

INTELIGÊNCIA ARTIFICIAL

Ficha 5

1.

a) É um problema de estado único.

b) **Conjunto de estados:** conjunto de pares ordenados de inteiros (X,Y) ,
em que X representa o balde com o volume máximo de 8L.
em que Y representa o balde com o volume máximo de 5L.

Estados iniciais: vazios $(0,0)$.

Estados finais: $(4, _)$ ou $(_, 4)$.

(Ou seja, jarro de 8 litros com apenas 4 litros OU jarro de 5 litros com apenas 4 litros)

Operadores:

- Esvaziar um balde.
- Encher (completamente) um jarro.
- Despejar um jarro até que o segundo fique cheio.
- Despejar um jarro até que o primeiro fique vazio.

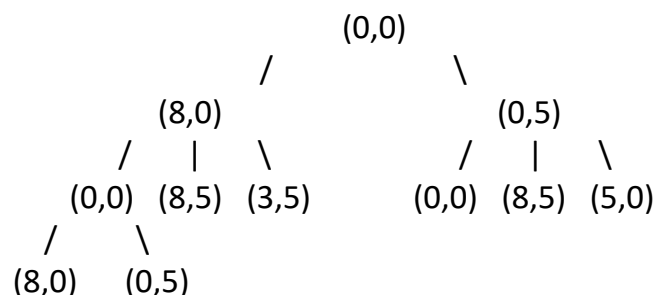
Transições (exemplo):

Transição	Estado Inicial	Condição	Estado Final	Descrição
1	$(0,0)$	If $(x < 8)$	$(8,y)$	Encher X
2	$(0,0)$	If $(y < 5)$	$(x,5)$	Encher Y
3	$(8,0)$	If $(x > 0)$	$(0,0)$	Esvaziar X
4	$(8,0)$	If $(y > 5)$	$(8,5)$	Esvaziar Y
5	$(8,0)$	If $(x > 0 \text{ e } y = 0)$	$(3,5)$	Despejar parte da água do jarro de 8L para o de 5L
6	$(0,5)$...		

Teste Objetivo: Um dos dois jarros deve conter somente 4 litros de água

Custo: Cada ação custa 1 unidade.

c) Pesquisa em largura:



Ou seja !!

- Os nós, em cada nível da árvore, são completamente examinados antes de se mover para o próximo nível.
- Uma busca em largura encontrará sempre o caminho menor entre o estado inicial e o estado-objectivo (final).
- O menor caminho é o caminho com o menor número de passos (não confundir com o caminho de menor custo).
- É necessário manter um conjunto de nós candidatos alternativos e não apenas o nó candidato como na busca em profundidade.
- O conjunto de nós candidatos é constituído por todos o nós de um dado nível da árvore de busca.
- Entretanto, mesmo este conjunto de caminhos candidatos não é suficiente então deve-se extrair um caminho de solução do processo de busca.
- Portanto, ao invés de manter-se um conjunto de nós candidatos, deve-se manter um conjunto de caminhos-candidatos.

Exemplo:

1. Conjunto de Candidatos Inicial:

$[[(0,0)]]$

2. Gera extensões de $[(0,0)]$:

$[[(8,0),(0,0)], [(0,5), (0,0)]]$ (1.º nível da árvore)

3. Remove o 1.º caminho candidato $[(8,0),(0,0)]$ e gera extensões deste caminho, colocando-as no final do conjunto:

$[[(0,5), (0,0)], [(0,0),(8,0),(0,0)], [(3,5),(8,0),(0,0)], [(8,5),(8,0),(0,0)]]$ (2.º nível da árvore)

4. Remove $[(0,5), (0,0)]$ e acrescenta a sua extensão ao final do conjunto de candidatos, produzindo:

$[[(0,0), (8,0),(0,0)], [(3,5),(8,0),(0,0)], [(8,5),(8,0),(0,0)], [(0,0),(0,5),(0,0)], [(5,0),(0,5),(0,0)], [(8,5),(0,5),(0,0)]]$ (2.º nível da árvore)

5. E continua a repetir o processo de extensão até a busca encontrar um caminho que tenha o nó-objectivo/final à cabeça ($[(_ ,4)]$ ou $[(4, _)]$) e assim este caminho apresentado com uma solução.

Resolver em Prolog:

```
inicial( jarros(0,0) ).  
final( jarros( \_ ,4) ).  
final( jarros(4, \_ ) ).
```

```
transicao( jarros(V1,V2), vaziao(1), jarros(0,V2) ) :- V1 < 0.  
transicao( jarros(V1,V2), vaziao(2), jarros(V1,0) ) :- V2 < 0.
```

transicao(jarros(V_1, V_2), encher(1), jarros(8, V_2)) :- $V_1 < 8$.

transicao(jarros(V_1, V_2), encher(2), jarros($V_1, 5$)) :- $V_2 < 5$.

transicao(jarros(V_1, V_2), encher(1,2), jarros(NV_1, NV_2)) :-

$V_1 > 0$,

NV_1 is $\max(V_1 - 5 + V_2, 0)$, % garantir que o resultado ou é 0 ou é superior a 0

$NV_1 < V_1$,

NV_2 is $V_2 + V_1 - NV_1$.

transicao(jarros(V_1, V_2), encher(2,1), jarros(NV_1, NV_2)) :-

$V_2 > 0$,

NV_2 is $\max(V_1 + V_2 - 8, 0)$, % garantir que o resultado ou é 0 ou é superior a 0

$NV_2 < V_2$,

NV_1 is $V_1 + V_2 - NV_2$.