Resolução de problemas como problemas satisfação de restrições

João Matos (32409), João Rouxinol (44451) e André Rato (45517)

14 de maio de 2022

1 Sudoku como um CSP

1.1 O problema

1.1.1 Estados

Para a representação do problema, os estados seguem todos a estrutura "`e(LNI, LI)`", em que LNI é a lista de variáveis não instanciadas e LI é a lista de variáveis instanciadas.

1.1.2 Variáveis

As variáveis apresentam a estrutura "`v(p(X, Y), D, V)`", onde X e Y são as coordenadas da variável, D o domínio e V o valor.

1.1.3 Restrições

O predicado responsável por verificar as restrições é o predicado `ve_restricoes`, que verifica as linhas, as colunas e os quadrantes, de modo a que todos os elementos destes três grupos sejam diferentes.

1.1.4 Estado inicial

O estado inicial segue a estrutura dos estados do problema, em que o primeiro elemento do predicado é composto por uma lista com todas as variáveis vazias, e o segundo é composto por uma lista com todas as variáveis já preenchidas (as já instanciadas).

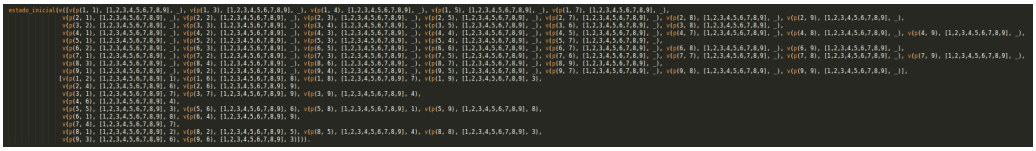

Figura 1: Estado inicial

1.1.5 Operador sucessor

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

Figura 2: Operador sucessor

1.2 Resolução com algoritmo de backtracking

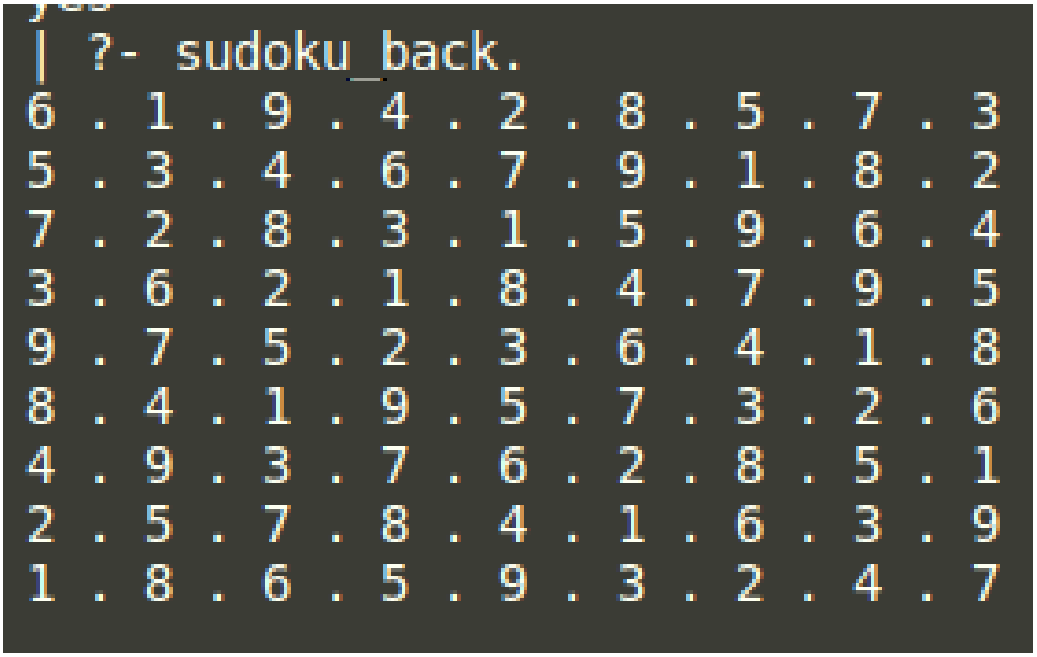

Figura 3: Estado final do tabuleiro após utilização do algoritmo de backtracking

1.3 Resolução com algoritmo de forward checking

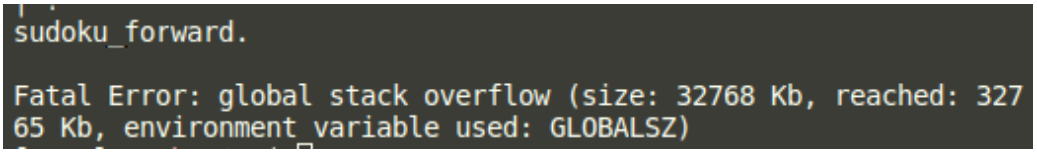

Figura 4: Output após utilização do algoritmo de forward checking

Isto acontece pois é excedida a memória disponibilizada para a execução do PROLOG.