Assignment 2

Rules:

- a number will appear once in each column.
- a number will appear once in each row.
- A number will appear once in each 3×3 small grid.

a. Give the representation of a solution (answer) of the problem, as explained during

the course. (1)

The answer to the problem will be given my a matrix M(9×9), where 1≤Mij≤9,

1≤i,j≤9, representing filled sudoku grid that satisfies the rules from above

So given an incomplete grid with some empty cells as input, the algorithm will fit and place numbers in those cells in a way that will satisfy the rules of Sudoku and return a filled matrix.

b. Give the equation for the restriction(s) of the problem. (0.5)

Let U = {1, 2, 3, 4, 5, 6, 7, 8, 9}

For each cell:

$$\forall i \in U, \quad \forall j \in U, \quad 1 <= M[i][j] <= 9$$

For each row:

$$\forall i \in U, \quad \forall j \in U, \quad \mathrm{M}[i][j] \neq \mathrm{M}[i][j'] \text{ where } j' \neq j$$

For each column:

$$\forall i \in U, \quad \forall j \in U, \quad \mathrm{M}[i][j] \neq \mathrm{M}[i'][j] \text{ where } i' \neq i$$

For each 3×3 grid:

Let Z = {1, 2 , 3}

- For each *k* representing the index of the 3×3 sub-grid (where *k* ranges from 1 to 3).
- For each *i* and *j* representing the indices of the cells within the sub-grid (where *i* and *j* range from 3*k*−2 to 3*k*).
- For each *i'* and *j'* representing the indices of the cells within the 3×3 sub-grid defined by the top-left corner (*a*,*b*) (where *a* and *b* are the top-left corner indices of each 3×3 sub-grid).
- The cell (*i,j*) should not be equal to any other cell (*i',j'*) within the same 3×3 sub-grid.

$$\begin{gathered}\forall k \in Z, \quad \forall i \in \{3k-2, 3k-1, 3k\}, \\ \forall j \in \{3k-2, 3k-1, 3k\}, \\ \forall (i', j') \in \{3a-2, 3a-1, 3a\} \times \{3b-2, 3b-1, 3b\}, \\ (i, j) \neq (i', j')\end{gathered}$$

c. What is consider as a state? In addition, explain why. (0.5)

In Sudoku, a state is the current snapshot of the grid, whether it’s the original input grid, or a grid with some of the empty cells filled and some empty, or a full grid.

It is used to perform constraint checks and to backtrack from a wrong cell choice.

d. Which is the initial state? In addition, explain why. (0.25)

The initial state is the input to the algorithm, which is a grid with some of the cells filled with numbers 1-9 and some left empty. It is the basis point from which the algorithm starts trying and placing numbers according to the rules.

e. Which is/are the possible action(s)? In addition, explain why. (0.25)

Possible actions are to place a number 1-9 into any of the empty cells, with a following validity check. In case the choice fails to satisfy the constraints, another possible action is backtracking to a previous state, except the initial state.

f. What is the maximum branching factor of the tree (b)? In addition, explain why. (0.25)

Maximum branching factor of the tree b = 9, since the highest complexity scenario would be an empty grid in which case the algorithm would try to find a solution for each cell. However, normally Sudoku puzzles are not empty so the factor is less.

g. What is the maximum depth of the search tree (m)? In addition, explain why. (0.25)

m = # of empty cells in the input grid

As every empty cell becomes a point of computation, the max depth of the tree is determined by the number of empty cells in the original grid.
In the worst case, if the grid is completely empty, m would equal 9*9 = 81