



UNIVERSIDADE DE ÉVORA

Relatório Inteligência Artificial

Rúben Peixoto & Vanessa Santos
37514 & 34191

6 de Abril 2020

Segundo Trabalho

Este programa foi criado em SWIProlog. Por isso, os ficheiros necessário para o seu funcionamento são: "problema.pl" e "online_dfs_agent.pl". O comando necessário para iniciar o programa no ambiente SWIProlog é:

swipl problema.pl online_dfs_agent.pl

Após entrar no top level, o comando para executar o programa é:

search.

O output do programa é dividido em três parte, uma que mostra o resultado como um conjunto de nós, vide figura a baixo:

```
Resulted Node:
'node([1,4],node([2,4],node([2,3],node([2,2],node([3,2],node([3,1],node([2,1],node([1,1],node([2,1],node([3,1],node([3,2],node([4,2],node([4,1],node([4,2],node([3,2],node([2,2],node([1,2],node([1,3],node([1,2],node([2,2],node([2,3],node([3,3],node([2,3],node([1,3],node([2,3],node([2,2],node([3,2],node([3,1],node([2,1],node([1,1],node([2,1],node([3,1],node([3,2],node([4,2],node([3,2],node([2,2],node([1,2],node([2,2],node([3,2],node([3,1],node([2,1],node([1,1],node([2,1],node([3,1],node([2,1],node([1,1],node([2,1],node([1,1],node([1,0],right,1),left,2),right,3),right,4),left,5),left,6),right,7),right,8),down,9),left,10),left,11),right,12),right,13),right,14),left,15),up,16),left,17),left,18),right,19),right,20),down,21),left,22),down,23),left,24),right,25),right,26),left,27),up,28),left,29),down,30),up,31),right,32),right,33),right,34),up,35),down,36),left,37),up,38),left,39),left,40),right,41),right,42),down,43),left,44),down,45),down,46),left,47)'
```

Figure 1: Resultado final representada por nós

os outros dois outputs serão apresentados nos sub-tópicos a seguir. Os próximos outputs têm como base a informação fornecida pelos nós.

Uma vez que o espaço de estados é desconhecido pelo agente, o grupo usou o algoritmo "Online Depth First Search Agent". Este algoritmo tem como finalidade fazer com que o agente dê um passo no espaço de estados. Após implementado este algoritmo, criamos um ciclo, que o vai chamando várias vezes, até chegar a uma solução.

No nosso caso, o programa chegou ao estado final tal como pretendido.

Caminho Seguido

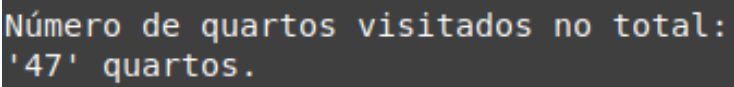
Na figura ao lado temos um esquema de quais as posições acedidas e quais as acções tomadas, sendo que, a primeira posição que se vê na imagem corresponde ao estado inicial e o ultimo ao estado final.

Por exemplo, na primeira linha, temos "[1,1] + right", isto significa que na posição [1,1] a acção que irá ser tomada é ir para a direita. A posição que origina dessa acção encontra-se sempre na linha de baixo, que neste caso será, "[2,1]".

```
Caminho seguido até chegar à saída:
-> [1,1] + right
-> [2,1] + left
-> [1,1] + right
-> [2,1] + right
-> [3,1] + left
-> [2,1] + left
-> [1,1] + right
-> [2,1] + right
-> [3,1] + down
-> [3,2] + left
-> [2,2] + left
-> [1,2] + right
-> [2,2] + right
-> [3,2] + right
-> [4,2] + left
-> [3,2] + up
-> [3,1] + left
-> [2,1] + left
-> [1,1] + right
-> [2,1] + right
-> [3,1] + down
-> [3,2] + left
-> [2,2] + down
-> [2,3] + left
-> [1,3] + right
-> [2,3] + right
-> [3,3] + left
-> [2,3] + up
-> [2,2] + left
-> [1,2] + down
-> [1,3] + up
-> [1,2] + right
-> [2,2] + right
-> [3,2] + right
-> [4,2] + up
-> [4,1] + down
-> [4,2] + left
-> [3,2] + up
-> [3,1] + left
-> [2,1] + left
-> [1,1] + right
-> [2,1] + right
-> [3,1] + down
-> [3,2] + left
-> [2,2] + down
-> [2,3] + down
-> [2,4] + left
-> [1,4]
```

Figure 2: Posição + acção tomada

Número de quartos visitados

A screenshot of a terminal window with a dark background. It displays two lines of text in a light-colored monospace font. The first line reads 'Número de quartos visitados no total:' and the second line reads ''47' quartos.'

```
Número de quartos visitados no total:  
'47' quartos.
```

Figure 3: Número de quartos visitados

No nosso caso, tal como mostra a figura a cima, o número total de quartos visitados até chegar a uma solução foram 47.