



UNIVERSIDADE DE ÉVORA

2º Trabalho

Inteligência Artificial

Professora: Irene Rodrigues

Realizado por: Miguel Menúria (43566) e Gonçalo Correia (43735)

26 de março de 2022

1

a) O espaço de estados e os operadores de transição de estados encontram-se no ficheiro `sudoku.pl`.

Para representar os vários estados resolvemos ter as variáveis com tuplos de 3 números (X,Y,Z). X e Y representam as coordenadas do quadrado no sudoku e Z representa o quadrante em que este está inserido (sendo que o tabuleiro possui 9 quadrantes (3x3)). O domínio vai de 1 a 9, uma vez que temos 9 quadrantes. Para o valor, apenas referimos o número relativo ao mesmo se já estiver inicialmente no tabuleiro, ou "_", caso não esteja.

Em relação às restrições, decidimos através do uso de `findalls`, encontrar todos os valores existentes para todas as posições com um X comum (na mesma linha), e utilizando um `all_diff` verificar se todos os valores encontrados são diferentes. O mesmo processo foi repetido para as variáveis Y e Z.

O predicado `sucessor` foi escrito da seguinte forma:

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

b) Utilizar os seguintes comandos: `'[sudoku, backtracking].'` e em seguida `'p.'`.

c) Utilizar os seguintes comandos: `'[sudoku, forwardchecking].'` e em seguida `'p.'`.

Por alguma razão, a nossa implementação de forward checking não conseguiu chegar a uma solução.

d) Modificando por exemplo, o código do sudoku. Através do uso de `findalls` para todos os valores das linhas da tabela pode ser pouco eficiente a nível de complexidade espacial, o que leva a uma procura constante entre os valores, o que irá aumentar também a complexidade temporal.

Exemplo de um output:

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