Assignment2: Backtracking algorithm on Sudoku problem

Aleksandra Ristic
arc19002@student.mdh.se

a. Representation

Solution of the Sudoku problem:

$$x = \begin{pmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,j} & \cdots & x_{m,m} \end{pmatrix}$$

Where $x_{i,j}$ represents the i-th row and j-th column of sudoku board and m=9.
Domain of each $x_{i,j}$ is $x_{i,j} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

b. Restrictions

The problem has the following constraints:
1) $x_{i,j} \neq x_{i,k}, \forall k \in \{1, 2, ..., 9\} \cup j \neq k$ (In every row can be only one element from domain)
2) $x_{i,j} \neq x_{k,j}, \forall k \in \{1, 2, ..., 9\} \cup i \neq k$ (In every column can be only one element from domain)
3) $x_{i,j} \neq x_{p,k}, \forall p \in \{\lfloor \frac{i-1}{3} \rfloor \cdot 3 + 1, \lfloor \frac{i-1}{3} \rfloor \cdot 3 + 2, \lfloor \frac{i-1}{3} \rfloor \cdot 3 + 3\} \cup \forall k \in \{\lfloor \frac{j-1}{3} \rfloor \cdot 3 + 1, \lfloor \frac{j-1}{3} \rfloor \cdot 3 + 2, \lfloor \frac{j-1}{3} \rfloor \cdot 3 + 3\} \cup (i \neq p \cap j \neq k)$ (In every small grid can be only one element from domain)

c. States

State - any board which fields are numbers 0-9 or blank)
With state we consider the current view of the Sudoku matrix which means that is one node in the backtracking algorithm. Changing one element of the matrix we will get a new state, in that way we are trying to find solution of the problem.

d. Initial State

Initial state - board contains blank fields
The initial state of this problem is a matrix with partially filled fields which is given like

task of the problem and our goal is to fill it so that all restrictions are respected. In initial state random fields contain a value from domain and other are empty. While, final state doesn't contain empty fields.

e. Actions

Actions - adding number on empty field
In order to find a solution we are performing an action. Choosing which number will be added on the board is doing with the help of the Backtracking algorithm. In this way we are changing the current state.

f. Branch factor

The maximum branching factor of the tree can be 9. Imagine if we have an initial board in which first row and first column are empty and our first 3x3 small grid is also empty, that means that our element in position $x_{0,0}$ can get all values from the domain, and there exist 9 values.
The branching factor depends on the number of possible actions.

g. Depth factor

The maximum depth of the search tree can be 81. The reason for this is that this is the number of fields in the sudoku board, and our worst situation can be if all fields of the board are empty. Theoretically this situation can happen, but practically not.
The depth factor depends on the number of empty fields.