

2 Trabalho de Inteligência Artificial

Problemas de satisfação de restrições

Carlos Palma(46520)

1. Quadrado Mágico

1.1) Estados, Variáveis, Domínio:

- No desenvolvimento deste problema decidi representar os estados como $e(L1, L2)$ sendo $L1$ correspondente à lista de variáveis não instanciadas e $L2$ a lista de variáveis instanciadas. As variáveis seguem a estrutura $v(p(X, Y), D, V)$ em que X e Y são respetivamente o numero da linha e coluna, D o domínio correspondente (dependendo do tamanho do problema estes valores diferem) e V o valor quando instanciado.

1.2) Restrições:

- A nível das restrições criei o predicado `ve_restricoes` que verifica se a soma dos elementos de uma linha/coluna/diagonal_principal/diagonal_secundária são iguais ao valor da constante mágica (valor da constante depende da dimensão do problema) e que ao mesmo tempo também verifica se todos estes valores são diferentes.

1.3) Estado inicial e operador sucessor:

- O estado inicial terá o numero de variáveis consoante o tamanho do quadrado, sendo o numero de variáveis igual ao numero de quadrados ($n_quadrados = \text{tamanho de tabuleiro} * \text{tamanho de tabuleiro}$), em relação ao domínio será de 1 a n , sendo n o numero de quadrados.

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

1.4) Exemplos:

1.4.1) Tamanho do tabuleiro = 3 com backtracking

4	.	8	.	3
9	.	1	.	5
2	.	6	.	7

1.4.2) Tamanho do tabuleiro = 3 com forward check

4	.	8	.	3
9	.	1	.	5
2	.	6	.	7

2. Sudoku

2.1) Estados, Variáveis, Domínio:

- No desenvolvimento deste problema decidi representar os estados como $e(L1, L2)$ sendo $L1$ correspondente à lista de variáveis não instanciadas e $L2$ a lista de variáveis instanciadas. As variáveis seguem a estrutura $v(p(X, Y), D, V)$ em que X e Y são respetivamente o número da linha e coluna, D o domínio correspondente (dependendo do tamanho do problema estes valores diferem) e V o valor quando instanciado.

2.2) Restrições:

- A nível das restrições criei o predicado `ve_restricoes` que verifica se os elementos de uma linha/coluna/quadrantes são todos diferentes, verifica também quais os valores do domínio já presentes em cada quadrante e quais podem ser colocados tendo em conta as restrições.

2.3) Estado inicial e operador sucessor:

- O estado inicial terá o número de variáveis consoante o tamanho do quadrado, sendo o número de variáveis igual ao número de quadrados ($n_quadrados = \text{tamanho de tabuleiro} * \text{tamanho de tabuleiro}$), em relação ao domínio será de 1 a n , sendo n o número de quadrados.

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

2.4) Exemplos:

2.4.1) Tamanho do tabuleiro = 9 com backtracking

6	.	2	.	9	.	4	.	1	.	8	.	5	.	7	.	3
3	.	4	.	5	.	6	.	7	.	9	.	1	.	8	.	2
7	.	1	.	8	.	3	.	5	.	2	.	9	.	6	.	4
5	.	3	.	2	.	1	.	8	.	4	.	7	.	9	.	6
9	.	7	.	4	.	5	.	3	.	6	.	2	.	1	.	8
8	.	6	.	1	.	9	.	2	.	7	.	3	.	4	.	5
4	.	9	.	3	.	7	.	6	.	5	.	8	.	2	.	1
2	.	5	.	7	.	8	.	4	.	1	.	6	.	3	.	9
1	.	8	.	6	.	2	.	9	.	3	.	4	.	5	.	7

2.4.2) Tamanho do tabuleiro = 9 com forwardchecking

-Global stack overflow, o tamanho do problema é demasiado grande para o tamanho da stack.

2.4.3) Tamanho do tabuleiro = 9 com forwardchecking e backtracking

6	.	2	.	9	.	4	.	1	.	8	.	5	.	7	.	3
3	.	4	.	5	.	6	.	7	.	9	.	1	.	8	.	2
7	.	1	.	8	.	3	.	5	.	2	.	9	.	6	.	4
5	.	3	.	2	.	1	.	8	.	4	.	7	.	9	.	6
9	.	7	.	4	.	5	.	3	.	6	.	2	.	1	.	8
8	.	6	.	1	.	9	.	2	.	7	.	3	.	4	.	5
4	.	9	.	3	.	7	.	6	.	5	.	8	.	2	.	1
2	.	5	.	7	.	8	.	4	.	1	.	6	.	3	.	9
1	.	8	.	6	.	2	.	9	.	3	.	4	.	5	.	7