

Inteligência Artificial - Trabalho 1

Elementos

35449 | Alexandre Rodrigues

35480 | Pedro Oliveira

Respostas

Grupo 1

Pergunta 1

Espaço de estados:

`estado_inicial(e(7, (2,0), [(0,0), (4,0), (0,2), (4,2)], [(2,1)], (5,3))).`

`estado_final(e(_, Pos, [], Carr, (5,3))):-`

`member(Pos, Carr).`

Operadores de transição:

`tr(
 e(Bat, Pos, Sujas, Carr, Dim),
 limpar,
 e(Bat1, Pos, Sujas1, Carr, Dim),`

1

):- Bat > 0,

Bat1 is Bat - 1,

member(Pos, Sujas),

delete(Sujas, Pos, Sujas1).

tr(

e(_, Pos, Sujas, Carr, Dim),

carrega,

e(Bat1, Pos, Sujas, Carr, Dim),

1

):- member(Pos,Carr),

Bat1 is 7.

tr(

e(Bat, (X, Y), Sujas, Carr, (N, M)),

cima,

e(Bat1, (X, Y1), Sujas, Carr, (N, M)),

1

) :-Bat > 0,

Y > 0,

Bat1 is Bat - 1,

Y1 is Y - 1.

tr(

e(Bat, (X, Y), Sujas, Carr, (N, M)),

direita,

e(Bat1, (X1, Y), Sujas, Carr, (N, M)),

```

1
) :-Bat > 0,
    Bat1 is Bat - 1,
    X1 is X + 1,
    X1 < N.

tr(
    e(Bat, (X, Y), Sujas, Carr, (N, M)),
    baixo,
    e(Bat1, (X, Y1), Sujas, Carr, (N, M)),
    1
) :-Bat > 0,
    Bat1 is Bat - 1,
    Y1 is Y + 1,
    Y1 < M.

tr(
    e(Bat, (X, Y), Sujas, Carr, (N, M)),
    esquerda,
    e(Bat1, (X1, Y), Sujas, Carr, (N, M)),
    1
) :-Bat > 0,
    Bat1 is Bat - 1,
    X > 0,
    X1 is X - 1.

```

Pergunta 2

Código fornecido nas aulas práticas:

```

pesquisa_largura(
    [ no(Ef, Pai, Op, C, P) | R],
    _,
    no(Ef, Pai, Op, C, P),
    stat(1, LL)) :-
    estado_final(Ef),
    length(R, L1),
    LL is L1 + 1.

```

```

pesquisa_largura( [N | R], V, S, Stats) :-
    N = no(E, _, _, _, _),
    member(E, V), !, % Não expandir estados visitados.
    pesquisa_largura(R, V, S, Stats).

```

```

pesquisa_largura([N | Resto], V, Solucao, stat(NN1, LL1)) :-
    N = no(E, _, _, _, _),
    expande_largura(N, Filhos),
    append(Resto, Filhos, Seguintes),
    pesquisa_largura(Seguintes, [E | V], Solucao, stat(NN, LL)),
    NN1 is NN + 1,
    length(Seguintes, LenSeguintes),
    max(LenSeguintes, LL, LL1).

```

```

expande_largura(no(E, A, O, C, P), Filhos) :-
    findall(
        no(Enovo, no(E, A, O, C, P), Onovo, Cnovo, Pnovo),
        ( tr(E, Onovo, Enovo, Cop),

```

$P_{novo} \text{ is } P + 1,$
 $C_{novo} \text{ is } C + C_{op}),$
Filhos).

Pergunta 3

Escolhemos este algoritmo porque nos permite ver todos os nós adjacentes ao estado atual.

Pergunta 3.i

O total de estados visitados é 1126.

Pergunta 3.ii

O número máximo de estados que ficaram simultaneamente em memória é 229.

Grupo 2

Pergunta 1

Uma heurística admissível é o cálculo da distância entre a posição atual e as casas sujas, selecionando a mais próxima, e verificando se, com a carga restante, consegue alcançar um carregador. Se sim, avança para essa casa. De seguida repete este procedimento.

heuristica(e(BAT, POS, SSUJAS, CARR,), VAL):-

$distSuja(POS, SSUJAS, X, POSX),$

$X1 \text{ is } X + 1,$

$distCarr(POSX, CARR, X2),$

$X2 = X2 + X1,$

X2 =< BAT,

VAL is X2.

heuristica(e(,,,),):- heuristica(e(,,,), 7).

%

%Distância para a próxima sala suja.

%

distSuja(_, [], _,).

distSuja(POS, [CAB/LISTA], RESULT, L) :-

manhattan(POS, CAB, COISA),

(nonvar(RESULT);write(RESULT),min(COISA, RESULT, MIN),

RESULT is MIN),

RESULT == COISA,

L is CAB,

distSuja(POS,LISTA, RESULT,L).

distSuja(POS, [LISTA], RESULT, L):- distSuja(POS, LISTA, RESULT, L).

%

%Distância entre a próxima sala suja e um carregador.

%

distCarr(_, [],).

distCarr(POSINI, [CAB/LISTA], DISTCARR):-

manhattan(POSINI, CAB, COISA2),

(nonvar(DISTCARR);min(COISA2, DISTCARR, MIN),

DISTCARR is MIN),

DISTCARR == COISA2,

distCarr(POSINI, LISTA, DISTCARR).

distCarr(POSINI, [LISTA], DISTCARR):- distCarr(POSINI, LISTA,

DISTCARR).

Pergunta 2

```
%
%algoritmo de pesquisa manhattan
%
manhattan((X,Y),(EndX,EndY),Val):-
    CurrX is EndX - X,
    changeSignal(CurrX,CurrX2),
```

```

    CurrY is EndY - Y,
    changeSignal(CurrY,CurrY2),

    Val is CurrX2 + CurrY2.

```

```
% muda o sinal se for negativo, caso contraria nao altera
changeSignal(N,M):- ((N < 0) -> M is N*(-1); M is N ).
```

Pergunta 2.i

Devido a não termos conseguido implementar na totalidade a heurística, não conseguimos responder às questões 2.i e 2.ii.

Pergunta 2.ii

Programas Usados

Para executar o programa, é necessário compilar o ficheiro `lib.pl`, de

seguida evocar a função “pesquisa/4”, como segue no exemplo:

1: [lib].

2: pesquisa('asp.pl'. largura, X, D).

Os ficheiros usados foram os seguintes:

- **lib.pl** : Ficheiro onde está implementado o código de pesquisa em largura
- **asp.pl** : Ficheiro onde estão implementados os estados inicial, final e os operadores de transição.
- **tools.pl** : Ficheiro com funções usadas para escrever o output.
- **lib2.pl** : Ficheiro onde está implementada a chamada da pesquisa.
- **heuristica.pl** : Ficheiro onde está implementada a função heurística que ajuda a escolher o caminho do aspirador, e o algoritmo de pesquisa “manhattan”.