Relatório do 1º Trabalho Prático

Inteligência Artificial

Universidade de Évora
Engenharia Informática 2020/2021

Docente: Irene Rodrigues
Discentes: Leonardo Catarro, 43025
Diogo Solipa, 43071

Desenvolvimento

Exercício 1

a) Espaço de estados e operadores de transição

```
% portas bloqueadas entre salas
% bloqueada(sala_1, sala_2).
bloqueada((1,5), (1,6)).
bloqueada((2,6), (1,6)).
bloqueada((1,7), (1,6)).

bloqueada((2,7), (3,7)).
bloqueada((4,7), (3,7)).

bloqueada((2,6), (3,6)).
bloqueada((4,6), (3,6)).
bloqueada((3,5), (3,6)).

bloqueada((3,2), (4,2)).
bloqueada((4,1), (4,2)).
bloqueada((5,2), (4,2)).

bloqueada((3,3), (4,3)).
bloqueada((5,3), (4,3)).

bloqueada((3,4), (4,4)).
bloqueada((5,4), (4,4)).
bloqueada((4,5), (4,4)).

bloqueada((7,5), (7,6)).
bloqueada((6,6), (7,6)).
bloqueada((7,7), (7,6)).
```

Restrições

```
% Regra que utiliza as operacoes que o agente pode usar
% op(estado_atual, operacao, estado_seguinte, custo).

op((X, Y), direita, (W, Y), 1):-
    tamanho(T),
    X < T,
    W is X+1,
    \+ bloqueada((X, Y), (W, Y)),
    nao_visitadas((W, Y)).

op((X, Y), cima, (X, Z), 1):-
    tamanho(T),
    Y < T,
    Z is Y+1,
    \+ bloqueada((X, Y), (X,Z)),
    nao_visitadas((X,Z)).

op((X, Y), baixo, (X, Z), 1):-
    Y > 1,
    Z is Y-1,
    \+ bloqueada((X, Y), (X,Z)),
    nao_visitadas((X,Z)).

op((X, Y), esquerda, (W, Y), 1):-
    X > 1,
    W is X-1,
    \+ bloqueada((X, Y), (W, Y)),
    nao_visitadas((W, Y)).
```

Operações de transição

```
% indica o tamanho do tabuleiro
tamanho(7).

% verifica se a sala ja foi visitada, e caso nao tenha sido, marca como tal
nao_visitadas(X):-
    \+ visitadas(X),
    asserta(visitadas(X)).

% tuplo com as coordenadas onde o agente se encontra
estado_inicial((2,1)).
visitadas((2,1)).
% tuplo com as coordenadas onde o agente pretende chegar
estado_final((5,7)).
```

Espaço de estados

b) Código Prolog do algoritmo de pesquisa em profundidade

```
% Pesquisa em Profundidade
pesquisa_profundidade([no(E,Pai,Op,C,P)|_],no(E,Pai,Op,C,P)) :-
    retract(visitados(X)),
    Y is X + 1,
    asserta(visitados(Y)),
    estado_final(E).
pesquisa_profundidade([E|R],Sol):-
    expande(E,Lseg),
    insere_inicio(Lseg,R,LFinal),
    retract(max_em_memoria(X)),
    length(LFinal, L),
    max(X, L, Z),
    asserta(max_em_memoria(Z)),
    pesquisa_profundidade(LFinal,Sol).
```

c)

i) 15 estados visitados
ii) 11 estados em memória

d) Heurísticas

Distância Euclidiana

```
% Heuristica admissivel(Distância Euclidiana)
heur2(P1, D) :- estado_final(PF),
                euclidean_distance(P1, PF, D).

euclidean_distance((X1, Y1), (X2, Y2), R) :- R is sqrt((X2-X1)^2 + (Y2-Y1)^2).
```

Distância de Manhattan

```
% Heuristica admissivel(Distância de Manhattan)
heur((X, Y), R):-
    estado_final(PF),
    distancia((X, Y), PF, R).

%Calcula a distancia
distancia((X, Y), (W, Z), D):-
    X1 is X-W,
    abs(X1, AX),
    Y1 is Y-Z,
    abs(Y1, AY),
    D is AX+AY.

abs(X, X):- X > 0.
abs(X, R):- R is -X.
```

e) Código Prolog da pesquisa a*

```
pesquisa_aux([no(E,Pai,Op,C,CH,P)|_],no(E,Pai,Op,C,CH,P)) :-
    retract(visitados(X)),
    Y is X + 1,
    asserta(visitados(Y)),
    estado_final(E).
pesquisa_aux([E|R],Sol):-
    expandeInformada(E,Lseg),
    insere_ordenado(Lseg,R,LFinal),
    retract(max_em_memoria(X)),
    length(LFinal, L),
    max(X, L, Z),
    asserta(max_em_memoria(Z)),
    pesquisa_aux(LFinal,Sol).

expandeInformada(no(E,Pai,Op,C,CH,P),L):-
    findall(no(En,no(E,Pai,Op,C,CH,P), Opn, Cnn, CHn, P1),
                (op(E,Opn,En,Cn), P1 is P+1, Cnn is Cn+C, heur2(En, H), CHn is Cnn + H),
                L).
```

f)
i, ii)

	Estados visitados	Estados em memória
Distância de Manhattan	12	13
Distância Euclidiana	19	9

Exercício 2

Este exercício é em tudo semelhante ao anterior, tirando o facto de o agente ter que empurrar uma caixa até ao estado final, ou seja, temos 2 “objetos” ao invés de 1 apenas.
Não nos foi possível terminar a resolução deste exercício, tendo apenas dado output “No”.
Ainda assim, implementamos todas as operações de transição e novos estados iniciais(um para a caixa e um para o agente) adaptadas a este problema. Basicamente, o que fizemos foi a cada transição feita, antes de esta ser aplicada, o agente deve se reposicionar de forma a puder empurrar a caixa para o estado seguinte, isto é, caso a caixa vá ser empurrada para a direita, o agente deve ser reposicionado à esquerda caixa(e isto para todas as restantes direções…).
Nota: A implementação das pesquisa, quer não informada quer informada, são as mesmas usadas para o exercício 1. Idem com as heurísticas utilizadas.

Operações de transição

```
% Regra que utiliza as operacoes que o agente pode usar
% op(estado_atual, operacao, estado_seguinte, custo).
op( ((Xa, Y), (Xc, Y)), direita, ((Wa, Y), (Wc, Y)), 1):-
    Xa =\= Xc-1 -> Xa is Xc-1,
    tamanho(T), Xc < T,
    Wa is Xa+1,
    Wc is Xc+1,
    \+ bloqueada((Xa, Y), (Wa, Y)),
    \+ bloqueada((Xc, Y), (Wc, Y)),
    nao_visitadas((Wa, Y)),
    nao_visitadas((Wc,Y)); fail.

op(((X, Ya), (X, Yc)), cima, ((X, Za), (X, Zc)), 1):-
    Ya =\= Yc-1 -> Ya is Yc-1,
    tamanho(T),
    Yc < T,
    Za is Ya+1,
    Zc is Yc+1,
    \+ bloqueada((X, Ya), (X,Za)),
    \+ bloqueada((X, Yc), (X,Zc)),
    nao_visitadas((X,Za)),
    nao_visitadas((X,Zc)); fail.

op(((X, Ya), (X, Yc)), baixo, ((X, Za), (X, Zc)), 1):-
    Ya =\= Yc+1 -> Ya is Yc+1,
    Yc > 1,
    Za is Ya-1,
    Zc is Yc-1,
    \+ bloqueada((X, Ya), (X,Za)),
    \+ bloqueada((X, Yc), (X,Zc)),
    nao_visitadas((X,Za)),
    nao_visitadas((X,Zc)); fail.

op( ((Xa, Y), (Xc, Y)), esquerda, ((Wa, Y), (Wc, Y)), 1):-
    Xa =\= Xc+1 -> Xa is Xc+1,
    Xc > 1, Wa is Xa-1,
    Wc is Xc-1,
    \+ bloqueada((Xa, Y), (Wa, Y)),
    \+ bloqueada((Xc, Y), (Wc, Y)),
    nao_visitadas((Wa, Y)),
    nao_visitadas((Wc, Y)); fail.
```

Conclusão

Com este trabalho conseguimos por em prática todo o conhecimento lecionado nas aulas teóricas e práticas da disciplina.
Embora tenhamos tido dificuldades na resolução do 2º exercício, visto que não percebemos bem como lidar com a caixa e o agente como 1 só, e como resolver o output “no” obtido, no geral consideramos que conseguimos aplicar o conhecimento adquirido.