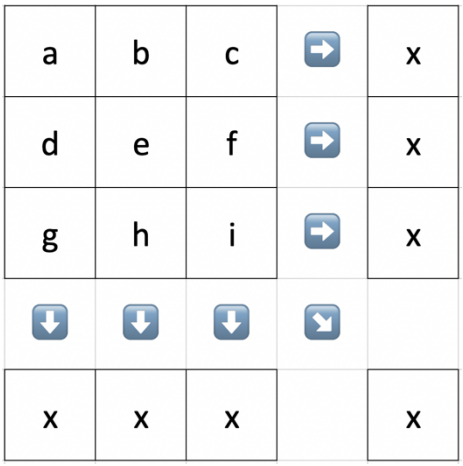
Universidade de Évora
Escola de Ciências e Tecnologias

Inteligência Artificial 1º Trabalho

Quadrado Mágico e Sudoku

(3x3)



	1				8		7	3
			5		9			
7						9		4
					4			
				3	5		1	8
8			9					
			7					
2	6			4			3	
		5			3			

Trabalho realizado por:
- Cláudia Dias nº35308
- João Queimado nº38176

Ano letivo 2018/2019

Índice

Introdução

Este trabalho consiste no desenvolvimento de um quadrado mágico e de um jogo de sudoku.

Um quadrado mágico é um jogo matemático que tem como regras:

- Todos os elementos são diferentes;
- A soma das linhas, das colunas e das duas diagonais principais são iguais.

O sudoku é um jogo matemático que exige lógica e raciocínio para se conseguir resolver. Tem um formato 9x9 (linhas x Colunas). As regras são:

- Todas as linhas têm que ter todos os números de 1 a 9, sem repetir nenhum;
- Todas as colunas têm que ter todos os números de 1 a 9, sem repetir nenhum;
- Todos os quadrados (3x3) têm que ter todos os números de 1 a 9, sem repetir nenhum.

Resposta às questões

Questão 1:

O estado inicial irá ser:

```
?- estado_inicial(E).
E = e([v(p(1,1),[1,2,3,4,5,6,7,8,9],_), v(p(2,1),[1,2,3,4,5,6,7,8,9],_),
v(p(3,1),[1,2,3,4,5,6,7,8,9],_), v(p(1,2),[1,2,3,4,5,6,7,8,9],_),
v(p(2,2),[1,2,3,4,5,6,7,8,9],_), v(p(3,2),[1,2,3,4,5,6,7,8,9],_),
v(p(1,3),[1,2,3,4,5,6,7,8,9],_), v(p(2,3),[1,2,3,4,5,6,7,8,9],_),
v(p(3,3),[1,2,3,4,5,6,7,8,9],_)]
```

O domínio é de 1 a 9, visto que a matriz é 3 por 3.

(3 x 3 = 9).

Para as restrições criámos a função ve_restricoes(E), que chama a função ver_tudo, ver_linhas, ver_colunas, ver_diagonal_dir e ver_diagonal_esq

```
/*** <Restrições> ***/

%Restricoes
ve_restricoes(E):-
    tamanho_tabuleiro(T),
    ver_tudo(E),
    ver_linhas(E,L1,T1),
    ver_colunas(E,L2,T2),
    ver_diagonal_dir(E,L3,T3),
    ver_diagonal_esq(E, L4, T4),

    conta(K),

    (=(T, T1)
        -> (=(K, L1)
            -> true
            ; fail)
        ; true),

    (=(T, T2)
        -> (=(K, L2)
            -> true
            ; fail)
        ; true),

    (=(T, T3)
        -> (=(K, L3)
            -> true
            ; fail)
        ; true),

    (=(T, T4)
        -> (=(K, L4)
            -> true
            ; fail)
        ; true).
```

A função sucessor indica que valor pode ser atribuído a qualquer variável desde que não entre em conflito com as restrições.

```
sucessor(e([v(N,D,V)|R],E),e(R,[v(N,D,V)|E])):- member(V,D).
```

Por exemplo, os resultados que encontrámos foram, numa matriz 3x3:

```
?- [pesquisaback].
true.

?- p(quadrado_magico).
2 . 9 . 4
7 . 5 . 3
6 . 1 . 8

true ;
8
2 . 7 . 6
9 . 5 . 1
4 . 3 . 8

true ;
8
4 . 9 . 2
3 . 5 . 7
8 . 1 . 6

true ;
6
4 . 3 . 8
9 . 5 . 1
2 . 7 . 6

true ;
6
6 . 7 . 2
1 . 5 . 9
8 . 3 . 4

true ;
4
6 . 1 . 8
7 . 5 . 3
2 . 9 . 4

true ;
4
8 . 3 . 4
1 . 5 . 9
6 . 7 . 2

true ;
2
8 . 1 . 6
3 . 5 . 7
4 . 9 . 2

true ;
2
false.

?-
```

Na matriz 4x4, o domínio é de 0 a 16, por exemplo:

```
?- [pesquisaback].
true.

?- p(quadrado_magico).
1 . 12 . 13 . 8
2 . 14 . 7 . 11
15 . 3 . 10 . 6
16 . 5 . 4 . 9

true ;
9
1 . 13 . 12 . 8
2 . 14 . 7 . 11
15 . 3 . 10 . 6
16 . 4 . 5 . 9

true ;
9

1 . 13 . 12 . 8
2 . 14 . 7 . 11
16 . 4 . 9 . 5
15 . 3 . 6 . 10

true .

?-
```

Questão 2:

O tabuleiro é 9x9, o domínio está entre os valores 0 e 9, visto que o sudoku é um quadrado 9x9, tem 9 quadrantes 3x3.

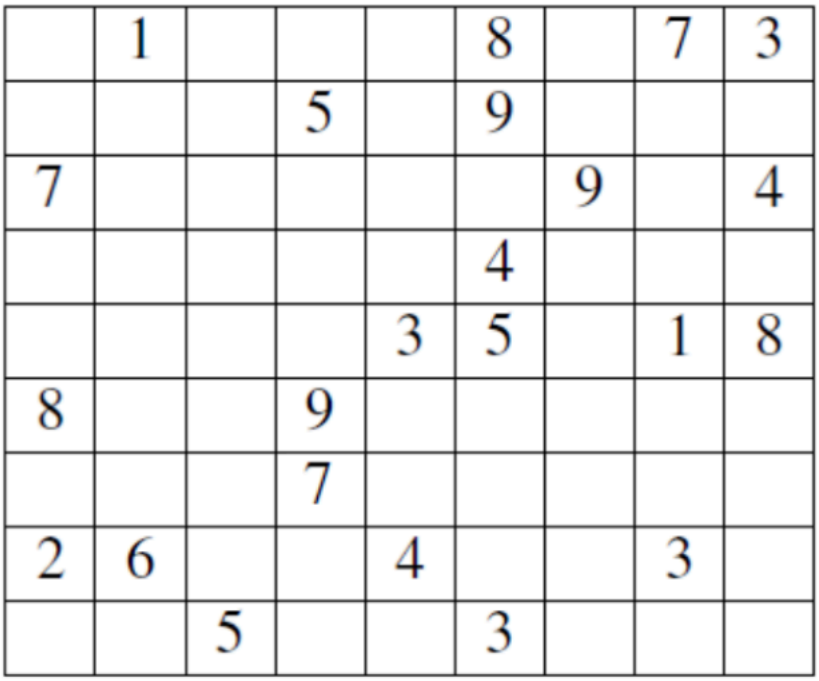
Para as restrições, baseámo-nos no problema anterior (quadrado mágico), sendo que retirámos a parte da soma das linhas e das colunas.

Fizémos uma restrição auxiliar (ver quadrantes) que verifica se todos os 9 quadrantes 3x3 têm valores diferentes entre eles.

```
ver_quad(L, X, Y, Y2, Q) :-
  Y = Y2, X1 is X+2,
  add_quad(L, X, Y, X1, Q).

ver_quad(L, X, Y, Y2, Q) :-
  Y < Y2, Y1 is Y+1,
  X1 is X+2,
  add_quad(L, X, Y, X1, L1),
  append(L1, L2, Q),
  ver_quad(L, X, Y1, Y2, L2).

add_quad(L, X, Y, X2, []) :-
  X = X2,
  \+member(v(p(X, Y), _, _), L).

add_quad(L, X, Y, X2, [V]) :-
  X = X2,
  member(v(p(X, Y), _, V), L).

add_quad(L, X, Y, X2, T) :-
  X < X2, X1 is X+1,
  \+member(v(p(X, Y), _, _), L),
  add_quad(L, X1, Y, X2, T).

add_quad(L, X, Y, X2, [V|T]) :-
  X < X2,
  member(v(p(X, Y), _, V), L),
  X1 is X+1,
  add_quad(L, X1, Y, X2, T).
```

```
/*** <Restrições> ***/

%Restricoes
ve_restricoes(E):-
  ver_linhas(E),
  ver_colunas(E),
  ver_quadrantes(E).

/*** verificação de linhas ***/

ver_linhas(e(_,[v(p(X,_), _, V)|R])):-
  findall(V1,member(v(p(X,_),_,V1),R),L),
   todos_diff([V|L]).

/*****************************/

/*** verificação de colunas ***/

ver_colunas(e(_,[v(p(_,Y), _, V)|R])):-
  findall(V1,member(v(p(_,Y),_,V1),R),L),
  todos_diff([V|L]).

/*******************************/

/*** verificação dos quadrantes ***/
ver_quadrantes(e(_,Afect)) :-
  ver_quad(Afect, 1, 1, 3, Q1),
  todos_diff(Q1),
  ver_quad(Afect, 1, 4, 6, Q2),
  todos_diff(Q2),
  ver_quad(Afect, 1, 7, 9, Q3),
  todos_diff(Q3),
  ver_quad(Afect, 4, 1, 3, Q4),
  todos_diff(Q4),
  ver_quad(Afect, 4, 4, 6, Q5),
  todos_diff(Q5),
  ver_quad(Afect, 4, 7, 9, Q6),
  todos_diff(Q6),
  ver_quad(Afect, 7, 1, 3, Q7),
  todos_diff(Q7),
  ver_quad(Afect, 7, 4, 6, Q8),
  todos_diff(Q8),
  ver_quad(Afect, 7, 7, 9, Q9),
  todos_diff(Q9).
```

Com a matriz:



	1				8		7	3
			5		9			
7						9		4
					4			
				3	5		1	8
8			9					
			7					
2	6			4			3	
		5			3			

O resultado é:

```
?- [pesquisaback].
true.

?- p(sudoku).
5 1 9 4 2 8 6 7 3
6 3 4 5 7 9 1 8 2
7 2 8 3 1 6 9 5 4
3 5 2 1 8 4 7 9 6
9 7 6 2 3 5 4 1 8
8 4 1 9 6 7 3 2 5
4 9 3 7 5 2 8 6 1
2 6 7 8 4 1 5 3 9
1 8 5 6 9 3 2 4 7
```

Conclusão

Com a realização deste trabalho aprendemos baste, principalmente como funcionam os algoritmos de pesquisa.

Conseguimos resolver o Quadrado Mágico para 3x3 e 4x4 rapidamente, sendo que io 4x4 demorou mais um pouco. No caso 5x5 demora muito tempo e por isso o algoritmo de backtracking não é aconselhado.

Resolvemos também o Sudoku, sendo apresentado o seu resultado rapidamente e sem erros.