Inteligência Artificial

Trabalho 2

42470: João Cavaco

42501: Gerson Abreu

43014: Nuno Sousa

2020/2021

Problema 1

a)

Domínio

```
%dominio
dominio(['Maria',
        'Manuel',
        'Madalena',
        'Joaquim',
        'Ana',
        'Julio',
        'Matilde',
        'Gabriel']).
```

As variáveis estão representadas através do número da cadeira *c(N)*, o domínio do problema e a pessoa que lá se encontra sentada.

v(c(nCadeira),dominio,pessoa)).

No estado inicial todas as cadeiras estão vazias e as pessoas estão em pé.

```
estado_inicial(e([
        v(c(1),D,_),
        v(c(2),D,_),
        v(c(3),D,_),
        v(c(4),D,_),
        v(c(5),D,_),
        v(c(6),D,_),
        v(c(7),D,_),
```

```
        v(c(8),D,_)], []):- dominio(D).
```

Restrições

```
restricoes([esq('Manuel','Maria'), frente('Joaquim','Maria'),
lado('Joaquim', 'Matilde'), cabeceira('Gabriel') ]).


%ve_restricoes(e(Nafect,[A])).
ve_restricoes(e(_Nafec,Afect)):-
    \+ (member(v(c(I),_Di,Vi), Afect),
    member(v(c(J),_Dj,Vj),Afect),
    I \= J,
    restric(I,Vi,J,Vj)).


%SUCEDE SE ALGUMA RESTRICAO FALHA
restric(I,X,J,Y):- restricoes(L), member(esq(X,Y),L),
                \+ (I is J+1; (I=1,J=8)).
restric(I,X,J,Y):- restricoes(L), member(dir(X,Y),L),
                \+ (I is J+1; (I=8,J=1)).
restric(I,X,J,Y):- restricoes(L), member(lado(X,Y),L),
                    \+ ((I is J+1; (I=8,J=1));(I is J+1; (I=1,J=8))).
restric(I,X,J,_):- restricoes(L), member(cabeceira(X),L),
                \+ (I=1,J=8).
```

operador sucessor

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

Problema 2

Sudoku representado como um problema de satisfação de problemas em prolog.

a)

Domínio

```
%n(n(x,y,quadrante),dominio,valor).
dominio([1,2,3,4,5,6,7,8,9]).
```

As variáveis estão representadas através da sua posição (x,y) em cada um dos 9 quadrantes que compõem um tabuleiro Sudoku e do seu valor.

v(n(X,Y,quadrante),dominior,valor).

Estado inicial que representa a imagem presente no enunciado. Neste estado inicial as casas já preenchidas encontram-se na segunda lista (Afetados).

```
estado_inicial(e(
        % Nao afetados
        [
        % quadrante 1
        v(n(1,1,1),D,_),v(n(1,3,1),D,_),
        v(n(2,1,1),D,_),v(n(2,2,1),D,_),
        v(n(2,3,1),D,_),v(n(3,2,1),D,_),
        v(n(3,3,1),D,_),
        % quadrante 2
        v(n(1,4,2),D,_),v(n(1,5,2),D,_),
        v(n(2,5,2),D,_),v(n(3,4,2),D,_),
        v(n(3,5,2),D,_),v(n(3,6,2),D,_),
```

```
% quadrante 3
v(n(1,7,3),D,_),v(n(2,7,3),D,_),
v(n(2,8,3),D,_),v(n(2,9,3),D,_),
v(n(3,8,3),D,_),
% quadrante 4
v(n(4,1,4),D,_),v(n(4,2,4),D,_),
v(n(4,3,4),D,_),v(n(5,1,4),D,_),
v(n(5,2,4),D,_),v(n(5,3,4),D,_),
v(n(6,2,4),D,_),v(n(6,3,4),D,_),
% quadrante 5
v(n(4,4,5),D,_),v(n(4,5,5),D,_),
v(n(5,4,5),D,_),v(n(6,5,5),D,_),
v(n(6,6,5),D,_),
% quadrante 6
v(n(4,7,6),D,_),v(n(4,8,6),D,_),
v(n(4,9,6),D,_),v(n(5,7,6),D,_),
v(n(6,7,6),D,_),v(n(6,8,6),D,_),
v(n(6,9,6),D,_),
% quadrante 7
v(n(7,1,7),D,_),v(n(7,2,7),D,_),
v(n(7,3,7),D,_),v(n(8,3,7),D,_),
v(n(9,1,7),D,_),v(n(9,2,7),D,_),
% quadrante 8
v(n(7,5,8),D,_),v(n(7,6,8),D,_),
v(n(8,4,8),D,_),v(n(8,6,8),D,_),
v(n(9,4,8),D,_),v(n(9,5,8),D,_),
% quadrante 9
v(n(7,7,9),D,_),v(n(7,8,9),D,_),
v(n(7,9,9),D,_),v(n(8,7,9),D,_),
v(n(8,9,9),D,_),v(n(9,7,9),D,_),
v(n(9,8,9),D,_),v(n(9,9,9),D,_)
],
```

```
    % Afetados
    [
    % quadrante 1
    v(n(1,2,1),D,1),v(n(3,1,1),D,7),
    % quadrante 2
    v(n(1,6,2),D,8),v(n(2,4,2),D,5),
    v(n(2,6,2),D,9),
    % quadrante 3
    v(n(1,8,3),D,7),v(n(1,9,3),D,3),
    v(n(3,7,3),D,9),v(n(3,9,3),D,4),
    % quadrante 4
    v(n(4,6,5),D,4),v(n(6,1,4),D,8),
    % quadrante 5
    v(n(5,5,5),D,3),v(n(5,6,5),D,5),
    v(n(6,4,5),D,9),
    % quadrante 6
    v(n(5,8,6),D,1),v(n(5,9,6),D,8),
    % quadrante 7
    v(n(7,4,8),D,7),v(n(8,1,7),D,2),
    v(n(8,2,7),D,6),v(n(9,3,7),D,5),
    % quadrante 8
    v(n(8,5,8),D,4),v(n(9,6,8),D,3),
    % quadrante 9
    v(n(8,8,9),D,3)
]):- dominio(D).
```

operador sucessor

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

Restrições

```
restricoes(e(_,[v(n(X,Y,Z), _, V)|Afect])):-
        findall(V3,member(v(n(_,_,Z),_,V3),Afect),L3),
        checkDifference([V|L3]),
        findall(V1,member(v(n(X,_,_),_,V1),Afect),L),
        checkDifference([V|L]),
        findall(V2,member(v(n(_,Y,_),_,V2),Afect),L2),
        checkDifference([V|L2]),
        L \= L2, L3\=L, L3\=L2.


checkDifference([]).
checkDifference([X|Afect]):-
        \+ member(X,Afect), checkDifference(Afect).
```

b)

Para resolver o problema com o algoritmo de backtracking basta consultar o ficheiro *pesquisas.pl* e recorrer ao comando *pb.*

```
| ?- pb.
compiling /home/jpbc/Documents/IA/trabalho2/ex2/sudoku.pl for byte code...
/home/jpbc/Documents/IA/trabalho2/ex2/sudoku.pl compiled, 100 lines read - 26412 bytes written, 59 ms
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 bastante acerca do funcionamento algoritmos de pesquisa. Infelizmente devido à dificuldade da utilização da linguagem Prolog, não conseguimos realizar todos os exercícios.