1. Publicação nº	2. Versão	3. Data	5. Distribuição
INPE-3945-PRE/967		*Julho, 1986*	☐ Interna ☒ Externa ☐ Restrita
4. Origem *DIN*	Programa *INTAL*		

6. Palavras chaves - selecionadas pelo(s) autor(es)
REDES ASSOCIATIVAS *ANÁLISE DE CENAS*
INTELIGÊNCIA ARTIFICIAL *REPRESENTAÇÃO DO CONHECIMENTO*

7. C.D.U.: *681.3.019*

8. Título *INPE-3945-PRE/967*

EMPREGO DE UM MODELO DE REDE ASSOCIATIVA NA ANÁLISE DE CENAS

10. Páginas: *15*

11. Última página: *13*

12. Revisada por

Edson L.F. Senne

9. Autoria *Paulo Ouvera Simoni*
Celso de Renna e Souza

13. Autorizada por

Marco Antonio Raupp
Diretor Geral

Assinatura responsável C. R. Sou

14. Resumo/Notas

Descreve-se um modelo de rede associativa para representação de conhecimento, baseado na estruturação de conceitos e no uso de fórmulas e regras de decisão para o inter-relacionamento de conceitos, relações de parte e predicados. O formato generalizado das regras de decisão é tal que expressões quantificadas podem ocorrer na premissa e na conclusão. Apresenta-se um processo de inferência para relações de parte, segundo o qual é possível haver uma compressão na representação, e discute-se o problema de conflitos inerentes a este processo. Para o emprego do conhecimento armazenado na rede, relativo a uma aplicação, define-se um mecanismo de controle com estruturas associadas que armazenam alterações, na forma de instâncias de conceitos e predicados, que devem ser propagadas, bem como problemas para os quais se deseja encontrar soluções. Mostra-se que estas estruturas de controle permitem o encadeamento progressivo e retroativo de regras de decisão. Discutem-se características de diversos modelos de análise de cenas e mostra-se como o mecanismo de controle, associado a um processo de iniciativa mista para a realização de medidas, permite uma generalização dos modelos citados.

15. Observações *Trabalho aceito para apresentação no 2º Encontro Português de Inteligência Artificial, a ser realizado nos dias 9, 10 e 11 de outubro de 1986 em Lisboa-Portugal.*

ABSTRACT

A knowledge representation associative network model is described, based in concept structuring and use of formulae and decision rules for relating concepts, plus part and predicate relations. The general decision rule format is such tmat quantified expressions may occur in the premiss and in the conclusion. An inference procedure for the part relation is presented allowing a representation compression, and the problem of conflicts brought about by this procedure is discussed. For the use of the knowledge stored in the network, relative to a given application, a control mechanism is defined, with associated structures that store alterations, in the form of concept and predicate instances, that must be propagated, as well as problems for which solutions are sought. It is shown that these control structures permit forward and backward rule chaining. Characteristics of various scene analysis models are discussed, and it is shown how the control mechanism, associated to a mixed initiative process for obtaining measures, permits a generalization of these models.

EMPREGO DE UM MODELO DE REDE ASSOCIATIVA NA ANÁLISE DE CENAS

P.O. Simoni
C.R. Souza

Ministério da Ciência e Tecnologia - MCT
Instituto de Pesquisas Espaciais - INPE
Caixa Postal 515 - 12201 - São José dos Campos - SP - Brasil

RESUMO

Descreve-se um modelo de rede associativa para representação de conhecimento, baseado na estruturação de conceitos e no uso de fórmulas e regras de decisão para o inter-relacionamento de conceitos, relações de parte e predicados. O formato generalizado das regras de decisão é tal que expressões quantificadas podem ocorrer na premissa e na conclusão. Apresenta-se um processo de inferência para relações de parte, segundo o qual é possível haver uma compressão na representação, e discute-se o problema de conflitos inerentes a este processo. Para o emprego do conhecimento armazenado na rede, relativo a uma aplicação, define-se um mecanismo de controle com estruturas associadas que armazenam alterações, na forma de instâncias de conceitos e predicados, que devem ser propagadas, bem como problemas para os quais se deseja encontrar soluções. Mostra-se que estas estruturas de controle permitem o encadeamento progressivo e retroativo de regras de decisão. Discutem-se características de diversos modelos de análise de cenas e mostra-se como o mecanismo de controle, associado a um processo de iniciativa mista para a realização de medidas, permite uma generalização dos modelos citados.

1 INTRODUÇÃO

Muitos esforços recentes de pesquisa em Inteligência Artificial foram direcionados para áreas de aplicação onde o conhecimento sobre o domínio ou sobre os processos envolvidos têm grande influência sobre o desempenho. Neste sentido, foram desenvolvidos diversos modelos de representação de conhecimento, tais como redes associativas ou semânticas, regras de decisão ou de produção, quadros ("frames") e fontes de conhecimento ("knowledge sources"), para os tipos declarativo e procedural de conhecimento.

Sistemas especialistas ("expert systems") têm sido construídos a partir desses modelos de representação, com o objetivo de captar o conhecimento de especialistas em um domínio específico e a maneira pela qual eles realizam inferências a partir de fatos expressos de maneira

incompleta ou imprecisa (empregando as chamadas *heurísticas*). Para aspectos introdutórios sobre este assunto, consultar: [BAR 81, 82], [HAY 84] e [NAU 83]. Existem muitas áreas para as quais estas técnicas foram aplicadas; algumas delas são: Medicina (auxílio ao diagnóstico), Geologia (para a elucidação de estruturas), análise de trechos conectados de voz, análise de imagens e outras.

Discutem-se as características do modelo de representação de conhecimento que foi projetado e implementado no INPE, baseado em redes associativas, e também as características do módulo de controle que permite a instanciação da rede para o caso específico da análise de cenas, e exibem-se algumas aplicações desenvolvidas; detalhes adicionais sobre o modelo e os algoritmos desenvolvidos são encontrados em [SIM 86].

2 O MODELO DE REPRESENTAÇÃO DE CONHECIMENTO

A área que serviu de inspiração para o desenvolvimento do modelo de representação e do módulo de controle a serem apresentados foi a de análise de cenas. Ao estudar as características dos problemas que devem ser enfrentados nesta área, foram observadas uma série de exigências que estes modelos deveriam satisfazer. O conhecimento estruturado a respeito do domínio era fundamental e necessário, tanto para representar classes de objetos e suas partes constituintes, como também para entidades que representassem condições contextuais. Além disto, a existência de situações onde estivesse disponível um conhecimento empírico a respeito de um problema a ser tratado sugeria a utilização, além dos mecanismos para a representação do conhecimento estruturado, daqueles fornecidos pelas regras de decisão ou de produção, com seus processos de inferência próprios. Mais ainda, era necessário integrar a estes tipos de conhecimento o tipo procedural, tendo em vista, por exemplo, as operações a serem realizadas sobre uma imagem correspondente à cena. Quanto ao controle, uma exigência do domínio de aplicação era que houvesse facilidade para efetuar operações sobre uma imagem, tanto a partir de comandos explícitos para isto durante as inferências, como a partir do modo espontâneo. Outra facilidade seria a de poder acessar regras de decisão tanto de um modo retroativo como de um modo progressivo.

Redes associativas ou semânticas foram propostas inicialmente para o armazenamento interno do significado de palavras [QUI 68], em que nós rotulados com palavras eram ligados por ramos associativos. O primeiro modelo foi modificado e melhorado de muitas maneiras, de modo a considerar estruturas explícitas para a hierarquia de particularizações, para a hierarquia de partes, para fórmulas proposicionais que relacionavam conceitos e predicados, e ainda de modo a tratar problemas de tempo e intervalos, operadores modais etc. ([FIN 79] dá uma descrição de alguns dos modelos conhecidos até aquela data). Uma característica bastante importante é a transferência de propriedades, pela qual se supõe que uma particularização de um conceito irá herdar as mesmas propriedades deste conceito.

As principais estruturas que compõem a *rede associativa* tal como foi implementada no INPE dizem respeito ao conhecimento que se possui antes de iniciar uma aplicação; são elas:

- Uma lista com os conceitos (que poderão ser empregados em hierarquias e fórmulas ou regras de decisão); seus nomes; ponteiros para lista composta de referências de nós de fórmulas ou regras, onde são utilizados predicativamente ou como relações de parte; ponteiros para as hierarquias de partes e de particularizações; e eventual índice para acessar código a ser executado em caso de ativação do conceito (por exemplo, que parcelas das hierarquias de partes ou de particularizações devem ser ativadas também).

- Uma lista com os predicados; seus nomes; ponteiros para nós onde eles são utilizados; seu tipo (calculável, não-calculável ou de relação geométrica); índice para código a ser executado (para os do tipo calculável ou relação geométrica), quando seu valor-verdade for necessário; e indicação se este cálculo envolve a realização de medidas sobre a imagem.

- Uma lista com os atributos; seus nomes; ponteiros para lista de referência; o tipo do seu valor (número, intervalo ou lista); eventual código para o cálculo de seu valor, além de uma indicação se seu valor deve ser calculado automaticamente para cada nova região criada na imagem durante o processo de análise.

- Uma lista de fórmulas e de regras de decisão, sendo a entrada destas entidades na rede feita com o auxílio de uma linguagem com sintaxe semelhante à do cálculo de predicados de primeira ordem. As regras de decisão possuem, neste modelo, uma sintaxe bastante geral, no sentido de ter um conjunto comum de quantificadores para a premissa e conclusão, o primeiro destes quantificadores devendo ser universal; as regras podem ter, também, expressões quantificadas na premissa e na conclusão, com conjuntos próprios de quantificadores.

O exemplo em (1)-(3) a seguir mostra como informações são introduzidas na rede para o caso simples de propriedades associadas a arcos. Em (1), tem-se elementos da hierarquia de partes; os conceitos intervenientes precisam estar definidos a priori. Em (2), tem-se uma fórmula em que aparece o uso predicativo de um conceito (nó ARCO), o uso de conceito como relação de parte (nós LATERAL e TOPO que expressam o fato de que as instâncias das partes devem pertencer ao mesmo arco para que a expressão possa ser satisfeita) e o emprego de um predicado do tipo calculável (nó ACIMA, devendo existir uma rotina que calcule o valor-verdade deste predicado quando seu nó na fórmula for percorrido e os referentes para as variáveis y e z forem conhecidos). A regra de decisão (3) é usada para inferir a interpretação a ser associada a uma região que corresponda à variável x quando a média das cores (em um ou mais canais de cores) dessa região estiver dentro de um certo intervalo; o grau 0.8 expressa a confiança associada à inferência.

LATERAL PARTE-DE ARCO
TOPO PARTE-DE ARCO (1)

$\forall x \; \forall y \; \forall z$ [[ARCO(x) E LATERAL(x,y) E TOPO(x,z)] ---> ACIMA(z,y)] (2)

$\forall x$ [COR(x,interv) ---> (0.8) TOPO(x)] (3)

Ainda com relação à hierarquia de partes, foram estudadas algumas condições de transferência de relações de parte, permitindo-se que estas transferências ocorram nas três situações (a)-(c) a seguir:

a) se B PARTE-DE A e C PARTE-DE B então C PARTE-DE A
b) se C PARTE-DE A e B PARTICULARIZAÇÃO-DE A então C PARTE-DE B
c) se B PARTE-DE A e C PARTICULARIZAÇÃO-DE B então C PARTE-DE A

Alguns exemplos são dados em (4)-(6) a seguir para os casos em que se efetua a transferência:

A = CORPO; B = CABEÇA; C = BOCA (4)
A = ANIMAL; B = MAMÍFERO; C = CORPO (5)
A = CORPO; B = DEDO; C = DEDO-PÉ (6)

Na implementação atual, as condições (a)-(c) não podem ser aplicadas às hierarquias diretamente; elas somente o são quando se tem uma fórmula ou regra que está sendo utilizada para realizar uma inferência e existe aí um nó que representa uma relação de parte; procura-se determinar, então, a existência de caminhos entre os nós que representam conceitos e os nós de relações de parte, e estes caminhos podem conter tanto relações de parte fornecidas explicitamente como relações que podem ser deduzidas pela aplicação das condições (a)-(c). Foi desenvolvido um algoritmo que determina um caminho constituído de relações de parte originais e inferidas, podendo ser provado que este algoritmo encontra tal caminho quando ele existe. É possível que, pela aplicação destas condições (a)-(c), se chegue a um conflito, que pode ser conseqüência de as hierarquias não terem sido bem especificadas por um usuário numa aplicação específica, conflito este caracterizado pela inferência de uma relação e de sua inversa. Este conflito pode ser evitado quando se escolhe uma das relações conflitantes como fornecida originalmente, pois isto inibe a inferência da relação inversa. Um caso de conflito é mostrado na Figura 1(a) a seguir; a Figura 1(b) mostra como uma relação B PARTE-DE C não poderia ser inferida a partir da condição (b), devido ao fornecimento explícito da relação inversa correspondente.

Outra situação em que estas inferências são interessantes ocorre quando as relações de parte e de particularização são inseridas na rede. O único teste atualmente realizado é o referente à existência de ciclos ou de relações inversas, somente dentre as relações originais; numa extensão futura, estas condições (a)-(c) de inferência também serão consideradas.

Estas características do modelo de representação referem-se ao conhecimento que é fornecido antes de qualquer aplicação. Durante a utilização do sistema para realizar alguma tarefa relacionada com a análise de cenas, monta-se uma outra estrutura denominada *banco de hipóteses*, o qual contém as parcelas já instanciadas da rede. Este banco é composto dos conceitos já instanciados, com os referentes associados a cada instância (os referentes podem ser, por exemplo, regiões de uma imagem sobre as quais são feitas conjecturas sobre as suas possíveis interpretações). Ele é composto também pelos predicados já instanciados, tanto os calculáveis, para os quais se obteve o valor-verdade, como para os não-calculáveis que tiveram seu valor atribuído durante o emprego de uma fórmula ou regra de decisão numa inferência. As fórmulas e regras já empregadas também são armazenadas, juntamente com os referentes das va-

riáveis (que foram considerados fixados antes do emprego destas fórmulas ou regras) e os elementos mapeados a priori a cada nó (veja Seção 3).

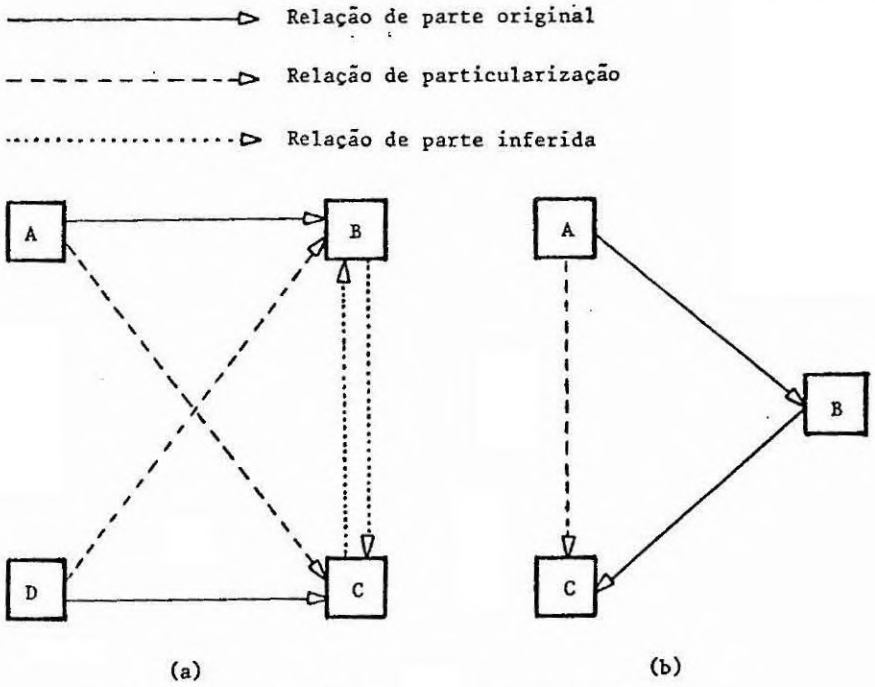


Fig. 1 - Inferência de relações de parte.
(a) - conflito
(b) - inibição

As instâncias do banco de hipóteses, em particular as hipóteses sobre a identificação de regiões de uma imagem, podem ser geradas, por exemplo, a partir de alguma evidência proveniente de uma medida realizada sobre a imagem (como no exemplo (3) anterior); elas podem ser geradas, também, de modo tentativo, de modo tentativo, seja por iniciativa do próprio programa, seja através de imposição externa ao programa. É possível que, uma vez atribuída uma interpretação, o mecanismo de controle a propague ou gere perguntas sobre sua validade, as quais devem ser respondidas pela ativação de fórmulas ou regras de decisão, que, por sua vez, podem solicitar a realização de novas medidas sobre a imagem. Podem existir diversas interpretações associadas a uma região e, também, evidências contraditórias a serem consideradas para cada uma delas.

3 CARACTERÍSTICAS DO MÓDULO DE CONTROLE

Todas as ações correspondentes à realização de inferências, ativação das hierarquias associadas aos conceitos, distribuição do processamento entre as tarefas voltadas ao processamento simbólico e as voltadas ao processamento sobre imagens, e ativação de ações de inicialização são realizadas pelo módulo de controle. Uma observação sobre diversos modelos para a análise de cenas e sobre como este módulo generaliza algumas das características destes modelos será feita mais adiante.

A maneira como os dois tipos de processadores se comunicam é mostrada na Figura 2, e as listas de medidas, de alterações e de problemas têm a função de servir como estruturas de controle; as demais estruturas servem como entrada de dados para o sistema ou para o armazenamento de resultados intermediários. A separação entre os dois processadores é realizada, na prática, através de um esquema de prioridades e de divisão da carga de trabalho do único processador disponível fisicamente.

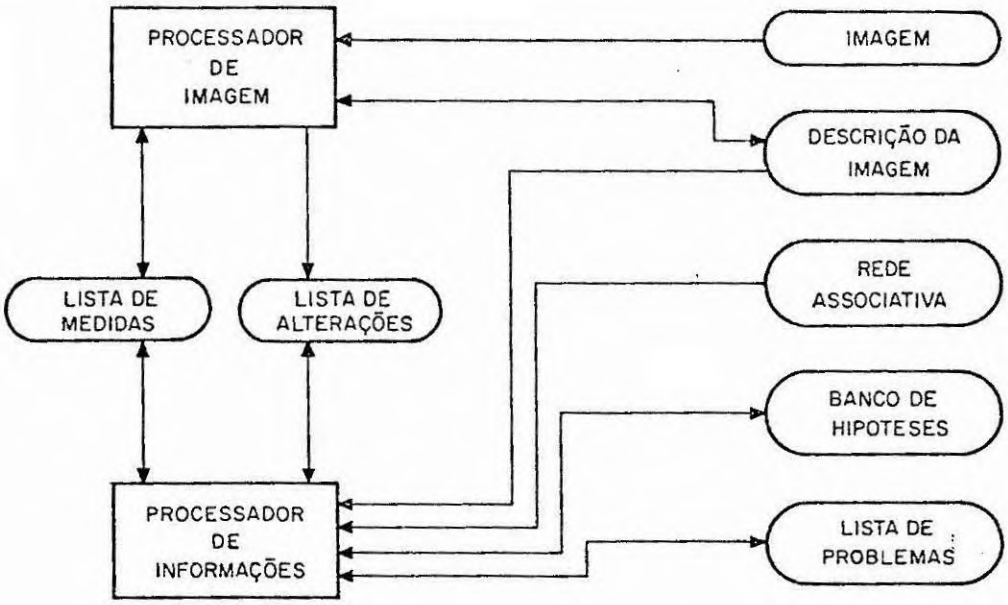


Fig. 2 - Fluxo de dados do sistema.

A imagem, na versão atual, pode ter até 256 níveis de cinza e resolução de 64x64; uma nova versão deverá ter a resolução aumentada para 1024x1024. A descrição da imagem consiste num conjunto de estruturas que são formadas a partir do processamento da imagem. Esta descrição contém as regiões já formadas, o histórico do processo de formação delas (se foram obtidas por separação - "split", por fusão - "merge" etc), as máscaras associadas às regiões, as interpretações associadas a cada uma delas etc. Este conjunto de estruturas, de descrição da imagem, é mantido estanque em relação às demais da rede, de modo a possibilitar extensões futuras a outras classes de aplicações (como no reconhecimento de fala, seqüências de imagens etc), com as estruturas a serem alteradas localizadas neste conjunto; a subestrutura que contém as interpretações, com informações comuns à rede e à imagem, deveria, em tais circunstâncias, ser também objeto de reexame.

A lista de alterações armazena ponteiros para os elementos que foram introduzidos ou que tiveram seu valor alterado no banco de hipóteses. Estas alterações devem ser propagadas, sendo a escolha das próximas a serem consideradas feita de acordo com a ordem de entrada (a primeira, a última ou todas elas), ou de acordo com algum critério estabelecido de maneira empírica. Quando uma delas é escolhida para ser propagada, o elemento correspondente à alteração no banco de hipóteses é mapeado aos nós de fórmulas e regras de decisão da rede, de acordo com a igualdade

dos conceitos ou predicados envolvidos e de acordo com a igualdade dos referentes determinados a priori para os nós na rede com os referentes do elemento escolhido. O elemento que está sendo propagado pode ser mapeado como um dado conhecido, a partir do qual se quer inferir novos fatos, ou como uma pergunta para a qual a fórmula ou regra pode fornecer uma resposta. Após a utilização das fórmulas ou regras mapeadas, as inferências eventualmente realizadas podem resultar em novos elementos colocados nesta lista.

A lista de problemas contém problemas expressos na forma de elementos para os quais se procuram referentes que o satisfaçam. Por exemplo, se A é um conceito, um problema expresso como "A()?" indica que se procura um referente para A e este referente deve ser encontrado através da utilização de fórmulas ou regras de decisão que possam inferí-lo. Assim, se existir uma regra que possua este conceito como um dos nós da conclusão, ela é candidata a gerar uma resposta e ser empregada de uma forma retroativa. Os problemas podem ser encadeados entre si, no caso de um deles precisar da solução de outros para a sua solução. Este encadeamento surge quando nós de fórmulas ou de regras são mapeados ao problema e, no processo de inferência, existem outros nós que também não possuem ainda elementos correspondentes no banco de hipóteses; a fórmula ou regra tem sua utilização suspensa até a obtenção da solução dos problemas associados. O encadeamento pode surgir também devido à geração de subproblemas provenientes de conceitos relacionados ao inicial como parte ou particularização.

A lista de medidas destina-se a armazenar pedidos ainda não realizados de operações sobre a imagem. Estes pedidos podem ser provenientes tanto de uma ação espontânea por parte do processamento de imagens, como da necessidade de saber o valor de um atributo ou predicado durante a realização de uma inferência pela utilização de fórmulas ou regras. Este último caso ocorre se houver uma inibição da realização de medidas durante o processo de inferência; ocorre, então, uma suspensão da utilização da fórmula ou regra e o pedido da medida é empilhado.

O controle também pode ser influenciado por um conjunto de comandos de controle, os quais assinalam opções pré-programadas de ações disponíveis. Assim, por exemplo, existem comandos para: fixar as prioridades de atuação dos dois tipos de processadores, escolher a ordem de propagação de alterações, armazenar valores encontrados para funções de Skolem, ativar as hierarquias de um conceito que foi instanciado, definir os locais onde ocorrem interrupções, controlar a geração de subproblemas e para as ações pré-programadas de inicialização a serem invocadas quando não houver ação possível etc.

O fluxo de controle do sistema está esquematizado na Figura 3; por simplicidade de exposição, diversos laços de realimentação que ocorrem dentro de cada subconjunto de ações foram eliminados.

Se existem alterações a propagar ou problemas a resolver
. elimine as alterações já propagadas;
. verifique se alguma ação foi especificada a priori;
. procure uma fórmula ou regra de decisão com utilização suspensa;
. escolha o problema a resolver;
. escolha a alteração devida a um valor desconhecido ou tentativo;
. efetue o acesso a fórmulas ou regras, para inferência.

Escolha o mapeamento a ser considerado;
. elimine as alterações que geraram acesso, se não houver mapeamentos;
. determine os indicadores de direção para o mapeamento escolhido;
. verifique a utilização anterior do mesmo mapeamento.

Empregue a fórmula ou regra escolhida para inferência;
. efetue percorrimento de nós de fórmula ou regra;
. execute a análise de pós-utilização;
. verifique as condições para a reutilização de fórmulas ou regras.

Determine condições para a realização de medidas sobre a imagem;
. escolha o pedido empilhado de medida;
. determine a medida a ser realizada de maneira espontânea;
. execute as ações de inicialização.

Fig. 3 - Fluxo de controle do sistema.

No início de cada ciclo, verifica-se se existe alguma ação a ser considerada para o exemplo que está sendo rodado, especificada através de uma rotina compilada separadamente e escrita para cada exemplo; esta rotina serve, por exemplo, para selecionar um tipo específico de problema a resolver, uma alteração a propagar etc.

Se existe alguma alteração a propagar ou problema a resolver que tenha sido escolhido, monta-se uma estrutura que irá conter os elementos a serem mapeados a nós de fórmulas ou regras, os referentes para os elementos que puderem ser determinados (elementos fictícios podem ser acrescentados em correspondência a nós que representam relações de parte e a nós para os quais não se pode determinar o referente de alguma variável - por exemplo, nó com variável existencial) e os diversos mapeamentos elemento-mó. O algoritmo que efetua estes mapeamentos (chamado algoritmo de acesso a fórmulas ou regras) efetua diversas ações como: determinar referentes de variáveis ou estabelecer que estes referentes deverão ser procurados durante o processo de inferência através de uma busca sobre os elementos já definidos no banco de hipóteses; determinar caminhos formados por nós de conceito e de relações de parte (originais ou inferidas através das condições (a)-(c) mostradas na seção anterior); gerar subproblemas correspondentes a um problema que originou a chamada ao algoritmo.

O emprego de uma fórmula ou regra para a qual se tenha gerado um conjunto de mapeamentos a seus nós envolve a escolha do mapeamento, a atribuição de sinalizadores de indicação de indicação de como o valor associado a um nó poderá ser usado (utilizado ou atribuído), a verificação se o mesmo mapeamento já foi empregado anteriormente e um percorrimento dos nós da fórmula ou regra para a resposta a uma pergunta ou propagação de dados. Para a escolha do mapeamento, podem ser acrescentadas heu

rísticas próprias a um determinado problema. A atribuição de indicadores visa estabelecer, por exemplo: que o nó correspondente a um elemento sobre o qual se fez uma pergunta deve ter seu valor atribuído, que um nó correspondente a um predicado do tipo calculável deve ter seu valor propagado, que a premissa de uma regra deve ter seu valor utilizado e a conclusão deve ter seu valor atribuído; outros casos consideram o fato dos elementos serem originais ou adicionados durante o algoritmo de acesso.

Durante o percorrimento para a inferência, monta-se uma lista de valores provisórios para os valores criados ou alterados de hipóteses, de relações de parte, de predicados e de atributos. Após o término do percorrimento, verifica-se se ele foi suspenso devido à alguma medida pedida ou subproblema gerado; verifica-se, também, se houve algum conflito entre valores antigos e recentemente concluídos para elementos do banco de hipóteses. Para a combinação de valores novos com antigos, podem ser chamadas rotinas específicas, de acordo com o modelo de combinação desenvolvido para cada caso. Os valores assim obtidos podem ser então inseridos no banco de hipóteses e, para o caso de conceitos instanciados, ativadas suas hierarquias de partes e de particularizações. Se os novos valores são tais que uma antiga inferência realizada não seja mais válida devido às alterações introduzidas, é possível que seja necessário um processo de reutilização das fórmulas ou regras.

Ações de inicialização podem ser executadas sempre que o processador de imagens estiver habilitado e não existir nenhuma medida a ser realizada. Estas ações podem resultar, por exemplo, na atribuição tentativa de uma interpretação a uma região que não possua nenhuma interpretação a ela associada.

O emprego das listas de problemas e de alterações permite os encadeamentos retroativos e progressivos de fórmulas ou de regras de decisão. O exemplo (7)-(11) a seguir mostra como se encadeiam as regras destas formas, onde: A, B, C, D, F, G, H, HG, I e J representam conceitos; são conhecidos a priori os valores-verdade das instâncias de (12), colocados como dados; e tem-se um problema estabelecido como "A()?".

```
∀x [[ B(x)  E C(x)] --> (0.9) A(x) ]                                  (7)
∀x [[ D(x)  E F(x)] --> (0.8) C(x) ]                                  (8)
∀x [[ C(x)  E G(x)] --> (0.7) H(x) ]                                  (9)
∀x [[ HG(x)  E I(x)] --> (0.6) J(x) ]                                 (10)
∀x [ B(x)           --> (0.5) I(x) ]                                  (11)
B(1), B(2), D(1), D(2), F(1), F(2), G(1), G(2): valor-verdade V        (12)
```

As ações executadas são, esquematicamente, dadas a seguir:

A()? : causa o acesso a (7) que gera dois mapeamentos, um correspondente a B(1) e outro a B(2); durante a utilização desta regra, geram-se os problema "C(1)?" e "C(2)?" para cada um dos mapeamentos escolhidos, os quais ficam suspensos (encadeamento retroativo de (7)).

B(2)? : causa o acesso a (7), que é rejeitada, pois a pergunta correspondente a um nó da premissa da regra;
causa o acesso a (8), a partir da qual o valor-verdade de C(2) é calculado;
causa o acesso a (9), que é rejeitada.

C(1): as mesmas regras acessadas para o caso anterior geram C(1). Após a resolução dos subproblemas acima, (7) pode voltar a ser percorrida, o que gera o valor-verdade V para A(1) e A(2), que serão propagados a seguir.

A(1) : causa o acesso a (7), que é rejeitada, pois corresponde a um nó da conclusão.

C(1) : causa o acesso a (7) e a (8) (que são rejeitadas) e a (9) que gera o valor-verdade V para H(1) (encadeamento progressivo de (9)); HG(1) também é gerado após a utilização de (9) através da hierarquia de particularizações.

HG(1) : causa o acesso a (10) que gera o problema "I(1)?", pois não é conhecida nenhuma instância para I e supõe-se que seja permitida a geração de problemas nestas circunstâncias, através de uma opção de controle.

I(1) ? : causa o acesso a (10) (que é rejeitada) e a (11) que permite calcular o valor-verdade de I(1) como V.
Após a resolução do problema, (10) volta a ser percorrida, gerando J(1).

J(1) : causa o acesso a (10) que é rejeitada.

I(1) : causa o acesso a (10) e a (11) que são rejeitadas.

H(1) : causa o acesso a (9) que é rejeitada.

A(2) : são executadas as mesmas ações que para a propagação de A(1).

Deve-se evitar que uma alteração que seja proveniente da resolução de um problema cause o acesso à mesma regra onde o problema foi gerado, a fim de evitar "loops" infinitos.

4 APLICAÇÕES DESENVOLVIDAS

Com a finalidade de testar o modelo de representação e as características do módulo de controle, foram desenvolvidas, inicialmente, duas aplicações. As operações disponíveis sobre a imagem incluem: separação de regiões através da análise de picos em histogramas, fusão de regiões, geração de envoltórias de menor resolução com uma estrutura do tipo pirâmide, diversas relações geométricas, e atributos de cor. Existem disponíveis diversos cabeçalhos de procedimentos que podem ser empregados para uma compilação separada de novas operações que possam ser desejadas; existe também uma maneira padronizada de definir argumentos para estes procedimentos.

A primeira aplicação destina-se a reproduzir o comportamento de um algoritmo já conhecido para a segmentação controlada pela interpretação [TEN 76]. Existem dois aspectos importantes a serem considerados neste algoritmo: em primeiro lugar, a realização da fusão em si e a escolha do par de regiões candidatas à fusão; em segundo lugar, a propagação de insconsistências, no sentido de empregar um conjunto de restrições para rejeitar algumas interpretações associadas a uma região, que, por sua vez, também devem ser propagadas (no início do processo, todas as regiões têm atribuídas a elas todas as interpretações possíveis). As restrições são do tipo (PICTURE INSIDE WALL). Este algoritmo apresenta

aspectos importantes para os testes pretendidos: é necessário considerar a existência de "backtracking", todas as ordens possíveis de pares de regiões e todas as ordens possíveis de restrições. O algoritmo teve seu comportamento simulado, especificando como regras de decisão as condições pelas quais se processa a rejeição de interpretações (ou seja, através de dados). Algumas das 19 regras geradas são mostradas em (13)-(15) a seguir, onde F, K, P são algumas das interpretações que podem ser atribuídas a regiões (mantém-se as iniciais do trabalho citado em inglês devido à ambiguidade das iniciais na tradução); as regras permitem a negação de uma condição que é válida na premissa; e os predicados DENTRO e ACIMA são do tipo calculável. Pode ser observado também o emprego de uma expressão quantificada na premissa das regras (14) e (15); o predicado não-calculável ATIVO foi acrescentado com a única finalidade de permitir que se encontre um referente para a variável y, pois, sem ele, ela só apareceria como argumento de predicados calculáveis e, nesta situação em particular, não haveria como determinar seu referente.

$\forall$ x [[P(x) E NAO PEQUENO(x)] --> (1.0) NAO K(x)] (13)

$\forall$ x [$\exists$y [P(x) E NAO W(y) E DENTRO(x,y)] --> (1.0) NAO P(x)] (14)

$\forall$ x [$\exists$y [[F(y) OU K(x) OU P(x)] E ATIVO(y) E ACIMA(x,y)]
--> (1.0) [NAO F(x) E NAO K(x) E NAO P(x)]] (15)

Foram realizados testes com o conjunto de regras de decisão citado anteriormente e o sistema simulou corretamente o processo de propagação de inconsistências. Observou-se, para esta aplicação, que o algoritmo que efetua o acesso a fórmulas e regras foi o responsável pela maior parcela do tempo de processamento, sendo realizadas também um grande número de tentativas de reutilizar regras de decisão anteriormente empregadas, sempre que uma interpretação era rejeitada.

A segunda aplicação mostra como considerar condições contextuais para a realização de medidas numa imagem. Um processo de separação por picos em histogramas considera, na situação comum, a geração de histogramas de níveis de cinza relativos a uma imagem ou região, a procura de picos nos histogramas, a escolha de um deles (em princípio, o pico mais acentuado), a separação da região ou imagem em sub-regiões correspondentes aos níveis de cinza dentro e fora do pico (que podem gerar mais de uma sub-região, para cada caso). Este processo foi utilizado considerando a sua chamada sempre que uma região não possuir nenhuma interpretação associada a ela ou, então, mais de uma interpretação, e sempre que a variância dos níveis de cinza em uma região estiver acima de um certo limiar; outra situação ainda é aquela em que este tipo de operação pode ser invocado por uma ação espontânea do processador de imagens. As regras de decisão (16)-(18) a seguir são algumas das utilizadas para efetuar este tipo de controle:

$\forall$ x [[VEGET1(x) E NAO CANDIDATOSEP(x) E NAO EFETUOUSEP(x) E
MAIORVAR(x,limiar)] --> (1.0) CANDIDATOSEP(x)] (16)

$\forall$ x [[ATIVO(x) E NAO CANDIDATOSEP(x) E NAO EFETUOUSEP(x) E
[NAO MAIORINTASSOC(x,0) OU MAIORINTVERD(x,1)]]
--> (1.0) CANDIDATOSEP(x)] (17)

$\forall$ x [[CANDIDATOSEP(x) E NAO EFETUOUSEP(x) E SEPARAREGIAO(x)]
--> (1.0) NAO CANDIDATOSEP(x)] (18)

Existem vários modelos para análise de cenas e para ciclos de percepção que foram propostos. Uma breve comparação pode ser feita com relação a alguns deles. A segmentação controlada pela interpretação [TEN 76] é limitada quanto ao escopo de aplicação, embora o algoritmo de propagação de inconsistências, assim como o de rotulação de cenas de blocos [WAL 75], tenha servido para uma extensão bastante importante, ou seja, o algoritmo de relaxação; foi mostrado que o mecanismo de controle do sistema implementado é capaz de reproduzir este comportamento. A abordagem para considerar o problema da fusão por regras do tipo antecedente e conseqüente [OHT 80] possui regras com utilização pré-fixada; embora não se tenha reproduzido completamente o comportamento deste sistema através do modelo implementado, diversas de suas operações podem ser simuladas através de regras de decisões conforme foram definidas. A análise por "schemata" (HAV 78, 83] emprega um ciclo recursivo de percepção, numa representação do tipo procedural, enquanto o modelo que está sendo apresentado, por sua vez, possui um ciclo já pré-programado (embora alterações possam ser incluídas nele) e a representação é basicamente do tipo declarativa, e possibilita, além das hierarquias e rotinas que podem estar associadas a sua ativação, o emprego de fórmulas e regras de decisão.

5 CONCLUSÕES

Descrevem-se diversas características de um modelo de representação de conhecimento por redes associativas, desenvolvido no INPE, para aplicações em análise de cenas. Destacam-se a sintaxe estendida de regras de decisão; sua representação de uma maneira uniforme, juntamente com fórmulas; a possibilidade de representar conhecimento do tipo procedural, tanto através de códigos associados a predicados e atributos como através de heurísticas e códigos associados à ativação de hierarquias. O módulo de controle é bastante flexível, de modo a permitir o emprego de opções pré-programadas e heurísticas em diversos pontos. O encadeamento progressivo e retroativo de regras de decisão também foi incorporado ao modelo, de modo que a forma com que as regras são usadas não seja fixada a priori e que uma mesma regra possa ser empregada em contextos diferentes segundo os dois tipos de encadeamento; além disto, é possível entremear os dois tipos de encadeamento.

O sistema doi desenvolvido na linguagem EXTENDED ALGOL para o computador Burroughs B-6800. O código fonte ocupa aproximadamente 35000 linhas de comandos com identação, e o código objeto ocupa da ordem de 1.4 Mbytes.

Foram estudadas as condições pelas quais este modelo poderá ser adaptado a outros domínios, tais como a visão robótica, sensoriamento remoto e análise de imagens.

6 REFERÊNCIAS BIBLIOGRÁFICAS

[BAR 81] Barr, A.; Feigenbaum, E.A., ed. *The handbook of Artificial Intelligence*. Stanford, CA, Heuristech Press, 1981, v.1.

[BAR 82] Barr, A.; Feigenbaum, E.A., ed. *The handbook of Artificial Intelligence*. Stanford, CA, Heuristech Press, 1982, v.2.

[FIN 79] Findler, N.V., ed. *Associative networks: representation and use of knowledge by computers.* New York, Academic, 1979.

[HAV 78] 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.* p. 244-253.

[HAV 83] Havens, W.S.; Mackworth, A.K. Representing knowledge of the visual world. *Computer*, 16(10):90-98, 1983.

[HAY 84] Hayes-Roth, F. Knowledge-based expert systems. *Computer*, 17(10):263-273, 1984.

[NAU 83] Nau, D.S. Expert computer systems. *Computer*, 16(2):63-85, 1983.

[OHT 80] Ohta, Y. *A region-oriented image-analysis system by computer.* Kyoto, Dr. Eng. Thesis, Dept. of Info. Sci., Kyoto Univ., 1980.

[QUI 68] Quillian, M.R. Semantic memory. In: Minsky, M., ed. *Semantic information processing.* Cambridge, MA, MIT Press, 1968, p. 227-270.

[SIM 86] Simoni, P.O. *Uma abordagem para redes associativas com aplicação na análise de cenas.* São José dos Campos, Tese de doutorado, INPE, Mar 1986. (no prelo).

[TEN 76] Tenenbaum, J.M.; Barrow, H.G. Experiments in interpretation-guided segmentation. Menlo Park, CA, Stanford Research Institute, 1976. (Tech. Note 123).

[WAL 75] Waltz, D.L. Understanding line drawings of scenes with shadows. In: Winston, P.H., ed. *The psychology of computer vision.* New York, McGraw, 1975, p. 19-91.