

## Lista de exercícios genérica

### Estratégias de busca cega

Obs.: Esta lista inclui revisão de Prolog. Para explorá-la, use o Luger & Stubblefield e Araribóia, no mínimo.

1. Seja uma situação do mundo em que as seguintes proposições são verdadeiras<sup>1</sup>:

a	$a \& b \rightarrow d$
b	$a \& c \rightarrow e$
c	$b \& d \rightarrow f$
$f \rightarrow g$	$a \& e \rightarrow h$

Considerando a busca no espaço de solução desse conjunto, responda:

- 1.1. h é verdadeira?
- 1.2. Se b for falsa, h é verdadeira?
- 1.3. Qual é o caminho mais curto (i.e., a sequência mais curta de inferências) para mostrar que X é verdadeira, sendo X uma proposição qualquer nesse mundo?

Obs.: Para este exercício, você deverá *primeiro* reproduzir o espaço de solução inteiro do Prolog, a partir da base de conhecimento dada pelo conjunto de proposições acima. Depois deverá simular a prova de teoremas do Prolog considerando esse espaço.

2. Diga qual dos métodos de busca você usaria, para resolver os seguintes problemas (*breadth-first* ou *depth-first*), justificando sua resposta:

- 2.1. Um programa de jogo de xadrez.
- 2.2. Um programa de diagnóstico médico a partir do exame clínico do paciente.
- 2.3. Um planejador de caminho de um robô que vai de uma posição A a uma posição B em uma sala.
- 2.4. Um programa que acha a melhor sequência de passos de manufatura para chegar a um produto acabado a partir de um conjunto de matérias-primas.

3. Seja o seguinte trecho de código em Prolog, em um programa de busca de solução<sup>2</sup>:

```
busca(T,Solução) :-  
    atinge_a_meta(T),  
    Solução = T.  
busca(T,Solução) :-  
    estende_ate_Filho(T, ExtensaoDeT),  
    busca(ExtensaoDeT, Solução).  
  
estende_ate_Filho([NoPai | Trajetoria], [NoFilho, NoPai | Trajetoria]) :-  
    deriva(NoFilho, NoPai),  
    nao_produz_circulos(NoFilho, Trajetoria).
```

Pergunta-se:

- 3.1. O que faz esse código?
- 3.2. O método apresentado nesse código coincide com o método de resolução do próprio Prolog. Qual é esse método?
- 3.3. Indique o trecho de código que implementa o método que você especificou anteriormente e a transformação resultante de sua aplicação.

<sup>1</sup> Exemplo 3.3.2, do L&S (p. 105).

<sup>2</sup> Código extraído de [Araribóia, 1988, Listagem 5.6].

**Obs.:** este código reflete o algoritmo de busca cega visto em aula (arquivo = algoritmos.doc). Use-o como auxiliar para fazer este exercício.

4. Transforme as seguintes sentenças em cláusulas em Prolog:

- João gosta de toda espécie de comida.
- Maças são comida.
- Galinha é comida.
- Qualquer coisa que alguém coma é comida.
- Chamamos qualquer pessoa de "alguém".

Aponte as comidas que João gosta de comer, da seguinte forma:

- 4.1. Especifique a meta que deve levar à resposta solicitada.
- 4.2. Utilize o raciocínio do provador Prolog para mostrar como encontrou as respostas.

5. Considerando o jogo individual *8-puzzle*, já bastante explorado em sala de aula e, sobretudo, por Luger & Stubblefield, cuja descrição genérica como um grafo é dada pela tripla  $[A, S, GD]$ , para<sup>3</sup>:

S: estado inicial =

2	8	3
1	6	4
7		5

GD = {UnicoEstado}, sendo UnicoEstado =

1	2	3
8		4
7	6	5

e  $A = \{\text{movaCima, movaBaixo, movaEsquerda, movaDireita}\}$

Diga como o sistema de produção assim descrito achará a solução quando:

- A direção considerada for *goal-driven*
- A direção considerada for *data-driven*

6. Dê a representação em grafo para os seguintes problemas:

- 6.1. Problema da Travessia (fazendeiro e carneiro ou repolho e lobo), descrito em (Araribóia, 1989, p. 91).
- 6.2. Fichas pretas e brancas (descrição também no Araribóia)

7. Em *depth-first*, os filhos de cada nó são gerados e expandidos em uma ordem determinada. Se essa ordem é fixa, há outros modos de se analisar o problema para tentar otimizar a busca. Que critério(s) deveria ser considerado para garantir a melhor ordem para gerá-los?

8. Quais são os problemas principais da resolução de problemas em IA?

9. O que é que torna imprescindível investigar diversas estratégias de resolução de problemas em IA?

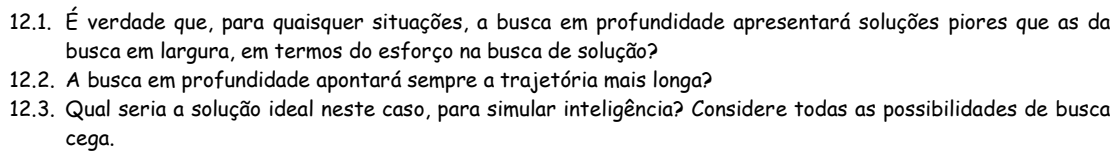
10. Em grafos AND/OR há situações em que o espaço de busca da solução pode ser consideravelmente reduzido? Diga quando isso ocorre, justificando (se quiser, use um diagrama para ilustrar).

11. Qual a diferença entre busca cega e busca informada em Inteligência Artificial?

12. Considere o seguinte espaço de solução para responder o que segue, justificando sempre:

---

<sup>3</sup> Exercício 8, L&S, p. 147 - este exercício está RESOLVIDO nesse capítulo.



```
membro(X,[X|_]).
membro(X,[Y|Z]) :- membro(X,Z).
```

b) Mostre o caminho de prova (i.e., o modo como o Prolog aplica as regras do programa acima) para a seguinte meta:

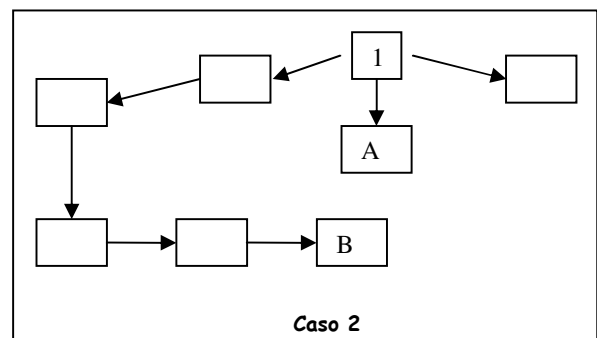
c) Suponha que o programa é alterado inserindo-se um corte na primeira regra, de modo que o mesmo resulte em

```
membro(X,[X|_]) :- !.  
membro(X,[Y|Z]) :- membro(X,Z).
```

14. Sabemos que, na recursividade, temos, no mínimo, uma declaração trivial (T) e uma geral, recursiva (G). Responda:

14.2. Em um programa em Prolog, quais devem ser as características comuns a  $T$  e  $G$  para que elas sejam declarações de um mesmo programa recursivo?

15. Sabemos que, num espaço de estados, quanto mais próximo um nó objetivo estiver da raiz, estado inicial do problema, mais curto será o caminho de solução. Para dois nós objetivo A e B, A situado em nível n e B, em nível k,  $k \gg n$ , compare o esforço da busca da solução quando são consideradas a busca em largura e a busca em profundidade, segundo os casos ilustrados abaixo ( $n = 2$  e  $k = 6$ ). Fixe o percurso na estrutura em um dos sentidos - LR ou RL.



16. Diga se as afirmações que seguem são verdadeiras ou falsas, justificando sua resposta consistentemente, de acordo com seu conhecimento sobre os aspectos teóricos da busca em espaço de solução:
- 16.1. Conhecendo-se o grafo completo representativo da busca de solução para um problema de IA, ele pode ser dado como entrada de um programa de IA. Neste caso, o programa pode ser convencional, isto é, um programa de busca em grafo qualquer. Não haveria, assim, a necessidade de se utilizarem métodos sofisticados definidos por uma Engenharia de Conhecimento complexa como a da IA.
- 16.2. Considerando os algoritmos dados, de busca cega, a busca em largura será mais inconveniente do que a busca em profundidade quando o número médio de filhos dos nós do grafo de solução for alto, pois acarretará um crescimento exponencial da lista OPEN, de candidatos a expandir.
- 16.3. Considere um algoritmo genérico de busca cega. Se houver dois grupos de ações que levam a duas soluções com caminhos de comprimentos distintos para certa configuração de problema, a fixação de um grupo de ações para a escolha que leve ao caminho mais curto primeiro não garantirá maior eficiência de resolução para todos os grafos possíveis.