Inteligência Artificial - Trabalho 1

Elementos

Número	Primeiro Nome + Apelido
35476	João Dias
35477	Eduardo Romão

Respostas

Grupo 1

Pergunta 1.1

Estado inicial e Estado final

Um estado é composto por um robo e por uma sala onde o mesmo se encontra. O robo contem a sua posição e a sua bateria. Já a sala é composta pela dimensão, a localização dos mosaicos sujos e a posição do carregador do robo.

```
estado_inicial(estado(
    robo(pos((2,0)), bateria(7)),
    sala(dim(5,3), sujas([(0,0),(4,0),(0,2),(4,2)]), carregad
or([(2,1)])
))).
```

```
estado_final(estado(
    robo(pos(A), bateria(_)),
    sala(_, sujas([]), carregador(B))
)):-
    member(A,B).
```

Transições

Existem seis transições:

- quatro para movimentar o robo na sala (cima, baixo, esquerda, direita);
- uma para carregar;
- e uma para limpar o mosaico.

```
tr(
    estado(robo(pos(X), bateria(Z)), sala(dim(N,M),sujas(Dirt
),Charger)),
    limpa,
    estado(robo(pos(X), bateria(Z1)), sala(dim(N,M),sujas(Dir
t2),Charger)),
    1
):-
    Z > 0, Z1 is Z - 1,
    member(X, Dirt),
    delete(Dirt, X, Dirt2).

tr(
    estado(robo(pos(X), bateria(_)), sala(dim(N,M),Dirt,carre
```

```
gador(Charger))),
    carrega,
    estado(robo(pos(X), bateria(Z1)), sala(dim(N,M),Dirt,carr
egador(Charger))),
    1
):-
    member(X, Charger),
    Z1 is 7.

tr(
    estado(robo(pos((X,Y)), bateria(Z)), sala(dim(N,M),Dirt,C
harger)),
    esquerda,
    estado(robo(pos((X1,Y)), bateria(Z1)), sala(dim(N,M),Dirt
,Charger)),
    1
):-
    Z > 0, Z1 is Z - 1,
    X >= 1,
    X1 is X - 1.

tr(
    estado(robo(pos((X,Y)), bateria(Z)), sala(dim(N,M),Dirt,C
harger)),
    direita,
    estado(robo(pos((X1,Y)), bateria(Z1)), sala(dim(N,M),Dirt
,Charger)),
    1
```

```
):-
    Z > 0, Z1 is Z - 1,
    X1 is X + 1,
    X1 < N.

tr(
    estado(robo(pos((X,Y)), bateria(Z)), sala(dim(N,M),Dirt,C
harger)),
    cima,
    estado(robo(pos((X,Y1)), bateria(Z1)), sala(dim(N,M),Dirt
,Charger)),
    1
):-
    Z > 0, Z1 is Z - 1,
    Y >= 1,
    Y1 is Y - 1.

tr(
    estado(robo(pos((X,Y)), bateria(Z)), sala(dim(N,M),Dirt,C
harger)),
    baixo,
    estado(robo(pos((X,Y1)), bateria(Z1)), sala(dim(N,M),Dirt
,Charger)),
    1
):-
    Z > 0, Z1 is Z - 1,
    Y1 is Y + 1,
    Y1 < M.
```

Pergunta 1.2

O melhor algoritmo de pesquisa para usar na resolução deste problema, é a pesquisa iterativa, pois é a que espande menos nós e ocupa menos espaço em memória. De forma a implementar esta pesquisa, utilizamos o algoritmo da pesquisa iterativa que nos foi fornecido em aula. Este algoritmo encontra-se no ficheiro `lib_ficha2.pl`.
Para correr este algoritmo usa-se o comando: `exercicio_1(iterativa).`.
Para correr em largura utiliza-se o comando `exercicio_1(largura).`.

Pergunta 1.3

1.3.1

O número de nós visitados é 151. Foi escolhido o algoritmo iterativo porque o em largura visita 1059.

1.3.2

O número de nós em memória é 52, na iterativa, enquanto que em largura são 235.

Programas Usados

- `lib_ficha2.pl`: contém os algoritmos de pesquisa em largura e pesquisa iterativa.
- `aspirador_op.pl`: contém as transições de estados do problema.
- `aspirador_est.pl`: contém o estado inicial e final do prolema. Também contem predicados para calcular o maximo e o minimo e

um predicado para fazer um print das respostas.

Grupo 2

Pergunta 2.1

A primeira heuristica consiste no calculo da menor distancia de manhattan do robo a um dos mosaicos sujos em cada estado. Sempre que um mosaico é limpo a distancia de manhattan é calculada para outro estado.
A segunda heuristica admissivel para este problema consiste no número de mosaicos por limpar em cada estado.
Para correr os predicados que utilizam as heuristicas anunciadas acima usam-se os comandos `exercicio_2_h1` e `exercicio_2_h2` para a primeira e segunda heuristica respetivamente.

Pergunta 2.2

Apesar de ambas as nossas heuristicas “rebentarem” com a global stack do prolog, acreditamos que o melhor algoritmo para resolver este problema é a pesquisa A*. Com esta pesquisa acrescentamos o valor acumulado dos custos das transições à heuristica de cada estado.
A implementação deste algoritmo encontra-se no ficheiro `lib_ficha3.pl`, que foi nos facultando durante as aulas.

2.2.1

Ambas as heuristicas rebentam a pilha global do prolog pelo que nao conseguimos calcular este valor.

2.2.2

Ambas as heuristicas rebentam a pilha global do prolog pelo que nao conseguimos calcular este valor.

Programas Usados

- `lib_ficha3.pl`: contem os algoritmos de pesquisa informada (ansiosa e A*).
- `aspirador_est.pl`: contem o estado inicial e o final do problema.
- `aspirador_h1.pl`: contem a implementação da heuristica com a distancia de manhattan.
- `aspirador_h2.pl`:contem a implementação da heuristica que calcula o numero de mosaicos sujos.
- `aspirador_op.pl`: contem todas as transições de estados do robo.