

1º Teste ou Exame de Inteligência Artificial
5/4/2016 - 11-13 horas

Grupo 1 Considere o problema de num tabuleiro de xadrez 8X8, ir da casa (1,1) até à casa (4,5) com um cavalo (move-se em L, 2 casas numa direcção mais uma casa na perpendicular a essa direcção).

1. Represente este problema como um problema de pesquisa no espaço de estados em Prolog (estado inicial e final, e os operadores de transição de estado)

estado - e(Linha,Coluna) - posição do cavalo no tabuleiro

estado_inicial(e(1,1)).

estado_final(e(4,5)).

op(e(Lc,Cc),move(M,N),e(Ls,Cs),1):-

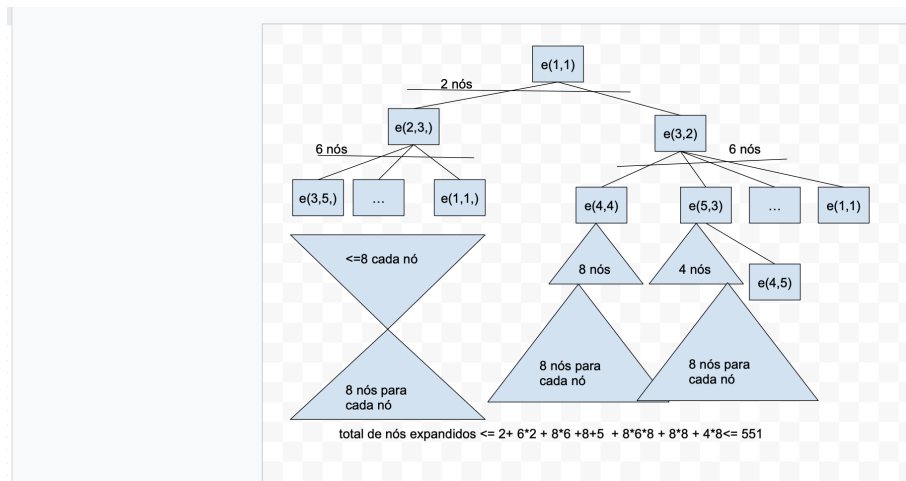
member((M,N),[

(1,2),(1,-2),(2,1),(2,-1),(-1,2),(-1,-2),(-2,1),(-2,-1)])

Ls is Lc+M, Cs is Cc+N, lim(Ls,Cs).

lim(L,C):- L>0,C>0,L<9,C<9.

2. Numa pesquisa em largura quantos estados são expandidos se o cavalo estiver na casa (4,4), e o quisermos colocar na casa (7,3).



Ou:

A solução está à profundidade 3 por isso o número máximo de nós expandidos são:

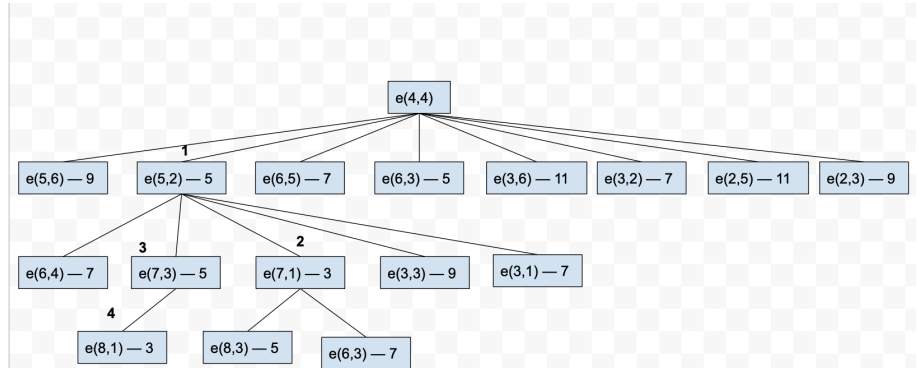
8 profundidade 1
 8*8 profundidade 2
 8*8*8 profundidade 3

8*8*8*8-8 na profundidade 4 assumindo que a solução é o nó mais à direita da árvore, quando encontra a solução não expande o nó

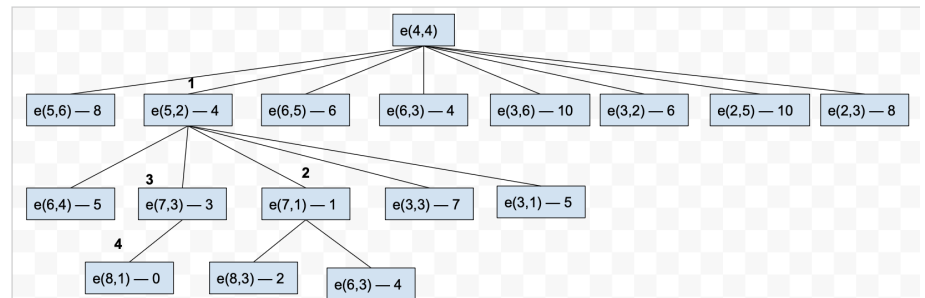
3. Considere a seguinte função de heurística que estima a distância entre duas casas:

```
f((X,Y),(W,Z),N):- modulo_dif(X,W,M),
                    modulo_dif(Y,Z,O),
                    N is M+O.
modulo_dif(A,B,C):- A>B, C is A-B.
modulo_dif(A,B,C):- C is B-A.
```

- (a) Desenhe a árvore de pesquisa do espaço de estados deste problema (quando o estado inicial é a casa (4,4) e o estado final é a casa (8,1) indicando a ordem de visita dos nós com o algoritmo A* e com o Ansioso, usando esta função para estimar a distância entre estados. (expanda a árvore só até ao nível 3. (Justifique a ordem de visita anotando nos nós expandidos o custo estimado).



Com o A*:



Com o ansioso:

- (b) Para este problema esta heurística é admissível? Justifique.

A herística não é admissível porque prevê mais passos para a solução do que os necessários, por exemplo, para o nó $e(5,2)$ prevê 4 passos e a solução está a 2.

Grupo 2 Considere o problema de colocar 4 cavalos num tabuleiro de 4x4 sem se atacarem.

Este problema pode ser representado e resolvido como um problema de satisfação de restrições (CSP).

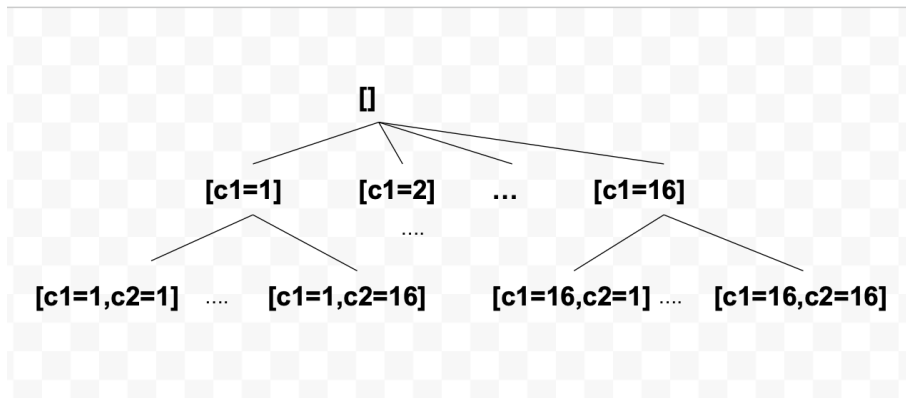
1. Proponha uma representação em Prolog para os estados do problema. Exemplifique representando o estado inicial deste problema.

Variáveis: 4 variáveis, $c1$, $c2$, $c3$ e $c4$ que representam os cavalos Domínio: 1..16, as casas onde os cavalos podem estar

estado inicial: $e([var(c1,[1..16]),-],..., var(c4,[1..16]),-],[])$

2. Defina o predicado $sucessor(Ei, Es)$ que sucede para todo Ei , Es em que Es é um sucessor de Ei . E expanda a árvore do espaço de estados até ao nível dois.

$sucessor(e([v(N,D,V)-R],E), e(R,[v(N,D,V)-E])):- member(V,D).$



3. Represente em Prolog as restrições deste problema. E defina o predicado $verifica\ restrições(Estado)$ que sucede quando um estado verifica todas as restrições.

As restrições do problema são:

- todos os cavalos têm de estar em posições diferentes
- um cavalo i não pode estar numa casa que um cavalo j ataque.

Definir o predicado $ataca(PosiçãoCavalo, Lista\ de\ posições\ que\ o\ cavalo\ ataca)$

```

ataca(1, [7,10]).
ataca(2, [9,11,8]).
...
ataca(16, [7,10]).

```

```

ve_restrições(e(Lni,Li)):- \+ (member(v(Ci,_Vi), Li),
                                member(v(Cj,_Vj), Li),
                                Ci\=Cj,
                                Vi==Vj; ataca(Vi,Ataca),
                                member(Vj,Ataca))

```

4. Indique, justificando, a melhor estratégia para resolver este problema como um problema de satisfação de restrições.

A melhor estratégia para este problema com a representação escolhida é a pesquisa backtracking com forward checking, o forward checking deve cortar do domínio das variáveis por instanciar o valor atribuído a uma variável e os valores atacados pelo cavalo que é representado pela variável.

Por exemplo, se o cavalo c1 é instanciado com o valor 1, as variáveis c2,c3 e c4 devem ficar com o domínio que tinham menos os valores 1, 7 e 10 (7 e 10 são as casas atacadas por um cavalo na casa 1).