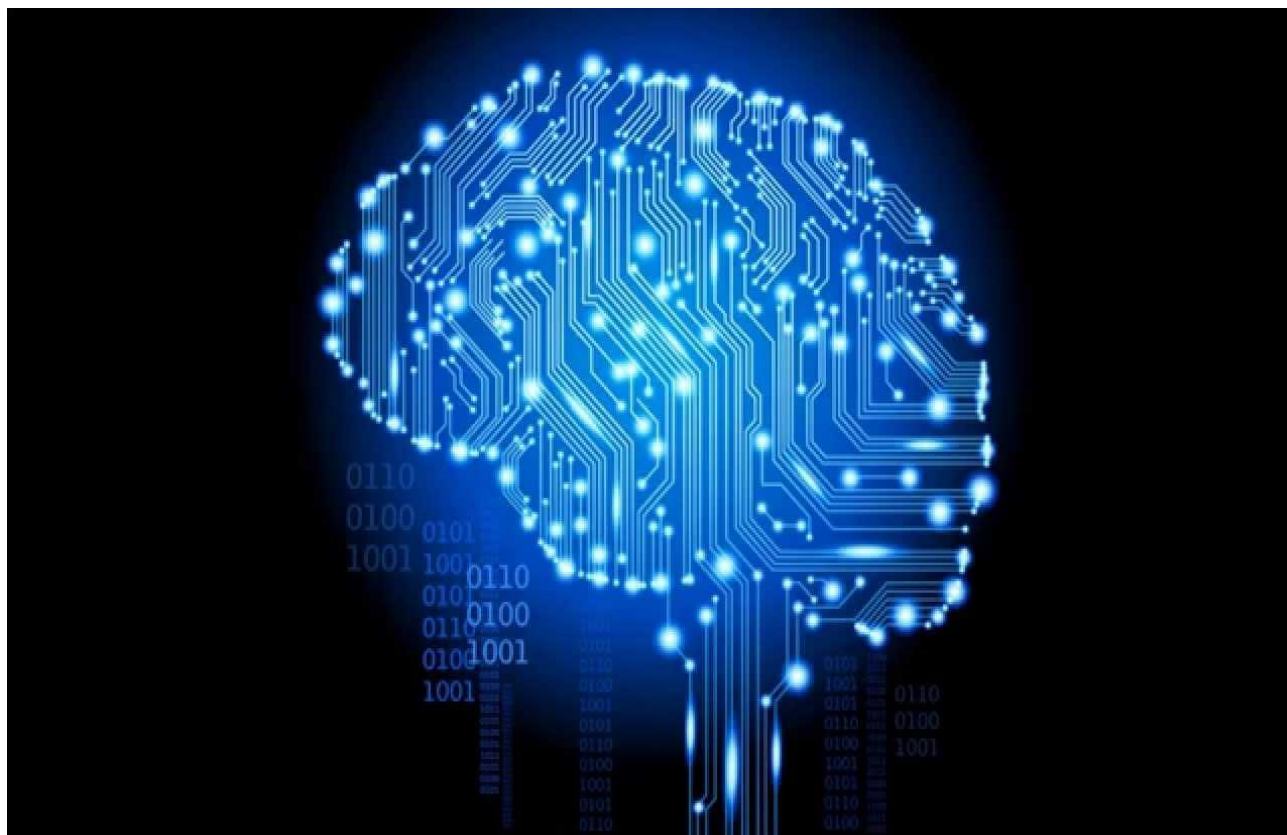


Inteligência Artificial



Trabalho realizado por:

Nuno Ribeiro 32471



1. Problema do quadrado mágico

Variaveis(Nome,dominio e valor) e dominio.

```
1  dimensao(3).
2
3  soma(S2):-
4      dimensao(R),
5      S1 is (R*((R*R)+1))/2,
6      S2 is round(S1).
7
8  dominio(S1):-
9      dimensao(R),
10     S2 is R*R,
11     dominioaux(S2,S1).
12
13  dominioaux(1,[1]):-!.
14
15  dominioaux(V,ListaV):-
16     V1 is V-1,
17     dominioaux(V1,ListaV2),
18     append(ListaV2,[V],ListaV).
19
20  matriz(1,[c(1,Y)],Y):-!.
21
22  matriz(N,ListaV,Y):-
23     N1 is N-1,
24     matriz(N1,ListaV2,Y),
25     append(ListaV2,[c(N,Y)],ListaV).
26     %N is < V
```

Estados

```
44
45  estado_inicial(e(S1,[])):-
46     dimensao(R),
47     colunamatriz(R,ListaC),
48     estado_inicial_aux(ListaC,S1).
49
50
51  estado_inicial_aux([],[]) :- !.
52  estado_inicial_aux([Head|Tail],[v(Head,D,_)|Tails]):-
53     dominio(D),
54     estado_inicial_aux(Tail,Tails).
55
56
```

Restrições

```
60  ve_restricoes(e(_,Afe)):-
61
62  findall( Valor, member(v(c(_,_),_,Valor),Afe),Listr),
63      sort(Listr,Lista),
64      length(Listr,Tlist),
65      length(Lista,Tlista),
66      dimensao(D),
67      ve_loop(D,Afe).
68
69
70  ve_loop(0,_):-!.
71  ve_loop(A,List):-
72      ve_coluna(A,List),
73      ve_linha(A,List),
74      Valor is A-1,
75      ve_loop(Valor,List)
76  .
```

```
79  ve_coluna(Vec,List):-
80      findall( Valor, member(v(c(Vec,_),_,Valor),List),
81          Listr), % vai a lista de v(c(x,y),D,V) e retira o v metendo na listr
82
83      sort(Listr,Lista), %retira duplicados e coloca-as em lista
84      length(Lista,Tlista), %tamanho da lista
85      length(Listr,Tlist), %comparar os tamanhos das listas
86      dimensao(D),
87      length(Listr,D),!,
88      ve_colunaaux(Lista,Nmagic),
89      soma(Nmagic).
90
91
92  ve_coluna(Vec,List):-
93      findall( Valor, member(v(c(Vec,_),_,Valor),List),
94          Listr), % vai a lista de v(c(x,y),D,V) e retira o v metendo na listr
95
96      dimensao(D),
97      \+length(Listr,D),!.
98
99
100  ve_colunaaux([],0):-
101      !.
102  ve_colunaaux([Head|Tail],T):- %somar os valores da lista
103      ve_colunaaux(Tail,Valor),
104      T is Head+Valor.
105
```

```

106 ve_linha(Vel,List):-
107     findall( Valor, member(v(c(_,Vel),_,Valor),List),
108         Listr), % vai a lista de v(c(x,y),D,V) e retira o v metendo na listr
109
110     sort(Listr,List), %retira duplicados e coloca-as em lista
111     length(Lista,Tlista), %tamanho da lista
112     length(Listr,Tlista), %comparar os tamanhos das listas
113     dimensao(D),
114     length(Listr,D),!,
115     ve_colunaaux(Lista,Nmagic),
116     soma(Nmagic) .
117
118 ve_linha(Vel,List):-
119     findall( Valor, member(v(c(_,Vel),_,Valor),List),
120         Listr), % vai a lista de v(c(x,y),D,V) e retira o v metendo na listr
121
122     dimensao(D),
123     \+length(Listr,D),!.
124
125

```

Sucessor

```

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

15 sucessor2(e([v(N,D,V)|R],E),e(NovoR,[v(N,D,V)|E])):- member(V,D),
16     altera_dominio(N,V,R,NovoR).
17

```

2 Sudoku

```

1  dimensao(9).
2
3  soma(S2):-
4      dimensao(R),
5      S2 is 9.
6
7  dominio(S1):-
8      dimensao(R),
9      S2 is 9,
10     dominioaux(S2,S1).
11
12     dominioaux(1,[1]):-!.
13
14     dominioaux(V,ListaV):-
15         V1 is V-1,
16         dominioaux(V1,ListaV2),
17         append(ListaV2,[V],ListaV).
18

```