Inteligência Artificial
Tarefa - Busca Não Informada

Problema das Jarras

1. **Modelagem do Problema**: Considere que queremos representar o problema das jarras em Prolog. Sabemos que:

   - A jarra 1 tem capacidade de 4 litros e a jarra 2 tem capacidade de 3 litros.
   - Inicialmente, as jarras estão vazias.
   - O objetivo do problema é colocar 2 litros de água na jarra 1.
   - As ações que podemos fazer são:
     - encher completamente as jarras (representadas pelas constantes *encher1*, *encher2*)
     - esvaziar completamente as jarras (representadas pelas constantes *esvaziar1*, *esvaziar2*)
     - passar a água de uma jarra para outra (representadas pelas constantes *passar12*, *passar21*)
   - Vamos representar o conteúdo das jarras usando o par ordenado (J_1, J_2) onde $J_1 \in [0, 4]$ e $J_2 \in [0, 3]$.

   (a) Escreva os fatos Prolog que representam quais são os estados finais do problema. Use o predicado unário *objetivo*

   (b) Defina o predicado ternário *acao((J1,J2),ACAO,(J1a,J2a))* que transforma o estado das jarras *(J1,J2)* no estado *(J1a,J2a)* quando uma ação ACAO (ACAO $\in \{encher1, encher2, esvaziar1, esvaziar2, passar12, passar21\}$) for executada.

   Por exemplo,

   ```
   acao((J1,J2),encher1,(4,J2)):- J1<4.
   ```

   representa que se a jarra J1 tiver menos que 4 litros de água, após fazer a ação *encher1*, a jarra 1 fica cheia (com 4 litros de água) e a quantidade de água na jarra 2 (J2) não muda. *Atenção: A definição deve ser feita usando regras e não fatos Prolog.*

   (c) O predicado ternário *findall* permite obter todas as instâncias que satisfazem uma dada consulta. Por exemplo, se tivermos o programa Prolog:

   ```
   p(a).
   p(b).
   q(X):-p(X).
   ```

   e fizermos a consulta *?- findall(Y,q(Y),L).*, ou seja, queremos obter a lista L formada por todos os elementos Y que satisfazem *q(Y)*, obteremos como resposta L = *[a,b]*.

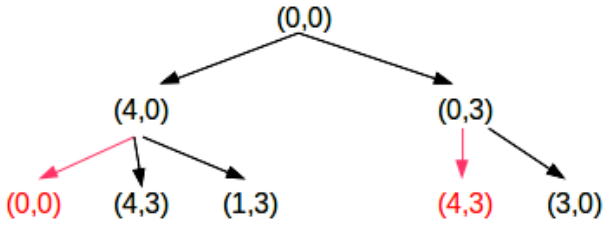
   Usando o predicado *findall*, defina o predicado binário *vizinho(N,FilhosN)* que dado uma configuração das jarras N retorna todas as configurações possíveis que podemos obter aplicando cada uma das ações definidas para o problema ao estado representado por N.

(d) Usando os predicados *objetivo* e *vizinhos*, implemente em Prolog o algoritmo de busca em largura apresentado em aula. Faça uma consulta que represente a busca de uma solução a partir do estado inicial (0,0). O que ocorre? Por que?

(e) Modifique o programa do item anterior para que seja guardada a sequência de configurações dos estados das jarras.

(f) Até agora não nos preocupamos em evitar os estados repetidos. Para isso, ao acrescentar os elementos novos na fronteira, devemos desconsiderar os nós que já foram gerados anteriormente. Podemos fazer isso comparando a lista formada pelos filhos de um dado nó com os nós já gerados, e acrescentar na fronteira apenas aqueles que estão na primeira lista e não na segunda.

Por exemplo, os nós em vermelho na figura devem ser desconsiderados.



Faça esta modificação no programa do item anterior. *Dica: use o predicado diferença de listas que foi resolvido na lista de exercícios de Prolog.*

(g) Repita cada exercício anterior supondo que agora queremos fazer uma busca em profundidade.