Inteligência Artificial

2016/2017

UNIVERSIDADE DE ÉVORA

Resolução de problemas de Satisfação de Restrições

Docentes:

Hiago Oliveira 29248

Gil Catarino 32378

Luís Sousa 32095

Discente:

Irene Rodrigues

Introdução

Neste segundo trabalho pretende-se ganhar conhecimento sobre a pesquisa backtrack para ajudar a resolver um problema sobre o cubo mágico.

Representação do problema

1.

a)

Para resolver este problema começa-se por representar o estado inicial, em que cada variável representa um número ou operador, cada uma dela tem um nome, domínio e valor instanciado, que inicialmente nenhuma se encontra instanciada.

```
estado_inicial(
    e([v(n(1),[1,2,3,4,5,6,7,8,9],_),
       v(n(2),[1,2,3,4,5,6,7,8,9],_),
       v(n(3),[1,2,3,4,5,6,7,8,9],_),
       v(o(1),['+','-','*','/'],_),
       v(o(2),['+','-','*','/'],_),
       v(n(4),[1,2,3,4,5,6,7,8,9],_),
       v(n(5),[1,2,3,4,5,6,7,8,9],_),
       v(n(6),[1,2,3,4,5,6,7,8,9],_),
       v(o(3),['+','-','*','/'],_),
       v(o(4),['+','-','*','/'],_),
       v(n(7),[1,2,3,4,5,6,7,8,9],_),
       v(n(8),[1,2,3,4,5,6,7,8,9],_),
       v(n(9),[1,2,3,4,5,6,7,8,9],_),
       v(o(5),['+','-','*','/'],_),
       v(o(6),['+','-','*','/'],_),
       v(o(7),['+','-','*','/'],_),
       v(o(8),['+','-','*','/'],_),
       v(o(9),['+','-','*','/'],_),
       v(o(10),['+','-','*','/'],_),
       v(o(11),['+','-','*','/'],_),
       v(o(12),['+','-','*','/'],_)],[])).
```

As variáveis referentes aos números têm domínio entre 1 e 9 e os operadores as 4 operações elementares.

De seguida são definidos os predicados referentes a todas as operações disponíveis, que têm como parâmetros 3 números e 2 operadores, como na seguinte imagem:

```
op(A, '+', B, '+', C, V) :-
    V is A + B + C.
op(A, '+', B, '-', C, V) :-
    V is A + B - C.
op(A, '+', B, '*', C, V) :-
    V is A + B * C.
op(A, '+', B, '/', C, V) :-
    V is A + B / C.
```

A abordagem às restrições passa por obter os valores das variáveis instanciadas e verificar que todas têm um valor diferente. Foi criado o predicado restr_todas_dif/1 que recebe a lista de variáveis instanciadas e sucede caso todos os seus valores sejam diferentes.

```
restr_todas_dif(L) :- restr_todas_dif(L, []).
restr_todas_dif([], L) :-
    %%all_distinct(L).
    fd_all_different(L).
restr_todas_dif([H|T], L) :-
    arg(1, H, n(_)),
    arg(3, H, V),
    restr_todas_dif(T, [V|L]).
restr_todas_dif([H|T], L) :-
    arg(1, H, o(_)),
    restr_todas_dif(T, L).
```

De seguida são verificadas as restrições para cada linha linha/coluna. O programa começa por verificar as restrições da primeira linha, sucedendo no caso óbvio ou quando ainda não há variáveis instanciadas suficientes para verificar a restrição. As restrições são depois verificadas linha a linha, e quando todas as variáveis referentes aos 9 números estão instanciadas pode-se começar a verificar colunas também.

```
restr_seg_linha(Inst) :-
    restr_todas_dif(Inst),
    member(v(n(4),_,Vn4),Inst),
    member(v(n(5),_,Vn5),Inst),
    member(v(n(6),_,Vn6),Inst),
    member(v(o(3),_,Vo3),Inst),
    member(v(o(4),_,Vo4),Inst), !,
    op(Vn4, Vo3, Vn5, Vo4, Vn6, 15),
    restr_ter_linha(Inst).
restr_seg_linha(Inst) :-
    restr_todas_dif(Inst).
```

O operador sucessor é definido da mesma maneira que no exercício das rainhas, ou seja

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

b)

```
true ? ;
5 - 3 + 4
+   *   +
1 * 8 + 7
+   -   *
6 + 9 * 2

true ? ;
5 + 7 - 6
*   *   +
4 + 2 + 9
-   +   +
8 * 1 * 3

true ? ;
5 - 7 + 8
+   +   +
3 * 2 + 9
+   +   +
4 * 6 * 1

true ? ;
7 + 5 - 6
+   +   +
2 + 9 + 4
+   +   +
3 * 1 * 8

true ?
```

c)

A introdução de forward check no problema causou alguns problemas. A primeira abordagem fez com que o programa demorasse mais tempo a encontrar uma solução do que sem o forward_checking. Na tentativa de melhorar a maneira como o forward_checking é feito, o predicado foi alterado e faz o que é suposto, remover do domínio das variáveis não instanciadas o valor das instanciadas, e isto pode ser verificado fazendo trace na execução. Porém quando é feito o backtrack para encontrar uma nova solução o programa faz redo do forward_check, que não é o que se pretende e não se conseguiu anular este comportamento, mesmo com introdução de cuts (!/0) ou predicados que não permitem backtrack (memberchk/2 em vez de member/2).