Inteligência Artificial

Trabalho 2

<u>Quadrado Mágico</u>

Docente: Irene Pimenta Rodrigues

Alunos: Rui Rodrigues nº14445

João Marques nº31514

João Chiola nº32456

Estado Inicial

```prolog
dominio([4,5,6,7,8,9]).
vazio([null]).

estado_inicial(e([ v(n(1,1),_,1),
                   v(n(1,2),['+'],_),
                   v(n(1,3),_,2),
                   v(n(1,4),['+'],_Op2),
                   v(n(1,5),_,3),
                   v(n(1,6),_,6),

                   v(n(2,1),['+','-','*','/'],_Op1),
                   v(n(2,2),X,_vazio),
                   v(n(2,3),['+','-','*','/'],_Op2),
                   v(n(2,4),X,_vazio),
                   v(n(2,5),['+','-','*','/'],_Op3),
                   v(n(2,6),X,_vazio),

                   v(n(3,1),D,_Val1),
                   v(n(3,2),['+','-','*','/'],_Op4),
                   v(n(3,3),D,_Val2),
                   v(n(3,4),['+','-','*','/'],_Op5),
                   v(n(3,5),D,_Val3),
                   v(n(3,6),_,15),

                   v(n(4,1),['+','-','*','/'],_Op1),
                   v(n(4,2),X,_vazio),
                   v(n(4,3),['+','-','*','/'],_Op2),
                   v(n(4,4),X,_vazio),
                   v(n(4,5),['+','-','*','/'],_Op3),
                   v(n(4,6),X,_vazio),

                   v(n(5,1),D,_Val1),
                   v(n(5,2),['+','-','*','/'],_Op4),
                   v(n(5,3),D,_Val2),
                   v(n(5,4),['+','-','*','/'],_Op5),
                   v(n(5,5),D,_Val3),
                   v(n(5,6),_,24),

                   v(n(6,1),_,12),
                   v(n(6,2),X,_vazio),
                   v(n(6,3),_,15),
                   v(n(6,4),X,_vazio),
                   v(n(6,5),_,18),
                   v(n(6,6),X,_vazio)],[])) :- dominio(D), vazio(X).
```

Testar linhas e colunas

```prolog
testar(Vi,ResL1,ResL2,ResL3,ResC1,ResC2,ResC3):-
    length(Vi,X),X<19.
testar(Vi,ResL1,ResL2,ResL3,ResC1,ResC2,ResC3):-

    %Primeira linha
    member(v(n(1,1),_,A),Vi),
    member(v(n(1,2),_,Op1),Vi),
    member(v(n(1,3),_,B),Vi),
    member(v(n(1,4),_,Op2),Vi),
    member(v(n(1,5),_,C),Vi),


    %Segunda linha
    member(v(n(3,1),_,D),Vi),
    member(v(n(3,2),_,Op3),Vi),
    member(v(n(3,3),_,E),Vi),
    member(v(n(3,4),_,Op4),Vi),
    member(v(n(3,5),_,F),Vi),


    %Terceira linha
    member(v(n(5,1),_,G),Vi),
    member(v(n(5,2),_,Op5),Vi),
    member(v(n(5,3),_,H),Vi),
    member(v(n(5,4),_,Op6),Vi),
    member(v(n(5,5),_,I),Vi),

    %Primeira coluna

    member(v(n(2,1),_,Op7),Vi),
    member(v(n(4,1),_,Op8),Vi),

    %Segunda coluna

    member(v(n(2,3),_,Op9),Vi),
    member(v(n(4,3),_,Op10),Vi),

    %Terceira coluna

    member(v(n(2,5),_,Op11),Vi),
    member(v(n(4,5),_,Op12),Vi),

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    teste(A,B,C,Op1,Op2,ResL1),
    teste(D,E,F,Op3,Op4,ResL3),
    teste(G,H,I,Op5,Op6,ResL5),

    teste(A,D,G,Op7,Op8,ResC1),
    teste(B,E,H,Op9,Op10,ResC2),
    teste(C,F,I,Op11,Op12,ResC3).
```

Backtracking

```prolog
p:- estado_inicial(E0), back(E0,A), esc(A).

back(e([],A),A).
back(E,Sol):- sucessor(E,E1), ve_restricoes(E1),
                    back(E1,Sol).

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