada e, dependendo do valor-verdade de $B(b)$, pode ser atribuído um valor a $A(b)$ ou não. Caso seja atribuído, a pergunta " $A()$?" é considerada resolvi

da pela utilização desta propriedade, na fase de análise de utilização de propriedades. Se, por outro lado, não existe b tal que $B(b)$ esteja armazenado no banco de hipóteses, então gera-se um subproblema " $B()$?" (ou seja, uma pergunta " $\exists x B(x)$?"), que deverá ser resolvido pela utilização de alguma outra propriedade; após a resolução deste novo problema, volta-se a utilizar a propriedade que o gerou para verificar se o problema original foi resolvido também.

Para o exemplo seguinte, se existe definida a ocorrência $B(b_1, b_2)$, pode vir a ser aceito o referente b_1 para x ; se não existe, gera-se o subproblema " $B(.,)$ ".

- pergunta : $A()$?

- propriedade acessada : $\forall x [\forall y [B(x, y)] \rightarrow A(x)]$.

A criação de problemas durante a fase de acesso a propriedades está restrita pelo código de controle (PILPERG : S); o número máximo de referentes a serem determinados está limitado pelo código (MAXREFPROBL : n). Durante a utilização de uma propriedade, também podem ocorrer situações em que se geram problemas, os quais podem ser subproblemas de um outro se a propriedade em questão estiver sendo utilizada para responder a uma pergunta ou, caso contrário, podem ser problemas colocados como iniciais. As condições que podem ocasionar esta geração são aquelas em que, para um dos nós cujo valor deve ser transmitido a seu antecessor, uma das variáveis do nó exige um processo de busca para a determinação de referentes e não existe definido no banco de hipóteses nenhum elemento que corresponda ao nó (no caso do comando de controle PROCELNCOR ou do comando PROCELDESC possuírem valor S) ou todos os elementos encontrados possuem valor desconhecido (PROCELDESC: S).

Deve-se evitar que uma alteração proveniente da solução de um problema ocasione o acesso à mesma propriedade que gerou o problema (no primeiro exemplo desta seção, supondo que foi gerado um subproblema "B() ?", após a utilização da propriedade para resolver o problema original "A() ?", deve-se propagar a alteração correspondente à solução B(b), que não deve acessar a mesma propriedade exibida, que gerou "B() ?"). Para isto, ao obter uma solução de um subproblema (no exemplo, B(b)), indica-se que a alteração é proveniente da solução de um problema e qual foi a ocorrência de propriedade onde foi solucionado. Quando for a vez desta alteração ser propagada, procura-se na lista de problemas um problema já solucionado com a mesma descrição que a descrição da alteração e tal que um dos referentes tenha sido obtido na propriedade que solucionou o problema em questão; a partir da propriedade que gerou o problema, comparam-se as propriedades, eliminando-se o acesso à propriedade se forem iguais.

3.6.2 - ENCADEAMENTO RETROATIVO E PROGRESSIVO DE PROPRIEDADES

O emprego de regras de decisão para resolver problemas dados por perguntas com referente a determinar constitui uma característica adicional do modelo sendo exposto e torna seu encadeamento semelhante ao que se verifica no encadeamento retroativo de regras ("backward chaining"). Verifica-se, então, que, com este mecanismo de resolução de problemas, torna-se possível tanto o encadeamento progressivo como o retroativo de regras de decisão (sem que a maneira como uma regra será utilizada seja fixada a priori) e de propriedades em geral (Simoni e Souza, 1985). O encadeamento retroativo é efetuado, por exemplo, no MYCIN (Shortliffe, 1976) e o progressivo no OPS5 (Forgy, 1981).

No seguinte exemplo, em que A, B, C, D, F, G, e H representam conceitos, são gerados, a partir de um problema inicial "A() ?", problemas em que existe a necessidade de determinação de referentes, durante a fase de acesso a propriedades; após a resolução dos problemas, a propriedade acessada pode ser utilizada.

- propriedades : P1) $\forall x [[B(x) \text{ E } C(x)] \text{ ---> } (0.9) A(x)]$

P2) $\forall x [[\text{NAO } D(x) \text{ E } F(x)] \text{ ---> } (0.8) B(x)]$

P3) $\forall x [[G(x) \text{ E } H(x)] \text{ ---> } (0.7) C(x)]$

- dados : C(1), C(2), F(1), F(2), G(1), G(2) - valor-verdade V

D(1), D(2) - valor-verdade F

- pergunta (inicial) : A() ?

ações executadas :

A() ? - acessa P1, que gera os problemas B() ? e C() ?.

C() ? - acessa P1, que é rejeitada (pergunta corresponde a um
nã da premissa)

acessa P3, que gera C(1) (supondo que se busque apenas
uma solução)

B() ? - acessa P1, que é rejeitada
acessa P2, que gera B(1)

A(1) - acessa P1, que é rejeitada (A(1) é obtida após a solução
dos problemas gerados no primeiro acesso a P1;
P1 é rejeitada, pois o dado corresponde a um
nã da conclusão da regra)

B(1) - acessa P1, que é rejeitada (o dado corresponde a
solução de problema gerado na propriedade)
acessa P2, que é rejeitada

C(1) - acessa P1, que é rejeitada
acessa P3, que é rejeitada

No próximo exemplo, a partir de um problema inicial "A()", é acessada uma propriedade que, na sua utilização, procura encontrar elementos já definidos no banco de hipóteses com valor conhecido, elementos estes correspondentes a alguns de seus nós, e, não os encontrando, gera problema da forma "C(1) ?", que interrompe a utilização da propriedade. Após a resolução dos problemas, durante a propagação dos elementos gerados na utilização de propriedades, recai-se na geração de problemas durante a utilização de uma outra propriedade.

- propriedades: P1) $\forall x [[B(x) \text{ E } C(x)] \text{ ---> } (0.9) A(x)]$

P2) $\forall x [[D(x) \text{ E } F(x)] \text{ ---> } (0.8) C(x)]$

P3) $\forall x [[C(x) \text{ E } G(x)] \text{ ---> } (0.7) H(x)]$

P4) $\forall x [[HG(x) \text{ E } I(x)] \text{ ---> } (0.6) J(x)]$

P5) $\forall x [B(x) \text{ ---> } (0.5) I(x)]$

- hierarquia : H PARTICULARIZAÇÃO-DE HG

- dados: B(1), B(2), D(1), D(2), F(1), F(2), G(1), G(2) -
valor-verdade V

- pergunta (inicial) : A() ?

ações executadas :

A() ? - acessa P1, que gera os problemas C(1) ? e C(2) ? (na
utilização)

C() ? - acessa P1, que é rejeitada (pergunta corresponde a um
nó da premissa)

acessa P2, que calcula C(2)

acessa P3, que é rejeitada

- C(1) ? - mesmas propriedades acessadas, sendo que P2 gera C(1)
- A(1) - acessa P1, que é rejeitada (este valor é obtido após a utilização da propriedade P1; no acesso atual, P1 é rejeitada pois o dado corresponde a um não da conclusão da regra)
- C(1) - acessa P1 e P2, que são rejeitadas, e P3, que calcula H(1); através da informação sobre a hierarquia, é gerado HG(1) como generalização de H(1)
- HG(1) - acessa P4, que gera o problema I(1) ? (na utilização)
- I(1)? - acessa P4, que é rejeitada, e P5, que calcula I(1)
- J(1) - acessa P4, que é rejeitada
- I(1) - acessa P4 e P5, que são rejeitadas
- H(1) - acessa P3, que é rejeitada
- A(2) - repete-se o mesmo ciclo de ações que foi executado para A(1).

Existem locais no fluxo de controle que permitem a inserção de heurísticas próprias a cada problema, através de rotinas específicas; duas destas heurísticas, relacionadas com a utilização de regras, são:

- para a escolha de problema a analisar, para o caso de haver mais de um problema não-resolvido;
- para a escolha de mapeamento de propriedade a executar.

3.7 - CONTROLE DE AÇÕES

3.7.1 - CARACTERÍSTICAS PRINCIPAIS

Para o controle das ações efetuadas, os passos seguidos estão esquematizados a seguir: a existência de alterações não propagadas na rede pode ocasionar a adição de elementos na lista de acesso, os quais geram o acesso a propriedades que são utilizadas e podem ocasionar uma série de ações sobre o banco de hipóteses, inclusive a geração de novas alterações; por outro lado, a existência de problemas a resolver pode também gerar o acesso a propriedades. Outro tipo possível de ação é a realização de medidas sobre a imagem. Nada disso sendo possível, existem procedimentos de escolha de ação de inicialização.

A cada elemento da descrição da cena podem ser associadas diversas hipóteses, tanto a partir de valores já inferidos, como também tentativamente. As hipóteses atribuídas tentativamente podem vir a ser refutadas, quando então se pode atribuir outras hipóteses, de maneira tentativa, ao elemento da descrição ou testar as consequências de o valor-verdade atribuído ser invertido.

Existem situações em que é necessário estabelecer uma ação a tomar por não haver nem alteração a propagar, nem problema a resolver. Algumas opções são: propagação de alterações que tiveram a propagação cancelada; atribuição tentativa de hipóteses a elementos sem hipótese associada; atribuição tentativa de valor-verdade V a elementos da descrição da cena com hipóteses associadas com valor desconhecido; modificação tentativa do valor-verdade de elementos que possuem todas as hipóteses associadas com valor-verdade F .

Variáveis existenciais podem ter seus referentes atribuídos através de referentes fictícios, sempre que não se encontram elementos que possam satisfazer as condições de existência estipuladas durante a utilização de uma propriedade. Antes deste tipo de atribui

ção, no entanto, pode ser tentada a atribuição de referente a elemento da descrição da cena que possua hipótese associada com valor desconhecido, tal que, na propriedade onde foi constatada a inexistência do referente, tal hipótese corresponda à parte num nó de relação de parte. Nestes casos, se a hipótese encontrada já possui relação de parte com outra que tenha referente fictício e o referente correspondente à hipótese que possui a parte também é fictício, então os referentes fictícios são tomados como idênticos.

Visto que os valores-verdade associados a hipóteses possuem indicação de sua origem, é possível, sempre que uma hipótese tem seu valor alterado, verificar as conseqüências desta alteração sobre outras hipóteses que tiveram seu valor atribuído em propriedades que fizeram uso do valor-verdade da hipótese que teve seu valor alterado. Pode, então, ocorrer uma reutilização da propriedade já utilizada, para a utilização dos valores alterados.

3.7.2 - DETALHES SOBRE O MECANISMO DE CONTROLE

O algoritmo principal destinado a efetuar a propagação de valores e resolução de problemas será mostrado a seguir.

- i) se não existem nem alterações nem problemas : vá para (v)
se existe alguma :
 - elimine as alterações já propagadas;
 - lista de acesso <--- vazio;
 - execute o algoritmo para determinação de ação;

- ii) se a lista de acesso está vazia :
 - se existe propriedade que estava com utilização suspensa devido a problemas e estes estão resolvidos :
 - se ainda não atribuiu indicadores de direção :
 - execute o algoritmo para atribuição de indicadores;
 - execute o algoritmo para verificação de emprego anterior

da propriedade;

se ainda não foi utilizada :

indique que pode utilizar a propriedade

se já foi : vá para (ii)

se existe propriedade que estava suspensa devido a medidas não realizadas e estas já o foram :

indique que pode utilizá-la

se existe problema não resolvido :

escolha um deles para resolver;

preencha a lista de acesso com informações sobre ele;

vá para (ii)

se a primeira alteração é proveniente de atribuição de valor feita tentativamente :

preencha lista de acesso com informações sobre alteração;

vá para (ii)

se a primeira alteração é correspondente a elemento do banco de hipóteses com valor desconhecido :

se pode efetuar pergunta sobre este tipo de elemento :

adicione pergunta à lista de acesso;

vá para (ii)

se não pode :

elimine alteração;

vá para (i)

se a lista de acesso não está vazia :

indique que encerra busca para acesso;

se deve introduzir todas as alterações (TODASALT : S) :

coloque todas as alterações na lista de acesso;

execute o algoritmo para efetuar o acesso a propriedades;

indique que deve procurar mapeamentos de propriedade

iii) se deve procurar mapeamento de propriedade :

se não existe mapeamento :

elimine alterações que geraram acesso a propriedades

vá para (i)

se existe mapeamento :

escolha um deles;

execute o algoritmo para atribuição de indicadores;

execute o algoritmo para determinação de utilização anterior;

se existe utilização anterior :

elimine o mapeamento atual;

se o tempo de execução é menor que o indicado para a execução de medidas :

v̄ para (v)

se é maior ou igual : v̄ para (iv)

iv) se deve utilizar propriedades :

execute o algoritmo para utilização de propriedades;

execute o algoritmo para análise de utilização;

se deve reutilizar propriedade : v̄ para (iv)

se não deve : v̄ para (iii);

v) se a fração de tempo para medidas não é adequada : v̄ para (i)

se é adequada :

se existe pedido empilhado de medida :

execute uma delas;

v̄ para (v)

se não existe :

execute o algoritmo para determinação de medida espontânea (vide Seção 4.2);

se encontrou alguma : v̄ para (v)

se não encontrou :

execute o algoritmo para determinação de ação inicial;

se determinou alguma ação : v̄ para (i)

se não encontrou : termine

Uma descrição sucinta dos demais algoritmos ainda não mencionados é:

- a) Algoritmo para verificação de emprego anterior de propriedades: Procura determinar se a propriedade já foi utilizada com os mesmos dados associados aos mesmos nós da propriedade.

- b) Algoritmo para análise de utilização: Efetua alterações no banco de hipóteses, a partir de valores que foram atribuídos ou criados durante a utilização de uma propriedade; efetua análise de compatibilidade entre valores novos e antigos do banco de hipóteses; efetua a propagação de alterações pelas hierarquias instanciadas de conceitos; efetua análise de resposta de perguntas, quando propriedade é utilizada para resolver problemas; elimina referentes criados em utilização anterior, quando é reutilização de propriedade, com as correspondentes alterações nas hierarquias; procura determinar hipóteses cujo referente pode ser empregado como referente de variável existencial, quando não é encontrado o referente de alguma variável que se supunha existir; determina que propriedade deve ser reutilizada.

CAPÍTULO 4

PROCESSO DE ANÁLISE

4.1 - ESTRUTURAS DE DESCRIÇÃO DE UMA CENA

Durante o processo de análise de uma cena, faz-se referência a seus elementos, além de se efetuarem medidas sobre a própria imagem.

Na versão atual da implementação do programa, as imagens podem possuir 64 x 64 elementos, cada qual com 256 níveis de cinza, em 3 canais.

As estruturas de armazenamento definidas são:

- a) Área de cabeçalho de segmentos, que possui associada a cada um seu número interno, ponteiro para área com pontos extremos, ponteiro para área com processo de formação, ponteiro para área de interpretações; além disso, a partir do número interno, é possível acessar a área de máscara.
- b) Área com pontos extremos, que possui, para a ordenada y, os pares de abcissas correspondentes a pontos do segmento.
- c) Área com processo de formação, que possui ponteiro para área de cabeçalhos, uma parte relativa à obtenção da segmentação atual (com ponteiro para segmentação que originou a atual, se ela foi obtida por separação - "split" - da anterior, e ponteiro para lista de aglomerações de que a segmentação atual é parte) e uma parte relativa a seus componentes (com ponteiro para lista de suas separações - pode haver mais de uma separação, cada uma com um conjunto de segmentos, obtidas por razões diferentes - e ponteiro para lista de seus componentes, quando a

segmentação atual for do tipo aglomeração); associa-se a cada separação e a lista de componentes a razão pela qual elas foram obtidas (fornecida, gerada por histograma, borda, envoltória, separação, etc.).

d) Área com interpretações, que possui ponteiros para elementos de hipóteses ativadas no banco de hipóteses.

e) Área com máscaras de regiões.

Além destas, existem áreas auxiliares ao processo de análise para o armazenamento de fração de vezes que se deve escolher determinada medida (numa determinação espontânea de medidas) e de memória de região sob enfoque para medidas espontâneas. Para o caso de separação de uma região em outras pela observação de picos em histogramas ("split"), em que, por exemplo, o segmento S_1 é subdividido em dois subconjuntos S_2, \dots, S_k e S_{k+1}, \dots, S_n , onde S_{k+1}, \dots, S_n forma a parte homogênea devido ao pico no histograma, diz-se que S_1 subdivide-se, por separação, em S_2, \dots, S_k e S_{n+1} , onde S_{n+1} é formada pelos segmentos S_{k+1}, \dots, S_n por aglomeração.

4.2 - CARACTERÍSTICAS DO PROCESSO DE ANÁLISE

Uma das características do modelo é a possibilidade de poder gerar medidas espontaneamente sobre a imagem que descreve uma cena, sem que estas sejam comandadas pelo processo de propagação de valores ou de resolução de problemas.

Quanto ao relacionamento entre os processos de medida e o de propagação de valores (vide Seção 3.7), é possível simular um comportamento semi-independente de ambos, através dos comandos de controle FRACMED e FRACMPED, que permitem uma divisão do tempo de processamento entre propagação de valores, realização de medidas solicitadas durante a utilização de propriedades e realização de medidas geradas de maneira independente.

Outra característica é a possibilidade de utilizar informações contextuais para a resolução de ambiguidades e atribuição de hipóteses a elementos da descrição da cena. Uma maneira é pela escolha de hipóteses a serem atribuídas a elementos que não possuem hipóteses associadas (vide Seção 3.7). Outra maneira é utilizar as rotinas para a escolha de política de ação e de política de ação inicial para que, sempre que existam diversos elementos com múltiplas interpretações e pelo menos dois elementos tais que as interpretações sejam partes instanciadas de um mesmo conceito, seja utilizado este conceito como contexto de busca. Quando existe mais de um contexto a ser levado em consideração, a escolha pode ser feita a partir de regras empíricas. Outro caso é o da existência de mais de um contexto, não contraditórios, para diversos elementos. Em qualquer dos casos, o emprego de informações contextuais deve ser tomado como uma heurística.

Segundo a maneira proposta para o problema de análise de cenas, o termo medida designa um conjunto de operações que podem ser efetuadas sobre uma imagem. Das diversas medidas possíveis, algumas são conhecidas a priori pelo algoritmo que efetua a escolha de medidas espontâneas.

Na implementação do modelo, tem-se pelo menos um procedimento cujo corpo pode ser compilado separadamente, destinado a efetuar as operações correspondentes à medida; define-se um predicado na rede associativa, cujo código de execução é aquele da medida que ele pretende executar (conforme os códigos associados a medidas, no algoritmo de escolha de medidas).

Pelo fato de o nome do predicado ser conhecido na rede associativa, ele também pode ser empregado em propriedades.

Outras medidas somente serão executadas a partir da utilização de propriedades em que os predicados correspondentes são nós das propriedades.

As medidas a serem executadas, solicitadas ou espontâneas, são inicialmente empilhadas e, de acordo com a parcela do tempo de processamento destinado a medidas, são desempilhadas e executadas. Pode ser introduzida uma indicação de prioridade para a execução de medidas.

As operações sobre a imagem podem ser agrupadas em três categorias:

- a) descrição de regiões: descritores de cor (média, variância), de forma (área, perímetro, centro de massa, etc.);
- b) geração de novas regiões: separação ("split"), fusão ("merge"), envoltória de região;
- c) relação geométrica: acima, ao lado, dentro, etc.

Algumas medidas são executadas automaticamente quando uma nova região é formada (área, perímetro e outras; pode ser informado quais delas devem ser sempre calculadas ou não).

A cada operação que seja definida, além destas mencionadas (que podem ser inseridas na forma de procedimentos compilados separadamente, empregando alguns dentre os diversos "cabeçalhos" já existentes), devem ser considerados: o código de execução da operação, com os tipos de entrada e saída das rotinas e seus números internos; a maneira como a informação sobre o resultado deve ser colocada na lista de alterações e a maneira como as rotinas podem ser chamadas pelos códigos de execução associados a atributos e predicados.

Quanto ao mecanismo de escolha de medida espontânea, ele leva em conta uma memória, para cada tipo possível de medida, onde se armazena a última região da imagem onde foi executada a medida; eventualmente, esta memória pode ser alterada externamente de modo a dirigir a atenção a alguma parte específica.

4.3 - APLICAÇÃO EM SEGMENTAÇÃO CONTROLADA PELA INTERPRETAÇÃO

O processo da segmentação controlada pela interpretação que foi desenvolvido por Tenenbaum e Barrow (1976) pode ser reproduzido utilizando as características do modelo presente. Diversos processos de busca e condições de alteração de valores-verdade são colocados como propriedades e parâmetros do fluxo de controle (ou seja, dados para o programa). Utilizam-se regras de decisão para propagar inconsistências entre interpretações; a ordem com que a propagação é feita não é necessariamente idêntica àquela do método dos autores citados, porém o resultado e a natureza das operações envolvidas o são.

Quando não se efetuam fusões ("merges") entre regiões, ou seja, quando se estuda apenas o processo de propagação, as restrições que surgem no artigo citado podem ser colocadas como nas regras (1)-(19) adiante. Estas restrições indicam, para cada relação geométrica, quais as interpretações permitidas entre pares de regiões para as quais vale a relação. Inicialmente, cada região recebe, de forma tentativa, todas as interpretações possíveis; estas interpretações são adicionadas à lista de alterações a propagar.

Para cada região, acessam-se as propriedades e introduzim-se, eventualmente, alterações no banco de hipóteses. As regras de decisão são escritas de forma que as alterações possíveis são aquelas em que se atribui valor-verdade F a hipóteses que são também mapeadas a nós da premissa e que possuem valor-verdade V antes da utilização das regras. Observa-se, nestes casos, uma inconsistência entre valores antigos e novos do banco de hipóteses devido à utilização de uma propriedade; este tipo de inconsistência tem a sua ocorrência assinalada durante a utilização de uma propriedade. Após o término da utilização, durante a fase de análise de seus resultados, é invocada uma heurística para a tentativa de solução deste tipo de problema; para esta aplicação em particular, mantêm-se o novo valor gerado (além disso, se o valor antigo estava colocado como alteração a ser propagada e ainda não tinha sido, apaga-se esta ocorrência antiga da alteração pelo uso do comando de controle (ELIMIG : S)).

-- propriedades (caso em que não efetuam fusões entre regiões):

- P1) $\forall x [[K(x) \text{ E } \text{NÃO PEQUENO}(x)] \text{ ---}>(1.0) \text{ NÃO } K(x)]$
- P2) $\forall x [[P(x) \text{ E } \text{NÃO PEQUENO}(x)] \text{ ---}>(1.0) \text{ NÃO } P(x)]$
- P3) $\forall x [\exists y [P(x) \text{ e } \text{NÃO } W(y) \text{ E } \text{DENTRO}(x,y)] \text{ ---}>(1.0) \text{ NÃO } P(x)]$
- P4) $\forall x [\exists y [K(x) \text{ E } \text{NÃO } D(y) \text{ E } \text{DENTRO}(x,y)] \text{ ---}>(1.0) \text{ NÃO } K(x)]$
- P5) $\forall x [\exists y [W(x) \text{ e } \text{NÃO } P(y) \text{ E } \text{ENVOLVE}(x,y)] \text{ ---}>(1.0) \text{ NÃO } W(x)]$
- P6) $\forall x [\exists y [D(x) \text{ E } \text{NÃO } K(y) \text{ E } \text{ENVOLVE}(x,y)] \text{ ---}>(1.0) \text{ NÃO } D(x)]$
- P7) $\forall x [\exists y [B(x) \text{ E } \text{NÃO } F(y) \text{ E } \text{ACIMA}(x,y)] \text{ ---}>(1.0) \text{ NÃO } B(x)]$
- P8) $\forall x [\exists y [D(x) \text{ E } \text{NÃO } F(y) \text{ E } \text{ACIMA}(x,y)] \text{ ---}>(1.0) \text{ NÃO } D(x)]$
- P9) $\forall x [\exists y [W(x) \text{ E } \text{NÃO } B(y) \text{ E } \text{ACIMA}(x,y)] \text{ ---}>(1.0) \text{ NÃO } W(x)]$
- P10) $\forall x [\exists y [B(x) \text{ E } \text{NÃO } W(y) \text{ E } \text{ABAIXO}(x,y)] \text{ ---}>(1.0) \text{ NÃO } B(x)]$
- P11) $\forall x [\exists y [F(x) \text{ E } [\text{NÃO } B(y) \text{ E } \text{NÃO } D(y)] \text{ E } \text{ABAIXO}(x,y)]$
 $\text{ ---}>(1.0) \text{ NÃO } F(x)]$
- P12) $\forall x [\exists y [B(x) \text{ E } \text{NÃO } D(y) \text{ E } \text{AOLADO}(x,y)] \text{ ---}>(1.0) \text{ NÃO } B(x)]$
- P13) $\forall x [\exists y [D(x) \text{ E } [\text{NÃO } B(y) \text{ E } \text{NÃO } W(y)] \text{ E } \text{AOLADO}(x,y)]$
 $\text{ ---}>(1.0) \text{ NÃO } D(x)]$
- P14) $\forall x [\exists y [W(x) \text{ E } \text{NÃO } D(y) \text{ E } \text{AOLADO}(x,y)] \text{ ---}>(1.0) \text{ NÃO } W(x)]$
- P15) $\forall x [\exists y [[B(x) \text{ OU } D(x) \text{ OU } F(x) \text{ OU } W(x)] \text{ E } \text{ATIVO}(y) \text{ E}$
 $\text{DENTRO}(x,y)]$
 $\text{ ---}>(1.0) [\text{NÃO } B(x) \text{ E } \text{NÃO } D(x) \text{ E } \text{NÃO } F(x) \text{ E } \text{NÃO } W(x)]]$

P16) $\forall x [\exists y [[B(x) \text{ OU } F(x) \text{ OU } K(x) \text{ OU } P(x)] \text{ E } \text{ATIVO}(y) \text{ E } \text{ENVOLVE}(x,y)]$
--->(1.0) [NÃO B(x) E NÃO F(x) E NÃO K(x) E NÃO P(x)]]

P17) $\forall x [\exists y [[F(x) \text{ OU } K(x) \text{ OU } P(x)] \text{ E } \text{ATIVO}(y) \text{ E } \text{ACIMA}(x,y)]$
--->(1.0) [NÃO F(x) E NÃO K(x) E NÃO P(x)]]

P18) $\forall x [\exists y [[D(x) \text{ OU } K(x) \text{ OU } P(x) \text{ OU } W(x)] \text{ E } \text{ATIVO}(y) \text{ E } \text{ABAIXO}(x,y)]$
--->(1.0) [NÃO D(x) E NÃO K(x) E NÃO P(x) E NÃO W(x)]]

P19) $\forall x [\exists y [[F(x) \text{ OU } K(x) \text{ OU } P(x)] \text{ E } \text{ATIVO}(y) \text{ E } \text{AOLADO}(x,y)]$
--->(1.0) [NÃO F(x) E NÃO K(x) E NÃO P(x)]]

Nestas propriedades, os conceitos B, D, F, K, P e W têm o significado de "rodapê", "porta", "piso", "maçaneta", "pintura" e "parede", respectivamente. Os predicados ABAIXO, ACIMA, AOLADO, DENTRO, ENVOLVE e PEQUENO são calculáveis e expressam relações entre posições de regiões e tamanho de regiões. Para efeito de teste do mecanismo de propagação, foram fornecidos como entrada seus valores-verdade para todos os pares de regiões e todas as regiões, respectivamente.

O predicado ATIVO, não-calculável, foi empregado para que se possa ter a capacidade de encontrar referente para uma variável existencial em algumas propriedades onde ela aparece só como argumento de predicados calculáveis e seus referentes precisam estar determinados antes desses predicados serem utilizados. Para o teste do mecanismo de propagação, foi atribuído ao predicado o valor-verdade V, para todas as regiões.

O teste empregou o mesmo tipo de cena que aquele do artigo citado (reproduzida na Figura 4.1). As regiões estão numeradas de 2 a 7; para efeito de representação interna no modelo presente, a região de número interno 1 é sempre aquela correspondente à imagem toda.

Para esta cena, valem os seguintes predicados (não foram colocadas propriedades que expressem a simetria da relação AOLADO nem que expressem o fato de as relações ABAIXO-ACIMA e DENTRO-ENVOLVE serem inversas).

ABAIXO : para os pares de regiões (6,2), (7,4) e (7,6)

AO LADO : para (2,4) e (3,2)

DENTRO : para (3,2) e (5,4)

PEQUENO : para as regiões 3 e 5.

A ordem com que as interpretações são colocadas inicialmente na lista de alterações é arbitrária; no caso do teste efetuado, para cada região (7, 6, 5, 4, 3 e 2) adicionam-se as interpretações W, P, K, F, D e B. Ao tomar a última alteração disponível como sendo a próxima a propagar, a ordem inicial com que elas devem ser tomadas é B(2), D(2), ..., W(7). À medida que são introduzidas novas alterações, para interpretações que são negadas, estas são colocadas ao fim da lista, sendo as próximas a serem consideradas.

Eventualmente, pode ser válida mais de uma relação geométrica entre duas regiões. Nestes casos, deve ser feita uma intersecção dos conjuntos possíveis de interpretações; isto é equivalente, na presente formulação, à aplicação de diversas propriedades consecutivamente, podendo-se reduzir, então, o conjunto de interpretações.

As ações executadas seguem o processo descrito de propagação de alterações. A alteração devida ao valor inicial de B(2) gera o acesso a P7, P9 (que é rejeitada no acesso), P10, P11 (rejeitada), P12, P13 (rejeitada), P15 e P16. A utilização de P7, P10, P12 e P15 não gera nenhuma alteração. Para P16, negam-se as hipóteses iniciais B(2), F(2), K(2) e P(2). Estas hipóteses negadas provocam a reutilização das propriedades P7, P10, P12 e P15, que não resulta em novas alterações. As alterações introduzidas são agora as primeiras a serem consideradas para propagação.

Desta forma, propagam-se P(2), K(2), F(2), B(2) (negadas), D(2), W(2), B(3) (que nega as interpretações B(3), D(3), F(3) e W(3)), etc. Pela propagação de D(6), obtêm-se a negação de D(6), K(6), P(6) e W(6).

Diversas propriedades são reutilizadas, devido a estas modificações e, dentre estas, P14 quando foi acessada por W(4); a nova utilização desta propriedade gera a negação de W(4). Ao fim do processo, obtêm-se a interpretação final W(2), P(3), D(4), K(5), B(6) e F(7), única, para esta como específica.

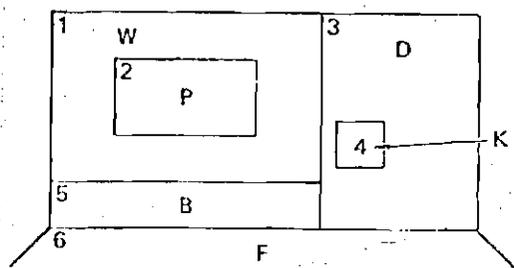


Fig. 4.1 - Cena para teste de fusão.

FONTE: Tenenbaum and Barrow (1976).

Sendo desejável reproduzir também o processo de fusão (vide Figura 1.1), devem ser consideradas as novas restrições provenientes do trabalho citado. Estas restrições determinam que, se uma relação geométrica (ABOVE, BESIDE, etc.) é válida entre duas regiões R1 e R2, então tem-se, para cada interpretação de R1, as possíveis interpretações de R2. Gera-se um novo conjunto de 48 propriedades (para as 8 relações e 6 interpretações), cada uma da forma a seguir:

P1') $\forall x [\exists y [ATIVO(x) \text{ E } ATIVO(y) \text{ E } ACIMA(x,y) \text{ E } B(x) \text{ E}$
 $N\tilde{A}O B(y) \text{ E } N\tilde{A}O F(y)] \text{ ---}>(1.0) N\tilde{A}O B(x)]$

A fusão s̄o deve ser realizada entre pares de regiões com pelo menos uma interpretação comum. A fusão mais segura ̄e aquela em que existe uma ũnica interpretação comum; ̄e este caso que ̄e considerado a seguir.

A determinação dos pares de regiões candidatas a sofrer fusão pode ser feita pela propriedade P49' a seguir. Pela propriedade P50', pode ser comandada a fusão de duas regiões que tenham uma ũnica interpretação cada uma. Inicialmente, todas as regiões presentes devem ter atribuídos o valor-verdade V para os predicados ATIVO e ATRIBINT e V para todas as interpretações; todos estes valores s̄o colocados na lista de alteraçõs a propagar.

P49') $\forall x \forall y [[ADJ(x,y) \text{ E } ATIVO(x) \text{ E } ATIVO(y) \text{ E}$
 $[[[B(x) \text{ E } B(y)] \text{ OU } [D(x) \text{ E } D(y)] \text{ OU}$
 $[F(x) \text{ E } F(y)]] \text{ OU}$
 $[[K(x) \text{ E } K(y)] \text{ OU } [P(x) \text{ E } P(y)] \text{ OU}$
 $[W(x) \text{ E } W(y)]]]]$
 $\text{---}>(1.0) CANDIDATO(x,y)]$

P50') $\forall x \forall y \forall z [[[CANDIDATO(x,y) \text{ E } ATIVO(x) \text{ E } ATIVO(y)] \text{ E}$
 $IGUALINTVERD(x,1) \text{ E } IGUALINTVERD(y,1) \text{ E}$
 $FUSAO(x,y,z)] \text{ ---}>(1.0)$
 $[[N\tilde{A}O ATIVO(x) \text{ E } N\tilde{A}O ATIVO(y) \text{ E } ATIVO(z) \text{ E}$
 $N\tilde{A}O ATRIBINT(z) \text{ E } N\tilde{A}O CANDIDATO(x,y) \text{ E}$
 $FUNDIU(x,y,z)]]$

P51') $\forall x \forall y \forall z [[FUNDIU(x,y,z) \text{ E } N\tilde{A}O ATRIBINT(z) \text{ E } B(x)] \text{ ---}>(1.0)$
 $[B(z) \text{ E } ATRIBINT(z)]]$

P52') $\forall x \forall y \forall z$ [[FUNDIU(x,y,z) E NÃO ATRIBINT(z) E D(x)] ---->(1.0)
[D(z) E ATRIBINT(z)]]

P53') $\forall x \forall y \forall z$ [[FUNDIU(x,y,z) E NÃO ATRIBINT(z) E F(x)] ---->(1.0)
[F(z) E ATRIBINT(z)]]

P54') $\forall x \forall y \forall z$ [[FUNDIU(x,y,z) E NÃO ATRIBINT(z) E K(x)] ---->(1.0)
[K(z) E ATRIBINT(z)]]

P55') $\forall x \forall y \forall z$ [[FUNDIU(x,y,z) E NÃO ATRIBINT(z) E P(x)] ---->(1.0)
[P(z) E ATRIBINT(z)]]

P56') $\forall x \forall y \forall z$ [[FUNDIU(x,y,z) E NÃO ATRIBINT(z) E W(x)] ---->(1.0)
[W(z) E ATRIBINT(z)]]

O predicado não-calculável ATIVO indica se uma região não é parte de outra por fusão. O predicado não calculável ATRIBINT indica se uma região já teve atribuído o conjunto de interpretações iniciais, quando é criada. O predicado não-calculável CANDIDATO indica se um par de regiões é candidato a sofrer fusão ("safe merge"). O predicado não-calculável FUNDIU indica que as regiões dos dois primeiros argumentos foram fundidas na região do último argumento. O predicado calculável ADJ indica se duas regiões são adjacentes. O predicado calculável FUSÃO gera, para o par de regiões dos dois primeiros, a nova região correspondente ao último argumento (que pode ser tomado como desconhecido na especificação do predicado). O predicado calculável IGUALINTVERD verifica se o número de interpretações com valor-verdade V, associadas a uma região, é igual a um dado número.

4.4 - APLICAÇÃO EM SEPARAÇÃO DE REGIÕES

Em outra aplicação desenvolvida, para demonstrar algumas das capacidades do modelo, emprega-se a separação de regiões ("split"), tanto de maneira espontânea como a partir de um comando da linguagem de acesso à rede.

A geração de separações foi elaborada na forma de uma rotina que acessa diversas outras auxiliares e que é a única a poder ser acessada externamente (através da linguagem de acesso). Outra opção de implementação seria, por exemplo, ter as diversas rotinas auxiliares acessíveis exteriormente, de modo a permitir que os diversos passos pudessem ser executados de acordo com uma solicitação explícita pela linguagem de acesso. Os passos executados são basicamente os seguintes: gera-se o histograma da região a ser separada, determina-se o melhor pico em cada um dos canais e procede-se à separação em que se gera a máscara da região (não necessariamente conexa) correspondente ao pico e da região diferença, determinam-se os componentes conexos e calculam-se os atributos iniciais para as novas regiões formadas.

A título de exemplo, considere-se o conjunto (P1)-(P8) de propriedades a seguir, em que se procura comandar a separação de uma região, sempre que ela for do tipo vegetação em geral e a variância de níveis de cinza for muito grande, ou sempre que uma região não possuir interpretação associada ou possuir mais de uma com valor-verdade V.

P1) $\forall x$ [[ATIVO(x) E COR(x,intv1)] ---->(1.0) TERRA(x)]

P2) $\forall x$ [[ATIVO(x) E COR(x,intv2)] ---->(1.0) VEGET1(x)]

P3) $\forall x$ [[ATIVO(x) E COR(x,intv3)] ---->(1.0) ÁGUA(x)]

P4) $\forall x$ [[VEGET1(x) E NÃO CANDIDATOSEP(x) E NÃO EFETUOUSEP(x) E MAIORVAR(x,lim)] ---->(1.0) CANDIDATOSEP(x)]

P5) $\forall x$ [[VEGET1(x) E EFETUOUSEP(x) E COR(x,intv4)] ---->(1.0) VEGET2(x)]

P6) $\forall x$ [[VEGET1(x) E EFETUOUSEP(x) E COR(x,intv5)] ---->(1.0) VEGET3(x)]

P7) $\forall x$ [[ATIVO(x) E NÃO CANDIDATOSEP(x) E NÃO EFETUOUSEP(x) E
[NÃO MAIORINTASSOC(x,0) OU MAIORINTVERD(x,1)]] ---->(1.0)
CANDIDATOSEP(x)]

P8) $\forall x$ [[CANDIDATOSEP(x) E NÃO EFETUOUSEP(x) E SEPARAREGIAO(x)]
---->(1.0) NÃO CANDIDATOSEP(x)]

O predicado não calculável ATIVO é empregado para designar cada nova região formada; inicialmente, a região correspondente à imagem toda tem atribuído o valor-verdade V para ela. A atribuição às novas regiões formadas é feita através da rotina que efetua a separação (de maneira espontânea ou não); este recurso é adotado para que as novas regiões formadas possam ser referenciadas como argumentos de conceitos ou predicados em propriedades.

Este predicado é necessário, por exemplo, em P1, onde o predicado calculável COR exige que o referente para x esteja determinado antes que possa ser utilizado. O predicado não-calculável CANDIDATOSEP indica se a região x foi designada como candidata a ser separada; atribui-se, tentativamente, o valor-verdade F a cada nova região formada:

O predicado calculável COR verifica se as médias de nível de cinza para a região x estão dentro do intervalo especificado. O predicado calculável EFETUOUSEP indica se a região x já sofreu um processo de separação. O predicado calculável MAIORVAR verifica se a variância dos níveis de cinza está acima de um dado limiar. O predicado MAIORINTASSOC verifica se o número de interpretações associado a uma região (qualquer que seja o valor-verdade associado às interpretações) é maior que um dado número. O predicado calculável MAIORINTVERD verifica se o número de interpretações com valor-verdade V é maior que um dado número. Para estes dois últimos predicados, seria possível, para este exemplo, em que o número de possíveis interpretações é reduzido, ter o teste dado diretamente pela linguagem de acesso.

Os conceitos ÁGUA, TERRA, VEGET1, VEGET2 e VEGET3 dão as interpretações possíveis a cada região. Pela maneira como elas são as sociadas, é possível que se tenha uma região com várias interpretações.

Para a Figura 4.2 (fora de escala), observá-se, para os controles de fração de tempo destinados à realização de medidas FRACMED e FRACMPED com os valores 0,6 e 0,3, respectivamente, o acesso às pro priedades P1, P2, P3 e P7, a partir da alteração correspondente ao va lor inicial para a instância de predicado ATIVO(1). Na utilização de P1, P2 e P3, o valor do predicado calculável $COR(x, intvi)$ é desconhecido e, como o cálculo de seu valor envolve a realização de medidas so bre a imagem, gera-se um pedido de realização de medidas que é empilha do, cuja propriedade correspondente tem sua utilização suspen sa. A utilização de P7 gera um novo valor da instância de predicado CANDIDATOSEP(1), com valor-verdade V; no processo de análise de compa tibilidade entre valores novos e antigos, executado após a utilização da propriedade, emprega-se uma heurística segundo a qual se mantém o valor mais recente.

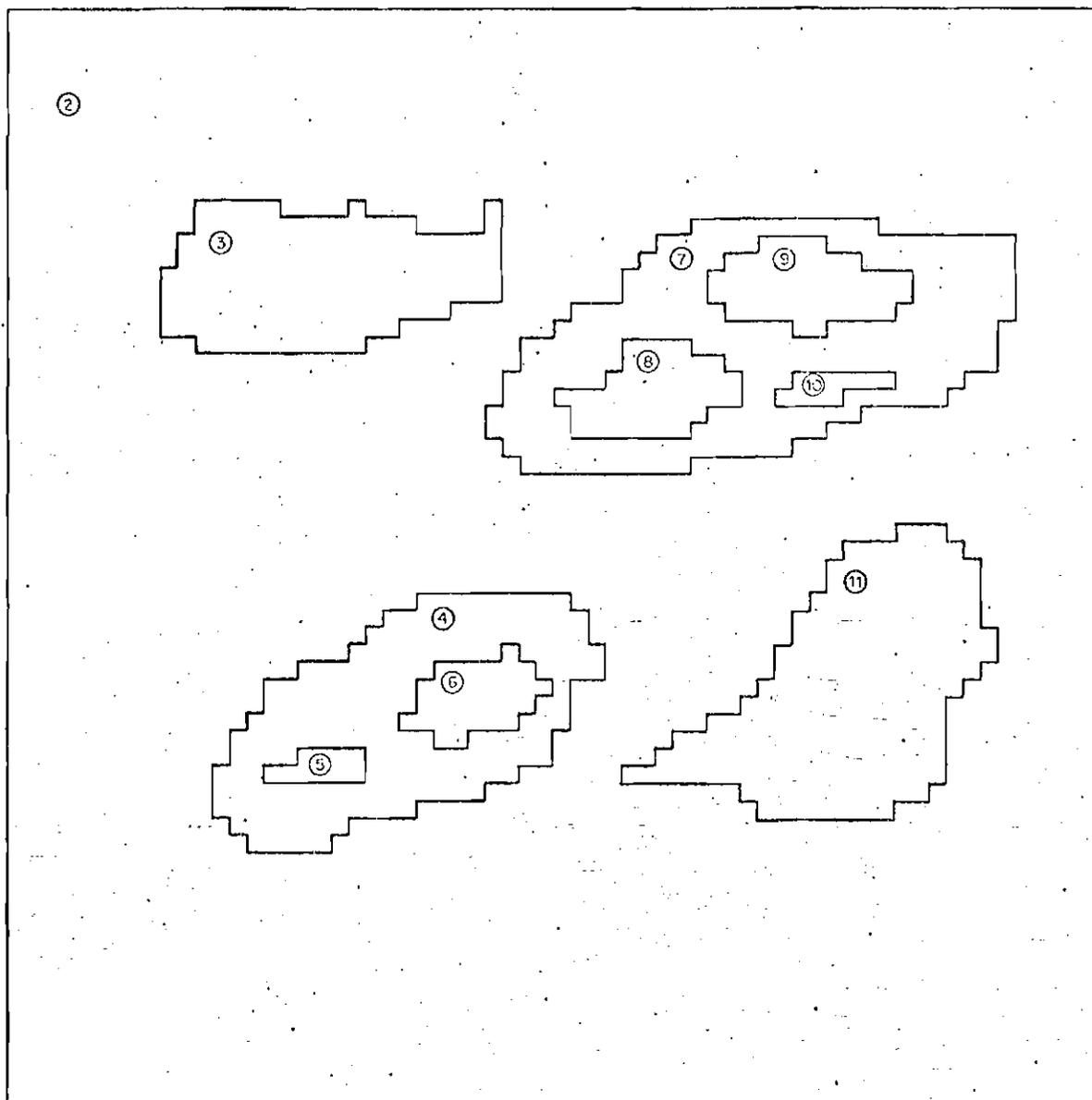


Fig. 4.2 - Cena para teste de separação.

4.5 - COMPARAÇÃO COM OUTROS MODELOS

4.5.1 - MODELOS DE REPRESENTAÇÃO

Serão apresentadas algumas vantagens e desvantagens da abordagem proposta para a representação de conhecimento em relação a alguns trabalhos relatados na literatura.

Inicialmente, pode-se arguir sobre a conveniência ou se melhança da representação baseada em conceitos com o tipo de represen tação utilizado por seres humanos. Sem levar em conta o problema de experimentos em psicologia (já mencionados na Seção 1.2), existem al gumas considerações do ponto de vista epistemológico. Aaron observa que pensar envolve conceitos, embora não se possa admitir, sem proble mas, que conceitos sejam o objeto do pensamento; não se pode afirmar que conceitos sejam impressão (quando se lembra de um objeto específico, não se trata do conceito correspondente, nem quando se vê tal objeto), nem que todos os conceitos sejam verbais.

No caso de conceitos verbais, estar-se-ia admitindo que conceitos são o objeto do pensamento conceitual e que as palavras de signam os conceitos; no caso de tradução de uma palavra de uma língua para outra, haveria um conceito envolvido (embora fatores culturais ou temporais possam alterar o sentido de palavras "correspondentes").

Uma alternativa seria considerar os conceitos como capa cidade ou propensão, o que esbarra na possibilidade de organizar ("frame") os conceitos; seguindo esta tendência, poder-se-ia pensar em uma capacidade, juntamente com um núcleo, usando termos predicativos (por exemplo, "--- é uma tirania"), acrescidos de disposição.

Voltando ao modelo proposto, são considerados conceitos aquelas entidades referenciadas especificamente por um nome (que deve ser entendido como uma designação interna; o que se dispõe numa aplica ção particular sobre o conceito é aquilo que foi especificado nas es

estruturas associadas e nas propriedades em que ele entra como entidade constituinte) e também aqueles expressos por propriedades. Os conceitos definidos com um nome podem ser usados como predicados (monádicos, nesta abordagem).

Desde a apresentação do modelo de Quillian (1968), diversas modificações e extensões foram acrescentadas a ele, embora ainda sejam observadas muitas de suas características. As modificações principais dizem respeito a um estudo mais detalhado das hierarquias que podem ser identificadas, à introdução de características vindas da Lógica, ao emprego de processos de inferência e à separação de características em diversos níveis. Não se têm "definições" do sentido de palavras (ou explicitação do conceito envolvido), mas diversas proposições em que os conceitos são elementos constituintes. As expansões de "esferas", a partir de nós iniciais também seria possível. A característica de permitir que um nó "token" seja utilizado como relação (relacionado a outros dois nós "token") não é reproduzida; os predicados devem ser definidos explicitamente como tais e não possuem organização em hierarquias.

Com relação ao modelo de Schubert (Schubert, 1976; Schubert et alii, 1979), as hierarquias de parte e de particularização são armazenadas separadamente aqui, o que facilita a transferência de estruturas e a sua visualização. Os nós das propriedades são de diversos tipos (predicado, conectivo, regra de decisão), o que tem correspondência com os ramos PRED e OP no modelo citado. As funções não recebem nós especiais na rede; são consideradas sempre como um tipo possível de argumento de predicados. No modelo sendo apresentado, a cada nó pode ser associado um valor-verdade, o que resulta em uma estrutura em árvore para os nós de uma propriedade; os demais elementos (variáveis, atributos, funções, etc.) destinam-se à associação de referentes apropriados. Não são empregados operadores modais e as funções de Skolem são montadas completas em relação às variáveis universais que dominam as existenciais; também não há tratamento para tempo, quantifi

cadores ou informações vagas. Não se efetua o compartilhamento de variáveis em formas normais; as transferências são executadas dinamicamente durante a utilização de propriedades e pode ocorrer de existirem diversos caminhos possíveis entre nós de conceito e de parte. Também não são empregadas hierarquias segundo tópicos para proposições.

Existem pontos em comum entre a abordagem proposta e os SI-Nets (Brachman, 1979; Brachman and Schmolze, 1985). A hierarquia de partes corresponde aos "papéis", sendo a relação de parte dada, no modelo citado, pelo conceito apontado por "restrição"; o "número" de entidades que satisfaçam ao papel é expresso por uma propriedade associada ao conceito e à parte correspondente. Não foi implementado o que seria o correspondente ao "nome"; deveria ser possível atribuir um nome diferente a um conceito, quando ele é empregado como parte de outro. A "modalidade", quando obrigatória, é representada por uma propriedade. As "relações estruturais" podem ser dadas diretamente por propriedades, associadas ao conceito e suas partes. A relação estrutural S3, da Figura 1.5, representa uma maneira de calcular a altura do vão de um arco; esta atribuição não pode ser feita diretamente no modelo proposto. Uma maneira de efetuar a atribuição seria fazer uso de um predicado calculável que atribuiria à função DISTANCE seu valor, a partir de argumentos conhecidos. Quanto aos tipos de relação, em vez de "individualização", usa-se "particularização"; "instanciação" permanece como relação entre conceito e objeto do mundo real; a relação "denotação" não é empregada aqui - prefere-se utilizar sempre instanciação como relação entre conceito e objeto, observando-se que sempre é possível ter mais de uma hipótese associada a um objeto. Quanto à transferência de estruturas, aquilo que é diferente entre um conceito e suas particularizações pode ser encontrado pesquisando as estruturas correspondentes e as propriedades estabelecidas para uma particularização; não se mantêm, então, explicitamente, todas as modificações efetuadas. No modelo proposto, um conceito pode ser particularização de apenas um outro. Não há menção ao tratamento de hipóteses conflitantes no modelo citado.

A abordagem de Fox (1979) para a representação de condições de exceção permite que elas estejam presentes, de forma explícita, na própria representação. No modelo proposto, procura-se evitar uma subdivisão exaustiva de hierarquias pelo emprego de propriedades conflitantes.

Existem poucas semelhanças, quanto ao aspecto de implementação, entre o modelo exposto e o de Hendrix (1976, 1979), visto que não se utilizam os conceitos de espaços, supernós e quantificadores implícitos. A linguagem de acesso à rede utilizada aqui é, devido a estas diferenças, bastante diferente. Quanto ao poder de expressão, elas são comparáveis sob diversos aspectos; isto pode permitir que sejam introduzidas, em extensões futuras, características semelhantes às do sistema PROSPECTOR.

Algumas extensões devem ser feitas ao modelo proposto de modo a torná-lo compatível com outros descritos na literatura, voltados para aplicações específicas (MYCIN, PROSPECTOR, SI-Nets, etc.) ou com os "esqueletos" correspondentes.

Com relação ao MYCIN (Shortliffe, 1976; Buchanan and Shortliffe, 1984), não foi implementada, no mecanismo de acesso atual, a obtenção de mapeamentos a partir de atributos. O mecanismo de controle, através da estrutura de subproblemas, permite o encadeamento retroativo de regras de decisão. O emprego de meta-regras poderá ser simulado através das heurísticas destinadas à escolha de ação do próximo mapeamento ou subproblema a ser considerado.

Para o caso do PROSPECTOR, a introdução de informações iniciais é possível através da rotina para a inserção de elementos iniciais no banco de hipóteses (atualmente, emprega-se um procedimento compilado separadamente; outra possibilidade seria através de uma linguagem de comandos específicos). Para os graus de necessidade e de suficiência, assim como para os mecanismos para manuseá-los, seria ne

cessária uma alteração na maneira como se armazena o grau associado a uma regra (lista, em vez de número) e no cálculo de valores atualizados.

A eventual adequação do modelo proposto às aplicações desenvolvidas em outras áreas pode torná-lo suscetível de ser empregado em sistemas de propósito mais geral, como os previstos para a próxima geração de computadores (Treleaven and Lima, 1982).

Dada a semelhança entre a forma como se expressam propriedades e fórmulas no cálculo de predicados de primeira ordem, poderia ser tentado um processo de avaliação ou dedução, proveniente do cálculo de predicados. Existem procedimentos de prova automática de teoremas que visam evitar o uso de dados irrelevantes ou a resolução de problemas sem sentido e que envolvem extensões visando o uso de estruturas como as empregadas neste trabalho, de partes e de particularizações (McSkimin and Minker, 1979; Chouraqui, 1981). Por exemplo, se se procura determinar o estado x tal que $REPRESENT(Sirica, x)$ seja satisfeito, onde o primeiro argumento do predicado deve ser da classe SENATOR ou REPRESENTATIVE, não é possível encontrar uma solução, dado que o indivíduo Sirica é da classe JUDGE.

Por outro lado, são aproveitadas algumas características dos teoremas tipo antecedente e conseqüente do PLANNER (Hewitt, 1971), para verificar conseqüências de fatos ou responder perguntas.

Existe uma série de características que foram tomadas como desejáveis no desenvolvimento do modelo apresentado, as quais não permitem o emprego de mecanismos de prova automática de teoremas. Podem-se citar a eventual presença de propriedades conflitantes, o emprego de regras de decisão de uma maneira tal que os mecanismos de acesso

e de utilização são basicamente os mesmos que para propriedades em geral, o controle passo a passo da propagação de resultados, a geração de subproblemas quando uma propriedade é acessada para responder uma pergunta na qual algum dos referentes é desconhecido, o emprego de valores-verdade no intervalo $[-1,1]$ e o tratamento de incompatibilidades entre valores novos e antigos do banco de hipóteses, que pode resultar na reutilização de propriedades.

4.5.2 - MODELOS DE ANÁLISE DE CENAS

O tipo de abordagem proposto apresenta diversas vantagens quanto ao teste de algoritmos que utilizem o conhecimento sobre um universo. O mecanismo de controle, descrito no Capítulo 3, permite que se introduzam heurísticas em diversos pontos.

Em relação ao que um ser humano realiza em tarefas semelhantes, existem muitos autores que concordam com o fato de que, num nível mais alto de processamento, emprega-se um conhecimento específico sobre um objeto ou classes de objetos para remover ambigüidades ou suprir elementos que estejam faltando.

Quanto ao processamento a um nível mais baixo ("low-level"), existem aqueles que questionam a utilização de conhecimento sobre um tipo específico de objeto para suprir partes não-presentes de um contorno ou eliminar elementos espúrios (Waltz, 1978).

Na abordagem presente, leva-se em conta o fato que podem existir muitas tarefas do sistema visual humano que podem ser executadas independentemente do domínio, embora algumas delas possam ser executadas utilizando um mecanismo de propagação de fatos conhecidos num meio como uma rede associativa.

As diversas medidas que podem ser efetuadas são, em geral, bastante caras em termos de recursos computacionais exigidos. Exis

tem propostas de utilizar um "hardware" especial para alguns tipos de medidas, que poderão levar a uma otimização dos processos hoje realizados. Além disso, parte do processo associativo também poderá ser realizado por um hardware" específico. No entanto, pode estar sendo formado um otimismo exagerado em relação a essas possibilidades devido a uma eventual interação entre os diversos módulos que seriam característicos num modelo.

A abordagem de Preparata e Ray (1972) considera a existência de apenas um domínio a que todas as entidades estão associadas, em preta normalização de resultados obtidos de medidas e tem a busca de interpretações totalmente direcionada pelo mapa semântico, além de considerar a segmentação fornecida. Estas limitações tornam o modelo inadequado; trabalhos posteriores tentaram removê-las, assim como expandir algumas das idéias expostas.

Quanto ao ciclo de percepção proposto por Havens e Mackworth (1980, 1983), que utiliza uma representação do tipo procedural, não leva em conta a possibilidade da percepção também possuir um componente errático, ou seja, observações que não forem solicitadas podem causar um redirecionamento na busca de interpretações. Além disso, a utilização do contexto não é considerada como uma simples heurística, conforme considerada na presente proposta. Havens (1978) critica a utilização das redes de similaridade (Winston, 1975; Minsky, 1975; Freuder, 1976) que, num processo de reconhecimento direcionado por hipóteses, causam o acesso a um "schema" substituto quando um não é aplicável a uma situação; estas redes deveriam ser suficientemente completas de modo a evitar falhas inexplicáveis. O autor citado propõe que problemas de busca "top-down" devem ser evitados pela utilização de evidências para a seleção de hipóteses adequadas. Além da busca direcionada por dados, propõe-se que as redes de similaridade podem ser substituídas por mecanismos mais gerais que não estejam baseados na geração prévia de todas as diferenças possíveis entre "schemata". Uma heurística possível de busca é aquela em que, dado que uma parte não foi encon

trada, para um dado conceito (ou que um atributo não tenha um certo valor esperado), procura-se investigar outro conceito que possua maior sobreposição com aquele em que ocorreu o problema. Outra característica diferente é a possibilidade de não necessariamente ter de esperar pelo completo reconhecimento de uma instância de um "schema" para tomá-lo como evidência para a pesquisa sobre outro em nível mais alto.

Ohta (1980) critica a segmentação direcionada pela interpretação, que segue um esquema "bottom-up", para o caso de múltiplas regiões separadas e que devem ter a mesma interpretação; propõe um uso misto com um esquema "top-down" (na forma de um sistema de produção). A partir de uma segmentação preliminar (que pode ser em excesso), são atribuídas interpretações aos segmentos os quais formam "planos" que podem ser usados no processo "top-down". O uso de regras torna fácil a modificação do conhecimento armazenado (são equivalentes aos teoremas "antecedente" e "conseqüente" do PLANNER (Hewitt, 1971)). A análise, também neste caso, é totalmente dominada por esta composição de processos. Os controles não são explícitos; conforme o tipo de regra, ações são tomadas num sentido ou noutro, gerando planos ou descrições da cena a partir de condições na base de dados. No trabalho mencionado, diversos atributos e operações não estão completamente definidos; é possível, no entanto, reproduzir parcialmente o conjunto de operações que ele especifica através de propriedades, em que se definem predicados calculáveis para o cálculo de atributos, funções e características das interpretações associadas.

Quanto às estruturas que possam facilitar o manuseio da imagem (como estruturas em pirâmide, cilindros generalizados, etc.), elas podem ser eventualmente adicionadas de acordo com os algoritmos que atuam sobre a imagem; é necessária a alteração das estruturas de dados relativas à imagem e à descrição da cena.

O tipo de representação empregado na abordagem proposta, basicamente do tipo declarativa (embora os códigos associados a predica

dos, atributos ou conceitos possam eventualmente alterar o controle), tem uma série de vantagens quanto à compactação da representação, à possibilidade de haver ordem explícita entre as informações e à separação entre representação e controle. Uma desvantagem é que alterações mais profundas no controle (ou modelo de percepção) devem ser introduzidas no esquema já descrito. A representação procedural do processo de percepção, como a de Havens (1978), exige, por sua vez, que todas as ações de controle sejam fornecidas explicitamente para cada aplicação, o que pode gerar especificações de controle bastante longas, complexas e difíceis de modificar; no entanto, pode apresentar maior eficiência computacional.

CAPÍTULO 5

CONCLUSÕES

5.1 - RESULTADOS OBTIDOS

Com a motivação inicial de desenvolver um modelo de representação e utilização de conhecimento que fosse adequado às necessidades encontradas no processo de análise de cenas, tanto a um nível mais alto quanto ao nível de interação com processos de medida, chegou-se a um modelo que poderá vir a ser adaptado para outras áreas de aplicação.

Inicialmente, foram estabelecidas as estruturas que se destinam ao armazenamento do conhecimento prévio disponível sobre um domínio. Estas estruturas, descritas no Capítulo 2, comportam a representação de conceitos, hierarquias de partes e particularizações, predicados, atributos, funções e propriedades, que podem ser do tipo fórmulas do cálculo de predicados de primeira ordem ou regras de decisão.

Para as regras de decisão, propõe-se uma forma em que expressões quantificadas compõem tanto a premissa como a conclusão. A inter-relação entre a premissa e a conclusão é feita através de variáveis que são comuns a elas.

Para as hierarquias de partes e de particularizações, propõe-se um tipo de inferência de relações de parte, que pode simplificar a forma como se expressa o conhecimento através de propriedades e reduzir a quantidade de relações necessárias; a expansão da hierarquia de partes somente é feita, durante uma aplicação, se for constatada a sua necessidade. Este tipo de inferência é tal que pode-se deixar de aceitar seu resultado por contradizer uma relação oposta fornecida explicitamente, ou pode-se chegar a relações conflitantes. No Apêndice B, mostram-se os algoritmos empregados e exhibe-se uma prova de sua convergência.

Na entrada de predicados e atributos, define-se o nome pelo qual eles podem ser referenciados em propriedades, os números de argumentos e, quando são calculáveis, um número interno que indica a rotina que deve ser calculada para encontrar seu valor (a escolha destas rotinas pode ser alterada após a entrada, seja alterando um dado, o número interno, ou compilando separadamente a rotina que efetua as chamadas, com outro nome de rotina de cálculo para o mesmo número interno).

O mecanismo pelo qual as informações armazenadas podem ser utilizadas durante uma aplicação específica, assim como a maneira como se inter-relacionam os processos voltados ao tratamento deste tipo de informação e aqueles voltados à análise de cenas, é descrito no Capítulo 3. São definidas as estruturas que são empregadas para representar as entidades instanciadas, assim como as destinadas ao controle - especificamente, as listas que armazenam alterações a propagar e problemas a resolver.

O emprego destes dois tipos de estruturas permite que se encadeiem regras de decisão (ou propriedades, em geral) tanto de maneira progressiva como retroativa. Assim, a partir de um conjunto de regras fornecidas de início, a maneira como elas serão empregadas irá depender das informações disponíveis sobre um problema em particular.

Tanto para a propagação de alterações como para a resolução de problemas, procura-se encontrar propriedades que possam ser utilizadas. O algoritmo para o acesso a propriedades emprega informações fornecidas de início e outras que possam estar presentes no banco de hipóteses para encontrar mapeamentos entre elementos e nós de propriedades, e referentes de variáveis e de nós intermediários em caminhos formados por relações de parte. O algoritmo para a geração de indicadores permite que se recuse o emprego de propriedades quando há conflito entre os valores que seriam necessários e os disponíveis e quando não é possível efetuar inferências. O algoritmo para a utilização de

propriedades efetua processos de busca no banco de hipóteses de modo a satisfazer expressões quantificadas e pode gerar novas alterações a serem introduzidas.

Problemas podem ser fornecidos de maneira externa ao fluxo de controle (através de heurísticas para a determinação da próxima ação a ser executada) e gerados durante o acesso ou durante a utilização de propriedades (quando houver permissão para tal). Após a utilização de propriedades, diversas condições são verificadas, tais como a existência de problemas resolvidos, de conflitos entre valores do banco de hipóteses, de geração ou alteração de hierarquias, etc.

Para a aplicação ao domínio da análise de cenas, foram definidas as estruturas destinadas ao armazenamento de descrições de uma cena, as segmentações realizadas, como estão inter-relacionadas e as razões pelas quais foram realizadas.

Define-se um mecanismo de geração de medidas espontâneas, através do qual podem ser geradas operações sobre uma cena, independentemente de um controle estabelecido através da utilização de propriedades. Este mecanismo tem associado uma memória local sobre as regiões em que foram realizadas as últimas operações.

Foram desenvolvidas duas aplicações para exibir as capacidades de representação e de controle do modelo. Na primeira delas, o objetivo é reproduzir os resultados de um processo de propagação de restrições, voltado para o problema de fusão de regiões de uma cena. As condições de compatibilidade são expressas por regras de decisão, em que uma ou mais interpretações, dadas como verdadeiras de início, tentativamente, podem ser negadas. Na segunda aplicação, mostra-se como mecanismo de geração de medidas espontâneas atua, assim como o emprego de uma mesma operação, de separação de regiões, tanto de maneira espontânea como solicitada por propriedades.

5.2 - EXTENSÕES FUTURAS

O modelo de representação e a implementação realizada comportam uma série de adições, que poderiam expandir sua área de aplicação; algumas delas são:

- a) Ter um compartilhamento efetivo de parcelas comuns de diversas propriedades; esta adição envolve a montagem de listas de referências para todos os nós de uma propriedade, de modo que se possa saber, para cada nó, a que propriedades pertence (para que se possa efetuar retiradas ou substituições de propriedades). Envolve, também, um procedimento mais complexo para a entrada de propriedades, pois seria necessário pesquisar, dentre as já existentes, se em algumas delas ocorre um isomorfismo entre as subestruturas.
- b) Ter uma aglutinação de fatos relacionados a um conceito, alterando, se desejável, o acesso a propriedades - na situação atual, o acesso é realizado diretamente a partir de nós nas listas de referência.
- c) Ter uma estimativa de custo de realização de medidas, que influencie a ordem de utilização das propriedades e de percurso de nós de uma propriedade.
- d) Empregar o algoritmo de inferência de caminhos na entrada de propriedades e de relações de parte e de particularização. Para o caso de propriedades, estaria sendo verificado um eventual conflito ou inexistência de caminho a partir das hierarquias que estivessem presentes até o instante da entrada da propriedade. Para as relações, seria necessário uma alteração mais profunda nos algoritmos, visto que os atuais supõem a existência de um dígrafo auxiliar, montado a partir de nós de uma propriedade.

- e) Implementar o tratamento de variáveis quantificadas numericamente, no processo de utilização de propriedades, no que se refere às condições de fim de processo de busca de referentes para variáveis e ao tratamento de referentes encontrados.

Outras extensões, como as comentadas a seguir, poderiam exigir alterações mais profundas nas estruturas de armazenamento e nos algoritmos:

- a) Ter a representação, no banco de hipóteses, de fórmulas não-atômicas; esta adição exigiria a alteração de mecanismos de propagação de resultados e de mapeamento entre elementos e nós de propriedade.
- b) Ter a possibilidade de mais de uma generalização por conceito; os algoritmos que fazem os percorrimentos dependeriam de todos os caminhos entre um nó e seus ancestrais.

A linguagem de acesso à rede (vide Apêndice A) fornece uma maneira fácil de introduzir elementos das hierarquias e propriedades na rede. Outras facilidades podem ser adicionadas, para se ter a inserção de informações iniciais e da escolha de ação a ser executada, e nos diversos pontos onde é possível introduzir heurísticas. Uma maneira de facilitar esta comunicação seria, por exemplo, através de uma linguagem de comandos, específica para cada caso e que resumiria as opções possíveis.

Já foram feitos, na Seção 4.5.1, comentários sobre as alterações que seriam necessárias de modo que o modelo de representação possa ser adaptado às aplicações de outros modelos, mencionados na literatura (MYCIN, PROSPECTOR, SI-Nets). Na Seção 4.5.2 foram comentadas, também, as relações entre o modelo de análise de cenas e outros existentes.

A aplicação ao domínio de análise de cenas pode ser expandida, de modo a considerar, por exemplo, outros tipos de operações que seriam acessíveis para a determinação de medidas espontâneas.

Uma extensão interessante de ser pesquisada seria a obtenção de eixos, mencionados por Nishihara (1983), e a investigação de como se poderia chegar à obtenção de primitivas para uma utilização de informações sintáticas, sem que o programa esteja direcionado para isto.

Existem ainda muitos fenômenos não explicados no processo de percepção visual humana, e um direcionamento no sentido de investigar, de maneira geral, a interação entre processos puramente simbólicos e puramente pictóricos pode ser produtivo no sentido de tentar reproduzir o comportamento deste processo.

Ainda com relação ao domínio de aplicação escolhido, pode ser tentada uma adaptação do modelo para outros domínios, como o de reconhecimento de fala contínua. A modularização das estruturas e dos algoritmos voltados para o domínio específico e para a representação de conhecimento tende a facilitar este tipo de adaptação.

REFERÊNCIAS BIBLIOGRÁFICAS

- AKIN, O.; REDDY, R. Knowledge acquisition for image processing research. *Computer Graphics and Image Processing*, 6(4):307-334, 1977.
- BARROW, H.G.; TENENBAUM, J.M. *MSYS: a system for reasoning about scenes*. Menlo Park, CA, Stanford Research Institute, 1976. (Tech. Note 121).
- BIEDERMAN, I. Perceiving real world scenes. *Science*, 177(4043): 77-80, 1972.
- BRACHMAN, R.J. On the epistemological status of semantic networks. In: FINDLER, N.V., ed. *Associative networks; representation and use of knowledge by computers*. New York, NY, Academic, 1979. p. 3-50.
- BRACHMAN, R.J.; SCHMOLZE, J.G. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171-216, 1985.
- BRADY, J.M. The changing shape of computer vision. *Artificial Intelligence*, 17:1-15, 1981. Special Volume on Computer Vision.
- BUCHANAN, B.G.; SHORTLIFFE, E.H. *Rule-based expert systems; the MYCIN experiments of the Stanford Heuristic Programming Project*. Reading, MA, Addison, 1984.
- CARNAP, R. *Meaning and necessity*. 2.ed. Chicago, The University of Chicago Press, 1970.
- CHOURAQUI, E. *Contribution a l'étude théorique de la représentation des connaissances; le système symbolique ARCHES*. These de Docteur d'Etat. s.l. Institut National Polytechnique de Lorraine. Oct. 1981.
- CLOWES, M.B. On seeing things. *Artificial Intelligence*, 2:79-116, 1971.
- DORIA, C.S. *Psicologia científica geral*. Rio de Janeiro, Agir, 1968.

- 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.
- DUDA, R.O.; HART, P.E.; NILSSON, N.J.; SUTHERLAND, G.L. Semantic network representation in rule-based inference systems. In: WATERMAN, D.A.; HAYES-ROTH, F., ed. *Pattern-directed inference systems*. New York, NY, Academic, 1978. p. 203-221.
- FISCHLER, M.A. On the representation of natural scenes. In: HANSON, A.R.; RISEMAN, E.M., ed. *Computer vision systems*. New York, NY, Academic, 1978. p.47-52.
- FORGY, C.L. *OPS5 user's manual*. Pittsburgh, PA, Dept. of Comp. Sci., Carnegie-Mellon Univ., 1981.
- FOX, M.S. On inheritance in knowledge representation. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979, *Proceedings*. s.l., IJCAI, 1979, p. 282-284.
- FREUDER, E.C. *A computer system for visual recognition using active knowledge*. Ph.D. Thesis, Cambridge, MA, Art. Intell. Lab., MIT, 1976.
- GASCHNIG, J. *Development of uranium exploration models for the PROSPECTOR consultant system*; final report. Menlo Park, CA, Stanford Research Institute, 1980.
- HAVENS, W.S. A procedural model of recognition for machine perception. In: BIENNIAL CONFERENCE OF THE CANADIAN SOCIETY FOR COMPUTATIONAL STUDIES ON INTELLIGENCE, 2., Ontario, ON, 1978. *Proceedings*, Ontario, CSCSI, 1978, p. 244-253.
- HAVENS, W.S.; MACKWORTH, A.K. Schemata-based understanding of hand-drawn sketch maps. In: BIENNIAL CONFERENCE OF THE CANADIAN SOCIETY FOR COMPUTATIONAL STUDIES ON INTELLIGENCE, 3., Victoria, BC, 1980. *Proceedings*, Victoria, CSCSI, 1980, p. 172-178.

- HAVENS, W.S.; MACKWORTH, A.K. Representing knowledge of the visual world. *Computer*, 16(10):90-98, Oct. 1983.
- HENDRIX, G.G. The representation of semantic knowledge. In: WALKER, D., ed. *Speech understanding research; final report*. Menlo Park, CA, Stanford Research Institute, 1976.
- HENDRIX, G.G. Encoding knowledge in partitioned networks. In: FINDLER, N.V., ed. *Associative networks; representation and use of knowledge by computers*. New York, NY, Academic, 1979. p. 51-92.
- HEWITT, C. *Description and theoretical analysis (using schema) of PLANNER: a language for proving theorems and manipulation models in a robot*. Ph.D. Thesis, Cambridge, MA, Art. Intell. Lab., MIT, 1971.
- MACKWORTH, A.K. How to see a simple world: an exegesis of some computer programs for scene analysis. In: ELCOCK, E.W.; MICHIE, D., ed. *Machine Intelligence, 8*. Chichester, Ellis Horwood, 1977. p. 510-637.
- McKEAN, K. In search of the unconscious mind. *Discover*, 6(2):12-18, Feb. 1985.
- McSKIMIN, J.R.; MINKER, J. A predicate-calculus based semantic network for deductive searching. In: FINDLER, N.V., ed. *Associative networks; representation and use of knowledge by computers*. New York, NY, Academic, 1979. p. 205-238.
- MINSKY, M. A framework for representing knowledge. In: WINSTON, P. H., ed. *The psychology of computer vision*. New York, NY, McGraw, 1975. p. 211-280.
- MULDER, J.A.; MACKWORTH, A.K. Using multi-level semantics to understand sketches of houses and other polyhedral objects. In: BIENNIAL CONFERENCE OF THE CANADIAN SOCIETY FOR COMPUTATIONAL STUDIES ON INTELLIGENCE, 2., Ontario, ON, 1978. *Proceedings*. Ontario, ON, CSCSI, 1978, p. 244-253.

- NAVON, D. Forest before trees: the precedence of global features in visual perception. *Cognitive Psychology*, 9:353-383, 1977.
- NILSSON, N.J. *Principles of artificial intelligence*. Palo Alto, CA, Tioga, 1980.
- NISHIHARA, H.K. Recognition of shape in visible surfaces. In: BRADDICK, O.J.; SLEIGH, A.C., ed. *Physical and biological processing of images*. Berlin, Springer, 1983. p. 335-348.
- OHTA, Y. *A region-oriented image-analysis system by computer*. Dr.Eng. Thesis. Kyoto, Dept. of Info. Sci., Kyoto Univ., Mar. 1980.
- OHTA, Y.; KANADE, T.; SAKAI, T. A production system for region analysis. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979. *Proceedings*. s.l., IJCAI, 1979, p. 684-686.
- PREPARATA, F.P.; RAY, S.R. An approach to artificial nonsymbolic cognition. *Information Sciences*, 4(1):65-86, 1972.
- QUILLIAN, M.R. Semantic memory. In: MINSKY, M., ed. *Semantic information processing*. Cambridge, MA, MIT Press, 1968. p.227-270.
- REDER, L.M.; ANDERSON, J.R. Use of thematic information to speed search of semantic nets. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979. *Proceedings*. s.l., IJCAI, 1979, p. 708-710.
- RIEGER III, C.J. Conceptual memory. In: SCHANK, R.C. *Conceptual information processing*. Amsterdam, North-Holland, 1975, Cap. 5, p.157-288.
- SCHUBERT, L.K. Extending the expressive power of semantic networks. *Artificial Intelligence*, 7(2):163-198, 1976.
- SCHUBERT, L.K.; GOEBEL, R.C.; CERCONE, R.J. The structure and organization of a semantic net for comprehension and inference. In: FINDLER, N.V., ed. *Associative networks; representation and use of knowledge by computers*. New York, NY, 1979. p. 122-175.

- SHORTLIFFE, E.H. *Computer-based medical consultations: MYCIN*. New York, NY. American Elsevier, 1976.
- SIMONI, P.O.; SOUZA, C.R. Encadeamento retroativo e progressivo de regras de decisão num sistema de representação por redes associativas. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 2., São José dos Campos, INPE, 1985, *Anais*. São José dos Campos, INPE, 1985, p. 131-132.
- TENENBAUM, J.M.; BARROW, H.G. *Experiments in interpretation-guided segmentation*. Menlo Park, CA, Stanford Research Institute, 1976 (Tech. Note 123).
- TRELEAVEN, P.C.; LIMA, I.G. Japan fifth-generation computer systems. *Computer*, 15(8):79-88, 1982.
- WALTZ, D.L. Understanding line drawings of scenes with shadows. In: WINSTON, P.H., ed. *The psychology of computer vision*. New York, NY, McGraw, 1975. p. 19-91.
- WALTZ, D.L. A parallel model for low-level vision. In: HANSON, A.R.; RISEMAN, E.M., ed. *Computer vision systems*. New York, NY, Academic, 1978. p. 175-186.
- WINSTON, P.H. Learning structural descriptions from examples. In: WINSTON, P.H., ed. *The psychology of computer vision*. New York, NY, McGraw, 1975. p. 157-209.
- ZEIDEL, D.W. Les fonctions de l'hémisphère droit. *La Recherche*, 15(153):332-340, 1984.

APÊNDICE A

LINGUAGEM DE ACESSO À REDE ASSOCIATIVA

São definidos quatro tipos de comandos na linguagem empregada para alterar informações da rede associativa: para a entrada; para a saída de propriedades; para alteração de estruturas, dados numéricos, atributos, predicados e funções de predicado; e para a exibição do conteúdo das listas que armazenam informações (estáticas, que são definidas antes de qualquer utilização num processo de análise de cenas; as informações relativas a uma aplicação são acessadas no decorrer do processo de análise).

a) entrada de propriedades

```
<propriedade> ::= <regra-de-decisão>/
                <fórmula>
<regra-de-decisão> ::= <quant.univ.><seq.quant.>[<arg.conec> RGP <peso>
                <arg.conec>]/
                <quant.univ.>[<arg.conec.> RGP <peso>< arg.conec>]
<peso> ::= (1.) /
            (0.<inteiro>)
<fórmula> ::= <form.quant.>/
            <form.com conec.>
<form.com conec.> ::= <arg.conec><conectivo><arg.conec>
<arg.conec> ::= <form.quant.>/
            [<form.com conec.>]/
            <literal>
<form.quant> ::= <seq.quant.>[<literal>]/
            <seq.quant.>[<form.com conec.>]
<seq.quant.> ::= <quantificador>/
            <seq.quant.><espaço><quantificador>
```


em que

- não há precedência pré-estabelecida entre conectivos; o escopo é determinado por pares de colchetes;
- os conectivos E, OU, e EQV podem, na atual implementação, possuir até 3 repetições sem colchetes para agrupá-los;
- se um conjunto de predicados afetados por conectivos estiver no escopo de um quantificador, não deve ser usado o par de colchetes relativo ao conectivo;
- todas as variáveis referenciadas devem ser quantificadas e um nome de variável só pode ocorrer em um quantificador em cada propriedade;
- os nomes de funções de Skolem de variáveis quantificadas existenciais ou numericamente são gerados automaticamente;
- o número máximo de ocorrências de uma variável quantificada numericamente deve ser positivo; além disso, este número máximo não pode ser menor que o número de variáveis sob o mesmo quantificador;

Além disso, pode ser observado da sintaxe que:

- seqüências de quantificadores não devem ter colchetes internos de separação, a menos que existam conectivos na expressão quantificada;
- a negação só pode ser aplicada diretamente a um nome de predicado, o que não diminui o poder de expressão da representação e facilita a escrita de alguns algoritmos.

A partir de um comando de entrada, identifica-se, inicialmente, o tipo de comando e, à medida que ele é percorrido criam-se estruturas necessárias na rede em caráter provisório; se a propriedade não é aceita, elas são apagadas.

b) alteração de listas de informações

```
<alt.lista>::=<numero.lista><tipo alt.><seq.dados>
<tipo alt>::=ENT/
           SAI/
           MOD
<número lista>::=<inteiro>
<seq.dados>::=<dado lista>/
           <seq.dados><espaço><dado lista>
<dado lista>::=<inteiro>/
           <string alpha.>
```

em que:

- o número da lista designa o tipo de informação da lista específica (conceito, relação de parte e de particularização, atributo, predicado, etc);
- para o comando ENT, tem-se um conjunto de informações associado a cada lista (nome, número de argumentos, número interno de rotina que calcula o valor, no caso de predicado ou atributo, etc);
- para o comando SAI, tem-se associado o número interno do nó da lista a ser eliminado;
- para o comando MOD, tem-se também um conjunto de informações associado a cada lista e o número interno do nó a ser modificado.

c) saída de propriedades

<inteiro>

em que:

- o inteiro designa o número interno da propriedade a ser eliminada.

d) exibição de conteúdos de listas e propriedades

<exib>::=<número lista>(<nō inicial><espaço><nō final>)

<nō inicial>::=<inteiro>

<nō final>::=<inteiro>

em que

- o número da lista designa o tipo de informação ou propriedade.

APÊNDICE B

DETERMINAÇÃO DE CAMINHOS ENTRE NÓS

Para a determinação de caminhos, através de relações de parte, entre um nó de uma propriedade e outros nós que representam partes do conceito correspondente ao primeiro nó, monta-se, inicialmente, um dígrafo (chamado dígrafo auxiliar) correspondente às ligações entre os nós de conceito e os de parte. Um nó correspondente a um nó de conceito aponta para um correspondente a um nó de parte se a primeira variável do nó de parte for a variável do nó de conceito; um nó correspondente a um nó de parte aponta para outro se a segunda variável do primeiro nó for a primeira do segundo nó.

Este dígrafo auxiliar irá conter, em geral, mais de um componente conectado. No entanto, todos os nós deste dígrafo que responderem aos nós de parte da propriedade, e pelo menos um nó correspondente a um nó de conceito, devem estar no mesmo componente. Esta restrição é introduzida visando simplificar o algoritmo de geração de caminhos. Outra restrição é que, dentre os nós de conceito do componente que contém os nós de parte, todos devem estar ligados aos mesmos nós de parte; eles não precisam estar numa mesma hierarquia de particularizações.

Na Figura B.1, tem-se o dígrafo auxiliar correspondente à propriedade e às estruturas do exemplo da Seção 2.5; as variáveis associadas a cada nó do dígrafo são aquelas dos nós correspondentes da propriedade.

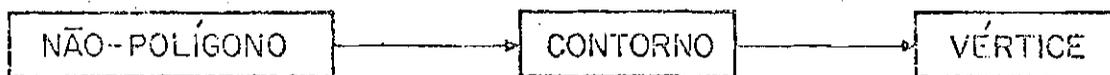


Fig. B.1 - Dígrafo auxiliar para o primeiro exemplo.

Na Figura B.2, tem-se o dígrafo auxiliar para a propriedade de ca hierarquia a seguir

B PARTE-DE A

$\forall x \forall y \forall z [A(x) \text{ E } B(x,y) \text{ E } B(x,z)]$

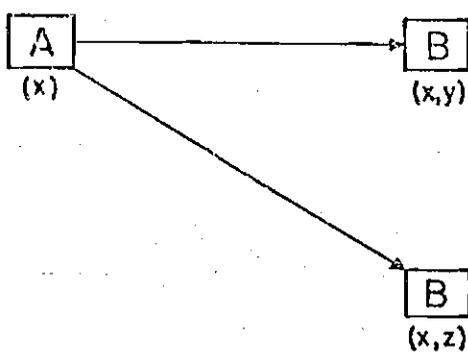


Fig. B.2 - Dígrafo auxiliar para o segundo exemplo.

Outro exemplo é o da Figura B.3, para a hierarquia e a propriedade dadas a seguir:

D PARTE-DE A

D PARTE-DE B

F PARTE-DE B

D PARTE-DE C

F PARTE-DE C

G PARTE-DE D

H PARTE-DE F

I PARTE-DE H

$\forall x \forall y \forall z \forall u \forall v \forall w \forall t \forall s$

$[[[A(x) \text{ E } B(y) \text{ E } C(y) \text{ E } D(z)] \text{ E } [G(u) \text{ E } H(w)]] \text{ --->}$
 $[[D(x,z) \text{ E } D(y,z) \text{ E } F(y,v) \text{ E } G(z,u)] \text{ E}$
 $[H(v,w) \text{ E } H(v,t) \text{ E } I(w,s)]]]$

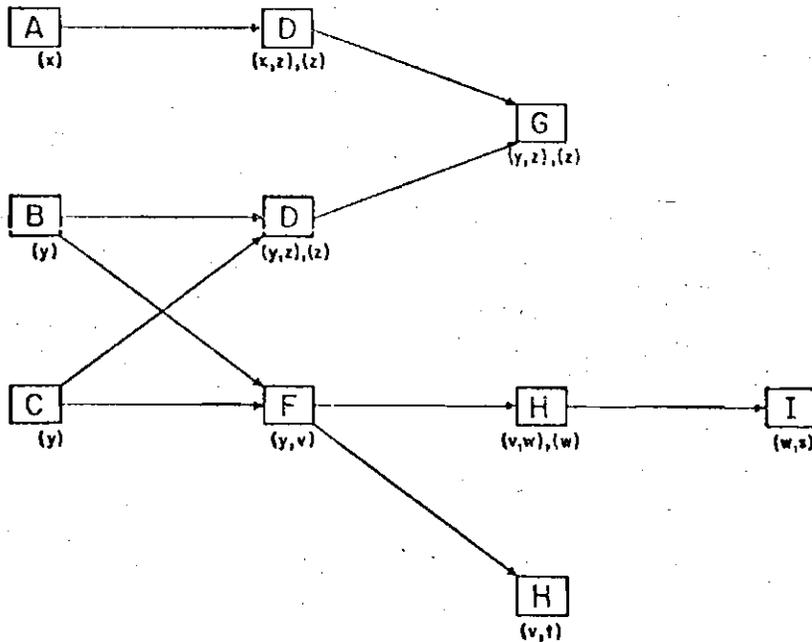


Fig. B.3 - Dígrafo auxiliar para o terceiro exemplo.

Outro dígrafo que é montado, após o dígrafo auxiliar, é chamado dígrafo de relações de parte. Neste dígrafo, os nós representam conceitos e os ramos são relações de parte; as relações de parte são aquelas fornecidas na estradas das hierarquias e, também, aquelas

inferidas pelas condições de transferência (a) - (c) mostradas na Seção 2.3 (Figura 2.3). Através deste dígrafo, pode-se pesquisar a existência de caminhos entre um nó inicial e um final especificado. Para cada nó, existe uma indicação se é nó inicial, final ou nenhum deles, indicação se já foi percorrido como particularização e se já armazenou partes, apontadores para generalização, primeira parte e primeiro elemento de que é parte.

Será descrito o algoritmo que permite a criação deste dígrafo; em princípio, cada vez que se emprega uma das condições de transferência (a)-(c), deve-se percorrer toda a hierarquia formada, a fim de verificar se alguma condição ainda é aplicável. O algoritmo a ser mostrado a seguir emprega uma maneira de percorrer os nós e de tentar a aplicação das condições de transferência para o qual pode ser provada a obtenção de um caminho, se ele existe.

- i) se o nó atual não é nó final e o conceito correspondente ao nó possui parte na hierarquia de entrada:
 - para cada relação de parte:
 - se o conceito que é parte ainda não está no dígrafo:
 - coloque-o;
 - se a relação de parte ainda não está no dígrafo :
 - armazene a relação e a inversa, como originais;
 - enquanto existe caminho inverso no dígrafo :
 - se existe relação não-original : elimine-a
 - se a relação já está no dígrafo : assinale-a como original;
- ii) se o nó atual do dígrafo foi acessado por ser generalização e possui partes associadas a ele no dígrafo :
 - execute (iii);
- iii) para cada nó não percorrido que corresponde a uma particularização do nó atual do dígrafo :
 - se este nó não é final, possui lista de partes recebida diferente da lista de partes do nó atual :

- para cada parte do $n\bar{o}$ atual que não seja deste $n\bar{o}$:
 - se existe caminho inverso no $d\bar{i}$ grafo :
 - se este caminho s \bar{o} cont \bar{e} m rela \bar{c} o \bar{e} s n \bar{a} o-originais :
 - assinale a exist \bar{e} ncia de conflito
 - se cont \bar{e} m alguma original :
 - assinale a rela \bar{c} o \bar{e} como n \bar{a} o-armazenada
 - se n \bar{a} o existe caminho inverso :
 - armazene a rela \bar{c} o \bar{e} e a inversa, entre a particulariza \bar{c} o \bar{e} e a parte, como n \bar{a} o-originais;
 - assinale que percorreu a particulariza \bar{c} o \bar{e} ;
 - se existe associada ao $n\bar{o}$ de particulariza \bar{c} o \bar{e} alguma sua particulariza \bar{c} o \bar{e} , no $d\bar{i}$ grafo :
 - execute (iii);
- iv) se existe, na hierarquia de particulariza \bar{c} o \bar{e} s, generaliza \bar{c} o \bar{e} do $n\bar{o}$ atual do $d\bar{i}$ grafo, tal que ainda n \bar{a} o exista no $d\bar{i}$ grafo ou existe e ainda n \bar{a} o tenha sido percorrida como generaliza \bar{c} o \bar{e} do $n\bar{o}$ atual :
 - se o $n\bar{o}$ correspondente \bar{a} generaliza \bar{c} o \bar{e} ainda n \bar{a} o est \bar{a} no $d\bar{i}$ grafo
 - armazene-o;
 - se a rela \bar{c} o \bar{e} de particulariza \bar{c} o \bar{e} ainda n \bar{a} o est \bar{a} : armazene-a;
 - assinale que $n\bar{o}$ atual do $d\bar{i}$ grafo ainda n \bar{a} o foi percorrido como particulariza \bar{c} o \bar{e} ;
 - fa \bar{c} a o $n\bar{o}$ atual \leftarrow $n\bar{o}$ da generaliza \bar{c} o \bar{e} ;
 - assinale que o $n\bar{o}$ atual foi acessado por ser generaliza \bar{c} o \bar{e} ;
 - execute (i);
- v) se o $n\bar{o}$ atual foi acessado por ser parte de outro $n\bar{o}$, o conceito possui particulariza \bar{c} o \bar{e} s, existe rela \bar{c} o \bar{e} de parte de algum $n\bar{o}$ para o atual que ainda n \bar{a} o foi transferida \bar{a} s particulariza \bar{c} o \bar{e} s do $n\bar{o}$ atual:
 - para cada rela \bar{c} o \bar{e} e cada $n\bar{o}$ de particulariza \bar{c} o \bar{e} na hierarquia:
 - se a particulariza \bar{c} o \bar{e} ou a rela \bar{c} o \bar{e} ainda n \bar{a} o est \bar{a} no $d\bar{i}$ grafo :

armazene-a;

para cada relação de parte entre o antecessor e o \bar{n} atual ainda não transferida :

se existe caminho inverso no dígrafo :

se este caminho \bar{s} contém relações não-originais :

assinale a existência de conflito

se contém relação original :

assinale que a relação não é armazenada

se não existe relação inversa :

armazene a relação e a inversa, entre o antecessor e a particularização, como não-original;

se o antecessor já está totalmente percorrido :

assinale-o como não percorrido ;

vi) se o \bar{n} atual não é o \bar{n} final, não existe particularização no dígrafo que seja \bar{n} final tal que não seja parte do \bar{n} atual e existe parte armazenada no dígrafo :

se existe parte que não esteja percorrida como parte :

se alguma delas corresponde ao \bar{n} final :

escolha-a antes;

assinale as outras como percorridas;

se alguma delas encontrou \bar{n} final :

assinale-a como percorrida;

para cada uma que vá percorrer :

escolha uma delas;

faça o \bar{n} atual \leftarrow \bar{n} correspondente a parte;

assinale que o \bar{n} foi percorrido como parte;

execute (i);

se achou o \bar{n} final : assinale o \bar{n} ;

se existe parte percorrida como parte que é \bar{n} final :

assinale as demais como percorridas;

vii) se o \bar{n} do dígrafo foi acessado por ser parte ou generalização de outro \bar{n} :

se encontrou o \bar{n} final : assinale;

retorne;

Para cada $n\bar{o}$ de conceito e cada $n\bar{o}$ de parte ligado a ele no $d\bar{i}$ grafo auxiliar, executa-se este algoritmo; deve-se juntar os $d\bar{i}$ versos $d\bar{i}$ grafos obtidos de modo a formar uma floresta.

Teorema: Se existe um caminho de um $n\bar{o}$ a outro e não existe conflito, o caminho ser \bar{a} obtido pelo algoritmo dado.

Prova:

Supondo que os $n\bar{o}$ s est \bar{a} o numerados de 1 a n e que se procura um caminho entre 1 e n , ser \bar{a} suposto tamb \bar{e} m que os $n\bar{o}$ s intermedi \bar{a} rios s \bar{a} o numerados de 2 a $n-1$, sucessivamente, e que as rela \bar{c} o \bar{e} s de parte entre um $n\bar{o}$ i e um $i+1$ tem r \bar{o} tulo $p(i)$. Se todas as rela \bar{c} o \bar{e} s $p(1), \dots, p(n-1)$ s \bar{a} o originais, elas s \bar{a} o adicionadas pelo passo (i) e percorridas pelo passo (vi), para cada $n\bar{o}$ (o passo (vi) necessariamente \bar{e} percorrido, ap \bar{o} s os demais). Se $p(i)$ \bar{e} a primeira rela \bar{c} o \bar{e} n \bar{a} o-original, ela pode ter sido adicionada pelas condi \bar{c} o \bar{e} s (b) ou (c) da Figura 2.3 ou ambas, ocorrendo as seguintes situa \bar{c} o \bar{e} s:

- 1) Se $p(i)$ pode ser adicionada por (b), deve existir um $n\bar{o}$ j como na Figura B.4 \bar{a} esquerda. O $n\bar{o}$ i poder \bar{a} ser alcan \bar{c} ado por aplica \bar{c} o \bar{e} s do passo (vi) aos $n\bar{o}$ s $1, \dots, i-1$, se n \bar{a} o existe conflito. Pelo passo (iv), acessa-se j (por ser generaliza \bar{c} o \bar{e} de i). Para o $n\bar{o}$ j , se a rela \bar{c} o \bar{e} $p(t)$ for original, o passo (2) causa a adi \bar{c} o \bar{e} de $p(i)$ e o $n\bar{o}$ $i+1$ \bar{e} alcan \bar{c} ado atrav \bar{e} s do $n\bar{o}$ j . Se a rela \bar{c} o \bar{e} $p(t)$ n \bar{a} o for original e ainda n \bar{a} o existe no $d\bar{i}$ grafo, ela tamb \bar{e} m poder \bar{a} ser obtida pela aplica \bar{c} o \bar{e} dos crit \bar{e} rios (b) ou (c), conforme os dois casos da Figura B.5. Ainda nestes ca \bar{s} os a rela \bar{c} o \bar{e} $p(s)$ pode ser original ou n \bar{a} o.

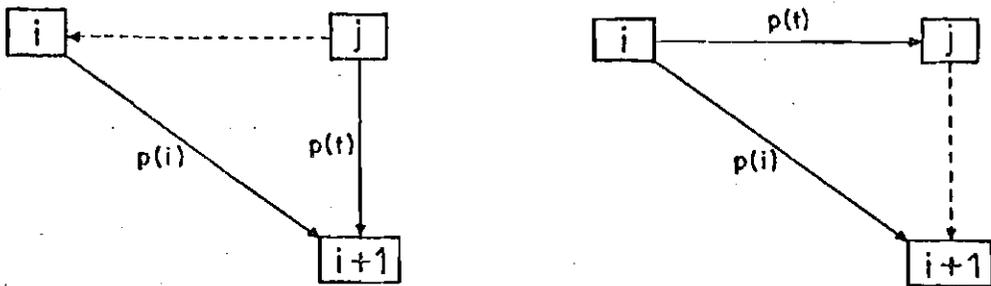


Fig. B.4 - Transferências possíveis em um caminho.

Para a Figura B.5 à esquerda, o nó k é acessado pelo passo (iv) a partir do nó j (por ser generalização de j). Se a relação $p(s)$ for original, ela será adicionada pelo passo (i); se não for original, recai-se nos casos da Figura B.5 e, como no dígrafo que contém as relações originais de parte e as de particularização é finito, chega-se a alguma relação que é original (pois existe o caminho, por hipótese). Após a adição da relação $p(s)$ ao dígrafo, a aplicação do passo (ii) causa a adição das relações $p(t)$ e $p(i)$; o nó $i+1$ é acessado como parte do nó k .

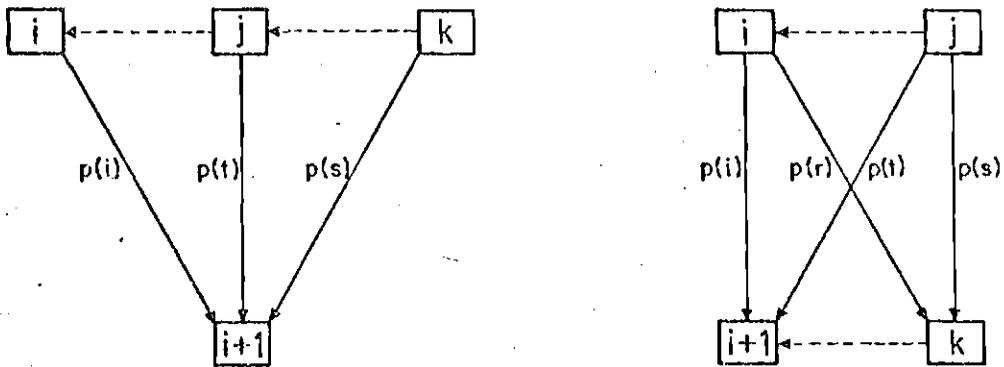


Fig. B.5 - Obtenção de relações para $p(i)$ vinda de (b).

No caso da Figura B.5 à direita, se a relação $p(s)$ for original, ela será adicionada pelo passo (i) para o nó j ; o passo (ii) causa a adição de $p(r)$ e o passo (vi) acessa o nó k . Para o nó k , o passo (v) causa a adição de $p(t)$ e o nó j é assinalado como não percorrido. Voltando ao nó j , uma nova aplicação do passo (ii) causa a adição de $p(i)$ e uma nova aplicação do passo (vi) acessa o nó $i+1$. Se a relação $p(s)$ não for original, ela é adicionada antes, recaindo-se nos casos da Figura B.5.

- 2) Se $p(i)$ pode ser adicionada pela condição (c), deve existir um nó j como na Figura B.4 à direita. A partir do nó i , se a relação $p(t)$ for original, acessa-se o nó j (pelo passo (vi)); o passo (v) para o nó j causa a adição de $p(i)$; retornando ao nó i , o nó $i+1$ é acessado como parte de i . Se a relação $p(t)$ não for original, ela pode ser obtida também pela aplicação das condições (b) ou (c), conforme a Figura B.6.

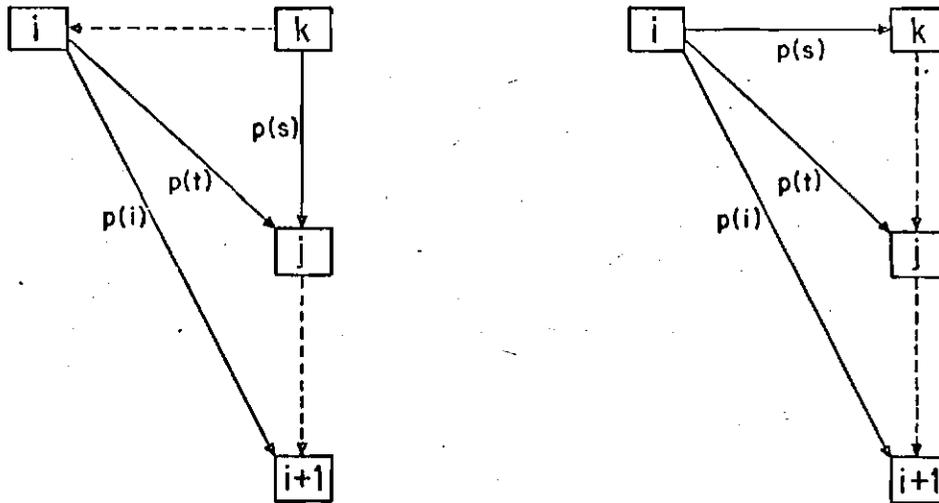


Fig. B.6 - Obtenção de relações para $p(i)$ vinda de (c).

No caso da relação $p(s)$ ser original, para a Figura B.6 à esquerda, o nó k é acessado a partir do nó i pelo passo (iv). Para o nó k , o passo (i) adiciona $p(s)$, o passo (ii) adiciona $p(t)$ e o passo (vi) acessa o nó j . Para o nó j , o passo (v) adiciona $p(i)$. Retornando aos nós k e i , o nó $i+1$ é acessado como parte de i .

Para a Figura B.6 à direita, o passo (i) adiciona $p(s)$ e o passo (vi) acessa o nó k . Para o nó k , o passo (v) adiciona $p(t)$ e o nó i é colocado como não percorrido. Retornando ao nó i , o passo (vi) acessa o nó j como parte de i . Para o nó j , o passo (iv) acessa o nó k como generalização. Retorna-se ao nó j e o passo (v) adiciona $p(i)$. Retornando ao nó i , o nó $i+1$ é acessado como parte de i .

Se a relação $p(s)$ não for original, ela pode ser obtida também por situações análogas às da Figura B.6 (Q.E.D.).

APÊNDICE C

IMPLEMENTAÇÃO DO MODELO

As estruturas de dados utilizadas na implementação já foram, na sua maioria, descritas nas Seções 2.1, 3.1 e 4.1. Muitos dos algoritmos empregados durante a aplicação do modelo também já foram descritos no Capítulo 3. Será visto como o programa é organizado, seus módulos componentes e o relacionamento destes módulos com os algoritmos.

O programa foi implementado na linguagem Extended ALGOL do computador Burroughs B-6800. Uma medida de seu tamanho é dada pelo número de linhas de codificação, com indentação, que é de aproximadamente 35000. Outra medida é dada pelo código objeto, que consome aproximadamente 1.4 Mbytes.

Os subprogramas estão subdivididos em quatro grupos:

- a) Análise de informações a serem inseridas, retiradas ou modificadas na rede: São aqueles correspondentes ao Apêndice A. Estas informações, que compõem o conhecimento declarativo que se possui sobre um universo de aplicação, devem ser todas fornecidas antes do início da utilização. Suas funções incluem a análise da expressão que descreve uma propriedade (efetua-se a análise sintática da expressão, gera-se sua estrutura interna, efetua-se a associação de variáveis a nós da propriedade, geram-se listas de referências de uso de conceitos, predicados, funções e atributos e montam-se funções de Skolem), retirada de propriedades (com o apagamento de todas as referências a ela), entrada de conceitos, atributos, predicados, funções, hierarquias (com a observação de relações inversa nas hierarquias), retirada destes tipos de elementos.

- b) Processo de análise, a partir de informações armazenadas na rede, de segmentações e imagens: Podem ser identificados diversos módulos, de acordo com a função específica: módulo de acesso a propriedades, de determinação de caminhos que envolvem relações de parte e nós de conceito, de determinação de indicadores de direção de fluxo de execução, de utilização de propriedade e de propagação de valores. Os algoritmos correspondentes já foram descritos anteriormente. Dispõe-se de um mecanismo de interrupções que permite interromper o processo em diversos pontos, escolhidos de acordo com comandos de controle; a frequência de interrupções também pode ser fixada; em cada uma, diversas ações, tais como interromper a execução, exibir resultados, armazenar resultados e alterar controles podem ser executadas.
- c) Operações sobre imagem: São aquelas que executam diversas funções elementares, gerando as estruturas de descrição de uma cena, mencionadas na Seção 4.1. Incluem módulos para geração de histogramas, entrada manual de segmentações, geração de fusão de regiões, geração de separação de regiões, cálculo de diversos atributos de regiões (área, compactação, perímetro, etc.).
- d) Manuseio de arquivos: para o armazenamento e recuperação de arquivos de regiões, segmentações, imagens e banco de hipóteses.