Inteligência Artificial - Trabalho 1

Elementos

Número	Primeiro Nome + Apelido
35476	João Dias
35477	Eduardo Romão

Respostas

Grupo 1

Pergunta 1.1

Neste trabalho do sudoku, estamos perante um problema com restrições, isto deve-se ao facto de para completarmos um sudoku, termos de possuir numeros diferentes em cada linha, coluna e quadrante.
No nosso trabalho um estado consiste em duas listas. A primeira contem todas as variaveis não afetadas e a segunda contem todas as variavéis afetadas, ou seja quadrados prenchidos com numeros.
Em baixo podemos ver o estado inicial:

```
estado_inicial(e(NonAffect, Affect)) :-
    gerar_num(11, NonAffect1),                                  %
 inicializar as variaveis nao afetadas
```

```
    num_iniciais(e(NonAffect1, []), e(NonAffect, Affect)).  %
 inicializar os valores iniciais do sudoku
```

Dentro de cada lista estão as nossas variaveis que são representadas atraves da sua posição X, Y e quadrante, pelo seu dominio e pelo seu valor.

```
v(Dominio, pos(Quadrante,PosX,PosY), Valor)
```

Quanto as restrições, após termos uma lista com as variavéis em cada linha, coluna ou quadrante, chama-se o predicado `list_dif` para saber se existem valores repetidos.

```
ve_restricoes(e(_,L)):-
    ve_linha(L),
    ve_coluna(L),
    ve_quadrante(L).

ve_linha([v(D,pos(_,_,Y),V)|L]):-
    findall(V1, member(v(D,pos(_,_,Y),V1), L), Lista),      %
 colocar todos os valores de uma
    list_dif([V|Lista]).                                    %
 linha numa lista

ve_coluna([v(D,pos(_,X,_),V)|L]):-
    findall(V1, member(v(D,pos(_,X,_),V1), L), Lista),      %
 colocar todos os valores de uma
    list_dif([V|Lista]).                                    %
```

```
 coluna numa lista

ve_quadrante([v(D,pos(Qd,_,_),V)|L]):-
    findall(V1, member(v(D,pos(Qd,_,_),V1), L), Lista),     %
 colocar todos os valores de um
    list_dif([V|Lista]).                                    %
 quadrante numa lista

list_dif([]).
list_dif([H|R]):-
    \+ member(H,R),
    list_dif(R).
```

Pergunta 1.2

Para o estado inicial proposto é possivel utilizar o algoritmo backtracking para encontrar uma solução. Após correrrmos `p` em `backtrack.pl` é criado o tabuleiro inicial, ou seja o `estado_inicial`, e através da função recursiva `back` inserem-se valores no tabuleiro criado um novo estado ao qual se verifica se as restrições são cumpridas.

```
p:- estado_inicial(E0), back(E0,A), esc(A,1).
%
back(e([],A),A).
back(E,Sol):- sucessor(E,E1), ve_restricoes(E1),
                            back(E1,Sol).
%
sucessor(e([v(D,N,V)|R],E),e(R,[v(D,N,V)|E])):-
```

```
    member(V,D).
```

Pergunta 1.3

Para o algoritmo forward check foi criada uma função `forwarding` que apesar de não resolver o problema em questão, consiste em sempre que se executa um sucessor de um estado, ir retirar do dominio das variaveis nao afetadas da mesma linha, coluna e quadrante o valor que foi inserido nesse mesmo estado. Deste modo uma variavel nao afetada com um dominio mais reduzido vai encontrar um valor certo em menos passos.
Para reduzir o dominio das variaveis na mesma linha é chamada a função `de_linha`. Esta função coloca numa lista todas as variaveis nao afetadas da linha em que o novo valor foi inserido. Em seguida é retirado o valor de todos os dominios das variaveis nessa mesma lista. As variaveis nao afetadas são por ultimo atualizadas.

```
forwarding(v(_, pos(Q,X,Y), V), NonAffect, NonAffect1):-
    de_linha(Y, V, NonAffect, NonAffect1),
    de_coluna(X, V, NonAffect, NonAffect1),
    de_quadrante(Q, V, NonAffect, NonAffect1).
%
de_linha(_,_,[],_).
de_linha(Y, V, NonAffect, NonAffect1):-
    findall(v(D1,pos(Q1,X1,Y), V1), member(v(D1,pos(Q1,X1,Y),
V1), NonAffect), Lista),
    remove(V,Lista, Lista2),
    deleteAll(NonAffect,Lista, Lista3),
    append(Lista3, Lista2, NonAffect1).
```

```
%
de_coluna(_,_,[],_).
de_coluna(X, V, NonAffect, NonAffect1):-
    findall(v(D1,pos(Q1,X,Y1), V1), member(v(D1,pos(Q1,X,Y1),
V1), NonAffect), Lista),
    remove(V,Lista, Lista2),
    deleteAll(NonAffect,Lista, Lista3),
    append(Lista3, Lista2, NonAffect1).
%
de_quadrante(_,_,[],_).
de_quadrante(Q, V, NonAffect, NonAffect1):-
    findall(v(D1,pos(Q,X1,Y1), V1), member(v(D1,pos(Q,X1,Y1),
V1), NonAffect), Lista),
    remove(V,Lista, Lista2),
    deleteAll(NonAffect,Lista, Lista3),
    append(Lista3, Lista2, NonAffect1).



deleteAll(A,[], A).
deleteAll(Tail, [L2|R],T):-
    delete(Tail, L2, Result),
    deleteAll(Result, R, T).

remove(_,[],_).
remove(V,[v(D, P,V1)|Lista], [v(D1,P,V1)|Lista2]):-
    delete(D, V, D1),
    remove(V, Lista, Lista2).
```

Pergunta 1.4

Para executar novos sudokus basta alterar as inserções inicias no predicado `num_iniciais`

Programas Usados

- `sudoku.pl`: construir o tabuleiro do sudoku, verificar restrições e executar operações para forward checking
- `execucao.pl`: executar algoritmo backtrack, algoritmo forward check e imprimir resultado do problema.