

INTELIGÊNCIA ARTIFICIAL

Turma C

Juan Henrique dos Santos 594946

Trabalho 1

Busy Police (Atari)



FATOS E REGRAS

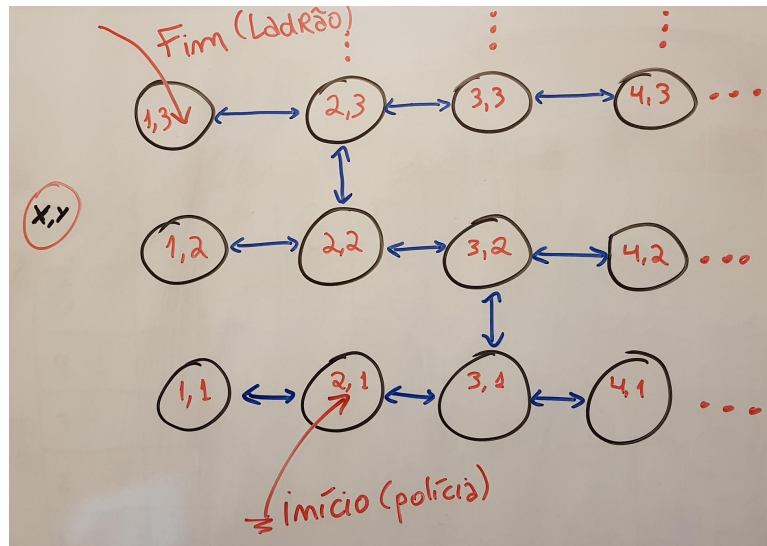
O projeto utiliza busca em profundidade.

Para implementar tal busca, foi escolhido criar fatos de forma ao final o resultado se torne um imenso grafo.

- Conexões horizontais criadas através de caminhos possíveis de 1 até 10.
- Conexões verticais criadas através do uso das escadas, com 5 possíveis andares.

Para facilitar ainda mais o entendimento, foram utilizadas as coordenadas cartesianas.

FATOS E REGRAS



Exemplo de um possível grafo

```

14 %MONTANDO AS FOLHAS DO GRAFO (HORIZONTAL)
15 caminho(posicao(1,1),posicao(2,1)).
16 caminho(posicao(2,1),posicao(3,1)).
17 caminho(posicao(3,1),posicao(4,1)).
18 caminho(posicao(4,1),posicao(5,1)).
19 caminho(posicao(5,1),posicao(6,1)).
20 caminho(posicao(6,1),posicao(7,1)).
21 caminho(posicao(7,1),posicao(8,1)).
22 caminho(posicao(8,1),posicao(9,1)).
23 caminho(posicao(9,1),posicao(10,1)).
24 caminho(posicao(1,2),posicao(2,2)).
25 caminho(posicao(2,2),posicao(3,2)).
26 caminho(posicao(3,2),posicao(4,2)).
27 caminho(posicao(4,2),posicao(5,2)).
28 caminho(posicao(5,2),posicao(6,2)).
29 caminho(posicao(6,2),posicao(7,2)).
30 caminho(posicao(7,2),posicao(8,2)).
31 caminho(posicao(8,2),posicao(9,2)).
32 caminho(posicao(9,2),posicao(10,2)).
33 caminho(posicao(1,3),posicao(2,3)).
34 caminho(posicao(2,3),posicao(3,3)).
35 caminho(posicao(3,3),posicao(4,3)).
36 caminho(posicao(4,3),posicao(5,3)).
37 caminho(posicao(5,3),posicao(6,3)).
38 caminho(posicao(6,3),posicao(7,3)).
39 caminho(posicao(7,3),posicao(8,3)).
40 caminho(posicao(8,3),posicao(9,3)).
41 caminho(posicao(9,3),posicao(10,3)).
42 caminho(posicao(1,4),posicao(2,4)).
43 caminho(posicao(2,4),posicao(3,4)).
44 caminho(posicao(3,4),posicao(4,4)).
45 caminho(posicao(4,4),posicao(5,4)).
46 caminho(posicao(5,4),posicao(6,4)).
47 caminho(posicao(6,4),posicao(7,4)).
48 caminho(posicao(7,4),posicao(8,4)).
49 caminho(posicao(8,4),posicao(9,4)).
50 caminho(posicao(9,4),posicao(10,4)).
51 caminho(posicao(1,5),posicao(2,5)).
52 caminho(posicao(2,5),posicao(3,5)).
53 caminho(posicao(3,5),posicao(4,5)).
54 caminho(posicao(4,5),posicao(5,5)).
55 caminho(posicao(5,5),posicao(6,5)).
56 caminho(posicao(6,5),posicao(7,5)).
57 caminho(posicao(7,5),posicao(8,5)).
58 caminho(posicao(8,5),posicao(9,5)).
59 caminho(posicao(9,5),posicao(10,5)).

```

FATOS E REGRAS

- Conexão vertical dos elementos do grafo

FATOS E REGRAS

```
62 %MONTANDO AS FOLHAS DO GRAFO (VERTICAL)
63 escada(posicao(1,2),posicao(1,3)).
64 escada(posicao(4,1),posicao(4,2)).
65 escada(posicao(5,4),posicao(5,5)).
66 escada(posicao(10,3),posicao(10,4)).
67
68 existe_subida_escada(X, Y):- escada(posicao(X,Y),_).
69
70 carrinho(posicao(2,1)).
71 carrinho(posicao(2,5)).
72 carrinho(posicao(3,4)).
73 carrinho(posicao(6,1)).
74 carrinho(posicao(7,4)).
75 carrinho(posicao(8,2)).
76 carrinho(posicao(10,1)).
77
```

Conexão vertical entre os elementos do grafo, através da escada.

FUNCIONALIDADE EXTRA

Para prender, o policial precisa de uma algema, certo?

```
8 % ===== FUNCAO EXTRA =====  
9 algemas(posicao(1,1)).  
10
```

FATOS E REGRAS

Esta parte do código possibilita que as ligações entre os elementos do grafo sejam bidirecionais, além de verificar se existe tal caminho.

```
79 %VERIFICO SE AS FOLHAS ESTÃO CONECTADAS (CAMINHO HORIZONTAL)
80 existe_caminho(X,Y,X2,Y2):- caminho(posicao(X,Y), posicao(X2,Y2)).
81 existe_caminho(X,Y,X2,Y2):- caminho(posicao(X2,Y2), posicao(X,Y)).
82 %VERIFICO SE AS FOLHAS ESTÃO CONECTADAS (CAMINHO VERTICAL)
83 existe_caminho(X,Y,X2,Y2):- escada(posicao(X,Y), posicao(X2,Y2)).
84 existe_caminho(X,Y,X2,Y2):- escada(posicao(X2,Y2), posicao(X,Y)).
```


AÇÕES POLICIAL

```
89 % ===== FUNCAO EXTRA =====
90 acao(pegar_algema, estado(posicao(X,Y), sem_algemas, SENTIDO), estado(posicao(X,Y), com_algemas, SENTIDO)):- algemas(posicao(X,Y)).
91
92
93 acao(subir, estado(posicao(X, Y), ALGEMAS, SENTIDO), estado(posicao(X, Y2), ALGEMAS, SENTIDO)):- Y2 is Y+1, existe_caminho(X, Y, X, Y2).
94
95
96 acao(virar_dir, estado(posicao(X, Y), ALGEMAS, esquerda), estado(posicao(X, Y), ALGEMAS, direita)).
97 acao(andar, estado(posicao(X, Y), ALGEMAS, direita), estado(posicao(X2, Y), ALGEMAS, direita)):- X2 is X + 1, existe_caminho(X, Y, X2, Y), not(carrinho(posicao(X2, Y))).
98
99
100 acao(pular, estado(posicao(X, Y), ALGEMAS, direita), estado(posicao(X2, Y), ALGEMAS, direita)):- X1 is X + 1, X2 is X1 + 1, %incremento das variaveis
101 carrinho(posicao(X1,Y)), %verifico se existe carrinho no meio
102 not(carrinho(posicao(X2,Y))), %se o espaco para onde o policial irá possui não tem carrinho
103 existe_caminho(X, Y, X1, Y), %se existe o caminho de transicao
104 existe_caminho(X1, Y, X2, Y), %se existe o caminho para o novo local
105 not(existe_subida_escada(X2, Y)), %se o novo local não tem uma escada para atrapalhar
106 not(ladrao(posicao(X2,Y))). %se o novo local não tiver o ladrão escondido
107
108
109 acao(descer, estado(posicao(X, Y), ALGEMAS, SENTIDO), estado(posicao(X, Y2), ALGEMAS, SENTIDO)):- Y2 is Y-1, existe_caminho(X, Y, X, Y2).
110
111
112 acao(virar_esq, estado(posicao(X, Y), ALGEMAS, direita), estado(posicao(X, Y), ALGEMAS, esquerda)).
113 acao(andar, estado(posicao(X, Y), ALGEMAS, esquerda), estado(posicao(X2, Y), ALGEMAS, direita)):- X2 is X-1, existe_caminho(X, Y, X2, Y), not(carrinho(posicao(X2, Y))).
114
115
116 acao(pular, estado(posicao(X, Y), ALGEMAS, esquerda), estado(posicao(X2, Y), ALGEMAS, esquerda)):- X1 is X - 1, X2 is X1 - 1, %incremento das variaveis
117 carrinho(posicao(X1,Y)), %verifico se existe carrinho no meio
118 not(carrinho(posicao(X2,Y))), %se o espaco para onde o policial irá possui não tem carrinho
119 existe_caminho(X, Y, X1, Y), %se existe o caminho de transicao
120 existe_caminho(X1, Y, X2, Y), %se existe o caminho para o novo local
121 not(existe_subida_escada(X2, Y)), %se o novo local não tem uma escada para atrapalhar
122 not(ladrao(posicao(X2,Y))). %se o novo local não tiver o ladrão escondido
```

AÇÕES POLICIAL

- Pegar algema;
- Subir/descer escada;
- Andar;
- Pular;
- Alterar sentido (direita/esquerda);

BUSCA EM PROFUNDIDADE

```
129 %solucao por busca em profundidade (bp)
130 solucao_bp(X, Y, Sentido, Solucao) :- bp([], estado(posicao(X, Y), sem_algemas, Sentido), Solucao).
131
132 %Se o primeiro estado da lista é meta, retorna a meta
133 bp(Caminho, Estado, [Estado|Caminho]) :- meta(Estado).
134 %se falhar, coloca o no caminho e continua a busca
135 bp(Caminho, Estado, Solucao) :- acao(_, Estado, Prox_Estado), not(pertence(Prox_Estado,[Estado|Caminho])), bp([Estado|Caminho],Prox_Estado,Solucao).
136
137
138 %PROCURA OS CAMINHOS PARA PRENDER O LADRÃO
139 procura_ladrao(X,Y, Caminho):- solucao_bp(X, Y, direita, Solucao), formata_saida(Solucao, Caminho).
140
```

EXEMPLOS DE USO

```
?- procura_ladrao(5,1,X).
X = [estado(posicao(9, 5), com_algemas, direita), estado(posicao(8, 5), com_algemas, direita), estado(posicao(7, 5), com_algemas, direita), estado(posicao(6, 5), com_algemas, direita), estado(posicao(5, 5), com_algemas, direita), estado(posicao(5, 4), com_algemas, direita), estado(posicao(6, 4), com_algemas, esquerda), estado(posicao(8, 4), com_algemas, esquerda), estado(posicao(8, 4), com_algemas, direita), estado(posicao(9, 4), com_algemas, esquerda), estado(posicao(9, 4), com_algemas, direita), estado(posicao(10, 4), com_algemas, esquerda), estado(posicao(10, 4), com_algemas, direita), estado(posicao(10, 3), com_algemas, direita), estado(posicao(9, 3), com_algemas, direita), estado(posicao(8, 3), com_algemas, direita), estado(posicao(7, 3), com_algemas, direita), estado(posicao(6, 3), com_algemas, direita), estado(posicao(5, 3), com_algemas, direita), estado(posicao(4, 3), com_algemas, direita), estado(posicao(3, 3), com_algemas, direita), estado(posicao(2, 3), com_algemas, direita), estado(posicao(1, 3), com_algemas, direita), estado(posicao(1, 2), com_algemas, direita), estado(posicao(2, 2), com_algemas, esquerda), estado(posicao(2, 2), com_algemas, direita), estado(posicao(3, 2), com_algemas, esquerda), estado(posicao(3, 2), com_algemas, direita), estado(posicao(4, 2), com_algemas, esquerda), estado(posicao(4, 2), com_algemas, direita), estado(posicao(4, 1), com_algemas, direita), estado(posicao(3, 1), com_algemas, direita), estado(posicao(1, 1), com_algemas, direita), estado(posicao(1, 1), com_algemas, esquerda), estado(posicao(1, 1), sem_algemas, esquerda), estado(posicao(3, 1), sem_algemas, esquerda), estado(posicao(3, 1), sem_algemas, direita), estado(posicao(4, 1), sem_algemas, esquerda), estado(posicao(4, 2), sem_algemas, esquerda), estado(posicao(4, 2), sem_algemas, direita), estado(posicao(4, 1), sem_algemas, direita), estado(posicao(5, 1), sem_algemas, esquerda), estado(posicao(7, 1), sem_algemas, esquerda), estado(posicao(7, 1), sem_algemas, direita), estado(posicao(5, 1), sem_algemas, direita)] ,
```

```
?- ■
```

EXEMPLOS DE USO

```
?- procura_ladrao(5,1,X).  
  <- andar <- andar <- andar <- andar <- subir <- andar <- pular <- virar_esq <- andar <- virar_esq <- a  
  ndar <- virar_esq <- subir <- andar <- andar <- andar <- andar <- andar <- andar <- andar <- a  
  ndar <- subir <- andar <- virar_esq <- andar <- virar_esq <- andar <- virar_esq <- subir <- andar <- pu  
  lar <- virar_dir <- pegar_algema <- pular <- virar_esq <- andar <- descer <- virar_esq <- subir <- anda  
  r <- pular <- virar_esq <- pular  
X = [estado(posicao(9, 5), com_algemas, direita), estado(posicao(8, 5), com_algemas, direita), estado(p  
osicao(7, 5), com_algemas, direita), estado(posicao(6, 5), com_algemas, direita), estado(posicao(5, 5),  
com_algemas, direita), estado(posicao(5, 4), com_algemas, direita), estado(posicao(6, 4), com_algemas,  
esquerda), estado(posicao(8, 4), com_algemas, esquerda), estado(posicao(8, 4), com_algemas, direita),  
estado(posicao(9, 4), com_algemas, esquerda), estado(posicao(9, 4), com_algemas, direita), estado(posic  
ao(10, 4), com_algemas, esquerda), estado(posicao(10, 4), com_algemas, direita), estado(posicao(10, 3),  
com_algemas, direita), estado(posicao(9, 3), com_algemas, direita), estado(posicao(8, 3), com_algemas,  
direita), estado(posicao(7, 3), com_algemas, direita), estado(posicao(6, 3), com_algemas, direita), es  
tado(posicao(5, 3), com_algemas, direita), estado(posicao(4, 3), com_algemas, direita), estado(posicao(  
3, 3), com_algemas, direita), estado(posicao(2, 3), com_algemas, direita), estado(posicao(1, 3), com_al  
gemas, direita), estado(posicao(1, 2), com_algemas, direita), estado(posicao(2, 2), com_algemas, esquer  
da), estado(posicao(2, 2), com_algemas, direita), estado(posicao(3, 2), com_algemas, esquerda), estado(  
posicao(3, 2), com_algemas, direita), estado(posicao(4, 2), com_algemas, esquerda), estado(posicao(4, 2  
) , com_algemas, direita), estado(posicao(4, 1), com_algemas, direita), estado(posicao(3, 1), com_algema  
s, direita), estado(posicao(1, 1), com_algemas, direita), estado(posicao(1, 1), com_algemas, esquerda),  
estado(posicao(1, 1), sem_algemas, esquerda), estado(posicao(3, 1), sem_algemas, esquerda), estado(pos  
icao(3, 1), sem_algemas, direita), estado(posicao(4, 1), sem_algemas, esquerda), estado(posicao(4, 2),  
sem_algemas, esquerda), estado(posicao(4, 2), sem_algemas, direita), estado(posicao(4, 1), sem_algemas,  
direita), estado(posicao(5, 1), sem_algemas, esquerda), estado(posicao(7, 1), sem_algemas, esquerda),  
estado(posicao(7, 1), sem_algemas, direita), estado(posicao(5, 1), sem_algemas, direita)] ■
```

EXEMPLOS DE USO

```
?- procura_ladrao(11,1,X).  
false.
```

Não existe posição de partida no grafo.

BIBLIOGRAFIA

Slides disponibilizados no AVA.

Matéria: Inteligência Artificial,
ministrada pelo Professor Murilo Naldi.