

Lista de exercícios 4 – Sistemas de Produção
Exercícios práticos e conceituais

Disponibilizada no 11-nov-07

OBSERVAÇÃO: Para fazer essa lista de exercícios é preciso estudar o material disponibilizado na Aquarela.

1. PERGUNTA-SE:
 - 1.1. Quais são os principais componentes de um sistema especialista?
 - 1.2. Especifique, para os componentes que você indicou no item anterior, quais são os dependentes e os independentes do domínio de aplicação, justificando sua resposta.
 - 1.3. Diga, em linhas gerais, quais os componentes que você indicou acima diretamente relacionados a um sistema de produção que usa a estratégia de busca em largura primeiro.
 - 1.4. Em que componente se encontram as regras de produção, em seu sistema especialista?
2. PERGUNTA-SE:
 - 2.1. O que é um shell de um sistema especialista?
 - 2.2. Qual é a principal característica de um sistema especialista com encadeamento regressivo (*backward chaining*)?
 - 2.3. E de um SE com encadeamento progressivo (*forward chaining*)?
3. Fazendo analogia com a memória de curto prazo (MCP), mostre como um sistema de produção raciocina para resolver um problema utilizando um método dirigido pelos objetivos.
4. Considere o seguinte algoritmo básico de resolução de um sistema de produção¹:

Procedure PRODUCTION

DADO ← dado inicial

Até que DADO satisfaça condição de término, faça:

{

 busque conjunto C, de regras aplicáveis a DADO

 selecione alguma regra R, do conjunto C

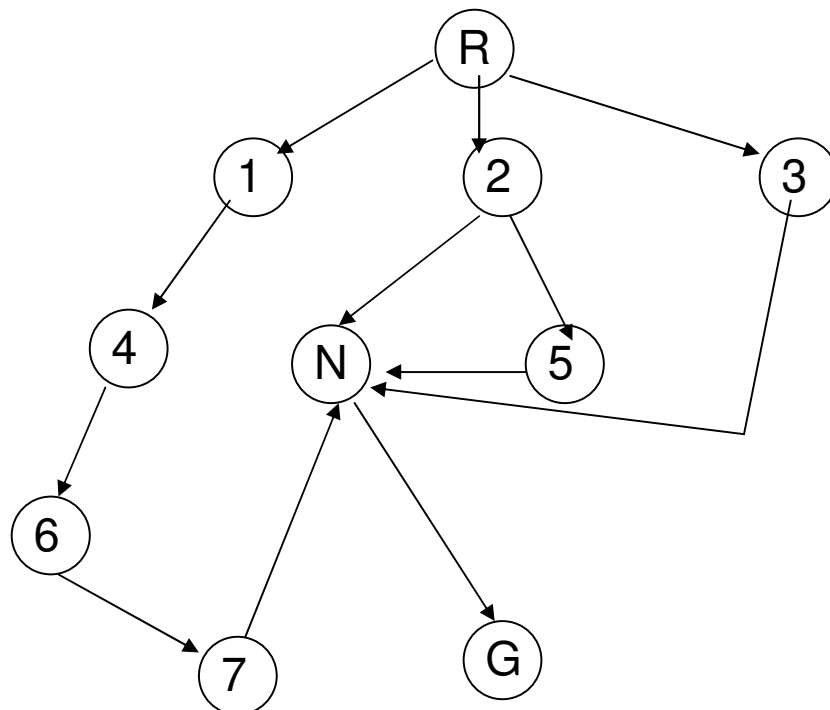
 DADO ← resultado da aplicação de R a DADO

}

Responda:

- a) Qual é o conjunto de conflito desse sistema de produção, segundo a caracterização de Luger & Stubblefield?
 - b) Que informação você teria que incluir nesse algoritmo para que o processo de resolução seja não-determinístico? Indique onde essa informação deve ser incluída, detalhando o algoritmo, ainda em linguagem de alto nível.
5. Dado o seguinte grafo de solução, em que a situação inicial é dada pelo nó R (raiz) e que a situação objetivo é dada pelo nó G, diga qual o conjunto de conflito para expandir o nó N quando:
- 5.1. A direção data-driven for usada (processo dedutivo).
 - 5.2. A direção goal-driven for usada (processo indutivo).

¹ Nilsson, p. 21.



6. Considere que sua lista OPEN, de trajetórias potenciais, seja expressa pela sequência de números das regras de sua base de conhecimento e que, caso uma mesma regra seja escolhida, a cada característica verdadeira de seu corpo, um valor que indica a relevância da característica seja acumulado, de modo que, se a regra for aplicada integralmente, o custo da trajetória expressará a relevância total das características correspondentes. Inicialmente, as possíveis trajetórias podem ser expressas em OPEN pelo conjunto de regras iniciais aplicáveis ao problema delineado, da seguinte forma²:

$$OPEN = \{ \text{regra}(1,0), \text{regra}(2,0), \text{regra}(3,0), \dots, \text{regra}(1000,0) \}$$

Suponha que, quanto mais relevante for o conjunto de características desejadas por um cliente, maior será sua pré-disposição para comprar o cão.

RESPONDA O MAIS CONCISAMENTE POSSÍVEL:

- 6.1. Indique *COMO* os índices de relevância das características podem ser indicados em cada regra. Use, para isso, uma regra genérica como

```

identify(Cão,NumeroDaRegra) se
    característica 1 for verdadeira e
    característica 2 for verdadeira e
    ..... e
    característica N for verdadeira
  
```

para N qualquer (conjunto de possíveis estados de seu espaço de solução).

- 6.2. Indique *COMO* esses índices serão acumulados para que a lista OPEN guarde, além das possíveis trajetórias a expandir, também sua relevância acumulada.

² Não se conhecem as características do cão e, portanto, a relevância de qualquer regra ainda é inexpressiva. A representação ilustrada tem a notação *regra(Nro_da_regra,Relevância_total_das_caract_indicadas_nessa_regra)*.

- 6.3. Esses índices serão usados para classificar a lista OPEN a cada iteração e, logo, indicar qual a trajetória parcial mais promissora para ser expandida, na busca pela solução. OPEN será classificada ascendente ou decendentemente? Justifique.
- 6.4. Qual a trajetória a ser escolhida para prosseguir raciocinando?
- 6.5. É possível que uma trajetória em expansão durante o raciocínio seja abandonada e outra seja escolhida para o próximo instante de raciocínio? Justifique com uma ilustração criada por você.
- 6.6. Qual seria a condição para que uma trajetória abandonada temporariamente fosse retomada em instante posterior da busca da solução?

NOTA: Não é necessário ilustrar suas respostas em Prolog. Se preferir, use uma linguagem algorítmica, o mais natural possível (NÃO USE CÓDIGO EM NENHUMA LINGUAGEM DE PROGRAMAÇÃO).

7. Considere o código Prolog de um sistema especialista (SE) sobre pássaros chamado 'NATIVO', cuja base de conhecimento(KB) parcial é ilustrada abaixo (arquivo 'passaros.kb').

Para responder às perguntas abaixo, considere o seguinte:

- Se achar mais fácil, você pode inserir cada resposta que remete ao código NATIVO diretamente como comentários no próprio código, iniciando-o numa nova linha, com o símbolo de comentário em Prolog "%".
- Se você precisar delimitar início e fim de um segmento de código, indique isso na própria linha de comentário. Por exemplo, escrevendo "% início da shell" ou "% fim da shell", como resposta às perguntas abaixo, nas linhas correspondentes ao código de início e fim da shell.

CONHECIMENTO FORNECIDO

Base de conhecimento: regras se-então, para dois tipos de albatroz e cisne:

SE família eh albatroz e cor eh branca ENTÃO pássaro eh albatroz_terreno SE família eh albatroz e cor eh preta ENTÃO pássaro eh albatroz_PesPretos	SE família eh cisne e voz eh baixa, musical e chorosa ENTÃO pássaro eh cisne_assobiador SE família eh cisne e voz é alta e grave ENTÃO pássaro eh cisne_tenor
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> % Código Prolog do SE NATIVO - exemplo simples go :- cumprimento, read(X), do(X). cumprimento:- write('Esse é o sistema Prolog NATIVO.'), nl, write('Entre "load", "consult", or "quit" quando prompt aparecer.'), nl. do(load) :- load_kb, !. do(consult) :- solve, !. do(quit). do(X) :- write(X), write('não é um comando permitido.'), nl. solve :- top_goal(X), write('A resposta eh '), write(X), nl. solve :- write('Nenhuma resposta encontrada.'), nl. </pre>	<pre> load_kb :- write('De o nome do arquivo da KB: '), read(F), consult(F). top_goal(X) :- passaro(X). passaro(ganso_canada):- familia(ganso), estação(inverno), pais(eua), cabeça(preta), bico(branco). passaro(ganso_canada):- familia(ganso), estação(verao), pais(canada), cabeça(preta), bico(branco). % Outras regras para definição de outros passaros certamente seguirao estas, mas segundo formato similar. </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

PEDE-SE:

- 7.1. Esboce o segmento de código em Prolog que corresponde à KB 'passaros.kb' em Prolog, conforme as regras se-então acima.
- 7.2. Indique também qual é o programa principal do NATIVO.
- 7.3. Delimite o segmento que corresponde à shell desse sistema (indique o programa inicial (*) e o programa final (**) dessa shell, que estão em seqüência no código NATIVO)³.
- 7.4. Indique, no próprio código (***), o programa que faz acesso à base de conhecimento do sistema.
- 7.5. Diga qual é o motor de inferência do SE NATIVO.
- 7.6. Qual é o objetivo "raiz" de uma árvore de solução do NATIVO, sabendo que esse sistema identifica qual é o pássaro cujas características são dadas por um usuário qualquer?
- 7.7. Uma vez esboçada sua KB e acrescentada ao código do NATIVO abaixo, qual é o programa PROLOG que inicia a consulta ao NATIVO?
- 7.8. Considerando que esse SE é um SBC (Sistema Baseado em Conhecimento), indique qual é seu programa mais importante.
- 7.9. Se você quiser implementar um programa *askable* para que o NATIVO obtenha as características desejadas para algum pássaro diretamente do usuário, onde você deve inserir o goal correspondente no NATIVO? Dê um exemplo deste caso.
- 7.10. Seu programa *askable* é parte da shell ou do sistema NATIVO, propriamente dito? Justifique.
- 7.11. Seu programa *askable* deve prever mesmo respostas "não" do usuário, para cada uma das perguntas, retrocedendo e buscando outras características de pássaros que possam ser aplicáveis? Justifique.

³ Marcas com asteriscos a serem usadas no próprio código, para cada caso.

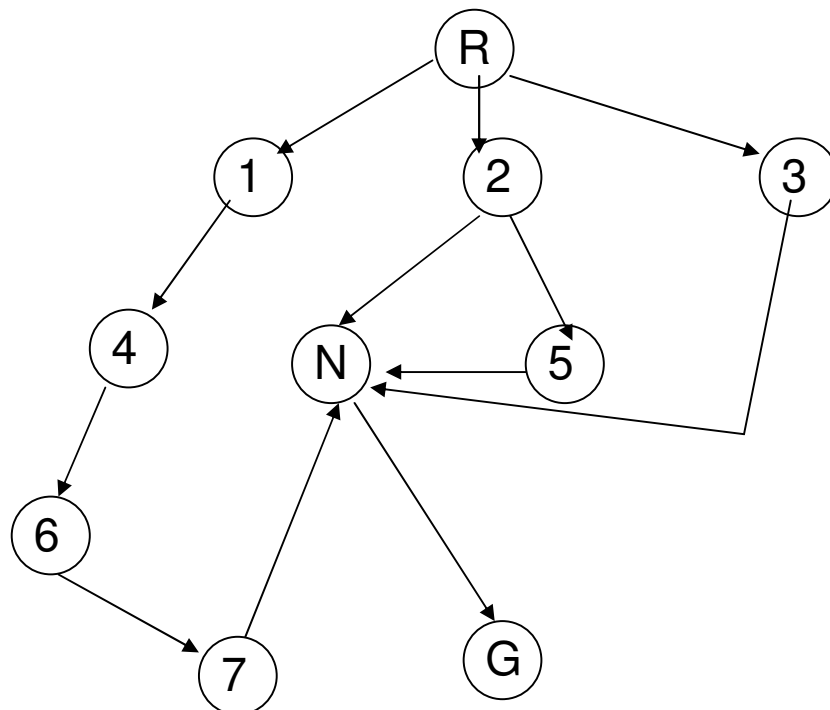
8. Tanto uma Rede Neural quanto um Sistema Baseado em Conhecimento são modelos de sistemas computacionais cujo objetivo é simular raciocínio. Um Sistema de Produção, no entanto, difere muito de uma Rede Neural. Aponte a diferença fundamental entre eles⁴.
9. Considere o problema dos jarros d'água (enunciado reproduzido abaixo), cuja representação formal no modelo de grafos já foi explorada.

Enunciado: Dadas duas jarras não graduadas, uma com capacidade de 5 litros, cheia d'água, e outra com capacidade de 2 litros, vazia, como se pode obter exatamente 1 litro d'água na jarra de 2 litros? A água pode ser transferida de uma jarra a outra ou jogada fora, mas exatamente 5 litros estão disponíveis inicialmente.

PEDE-SE:

- 9.1. Indique como seria essa representação, agora no modelo de um sistema de produção.
- 9.2. Um exemplo de conjunto de conflito
- 9.3. Duas regras de produção que expressem ações atômicas válidas para esse problema
- 9.4. Um possível estado final para esse problema.

⁴ Rowe, p. 218, exercício 9-13.



6. Considere que sua lista OPEN, de trajetórias potenciais, seja expressa pela sequência de números das regras de sua base de conhecimento e que, caso uma mesma regra seja escolhida, a cada característica verdadeira de seu corpo, um valor que indica a relevância da característica seja acumulado, de modo que, se a regra for aplicada integralmente, o custo da trajetória expressará a relevância total das características correspondentes. Inicialmente, as possíveis trajetórias podem ser expressas em OPEN pelo conjunto de regras iniciais aplicáveis ao problema delineado, da seguinte forma²:

$$OPEN = \{ \text{regra}(1,0), \text{regra}(2,0), \text{regra}(3,0), \dots, \text{regra}(1000,0) \}$$

Suponha que, quanto mais relevante for o conjunto de características desejadas por um cliente, maior será sua pré-disposição para comprar o cão.

RESPONDA O MAIS CONCISAMENTE POSSÍVEL:

- 6.1. Indique *COMO* os índices de relevância das características podem ser indicados em cada regra. Use, para isso, uma regra genérica como

```

identify(Cão,NumeroDaRegra) se
    característica 1 for verdadeira e
    característica 2 for verdadeira e
    ..... e
    característica N for verdadeira
  
```

para N qualquer (conjunto de possíveis estados de seu espaço de solução).

- 6.2. Indique *COMO* esses índices serão acumulados para que a lista OPEN guarde, além das possíveis trajetórias a expandir, também sua relevância acumulada.

² Não se conhecem as características do cão e, portanto, a relevância de qualquer regra ainda é inexpressiva. A representação ilustrada tem a notação *regra(Nro_da_regra,Relevância_total_das_caract_indicadas_nessa_regra)*.