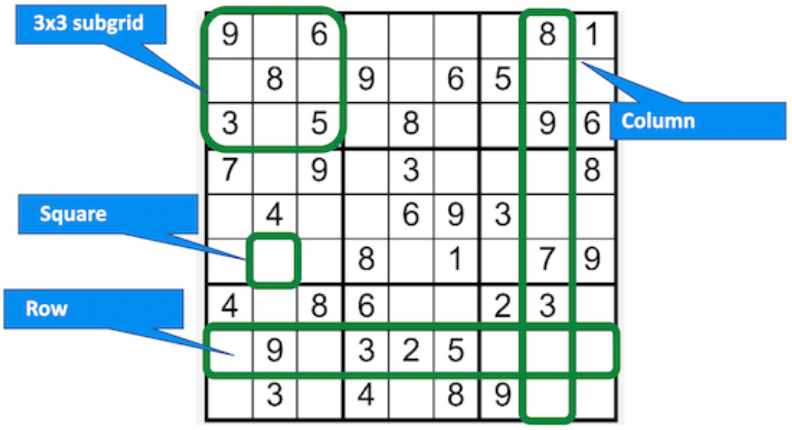
Sudoku

GitHub repo: https://github.com/pchs20/sudoku-mp

Let us define the following Sudoku **grid**, with the names of ***row***, ***column***, ***subgrid*** and ***square***:



Sudoku structure [Extracted from here]

Parameters

$n \in \mathbb{N}$	Defines the size of the problem. A typical Sudoku (with a grid of 9x9) would have an n of 3, as 3^2 equals 9. Strictly, then, n represents the number of subgrids per dimension of the Sudoku.

Sets

$R^g = \{1, ..., n^2\}$	Set of rows of the grid.
$C^g = \{1, ..., n^2\}$	Set of columns of the grid.
$R^{sg} = \{1, ..., n\}$	Group of rows taking into account the subgrids.
$C^{sg} = \{1, ..., n\}$	Group of columns taking into account the subgrids.
$K = \{1, 2, ..., n^2\}$	Allowed values in each square of the grid.

Variables

$x_{r^g, c^g, k} \in \{0, 1\}$	Equals 1 if a value k is placed in the square ($r^g\ x\ c^g$), equals 0 otherwise.	$\forall r^g \in R^g$, $\forall c^g \in C^g$, $\forall k \in K$

Constraints

In a row, each value must be used exactly once

$$\sum_{c^g \in C^g} (x_{r^g, c^g, k}) = 1,\ \forall r^g \in R^g,\ \forall k \in K$$

In a column, each value must be used exactly once

$$\sum_{r^g \in R^g} (x_{r^g, c, k}) = 1,\ \forall c^g \in C^g,\ \forall k \in K$$

In a subgrid, each value must be used exactly once

$$\sum_{i = r^{sg} \cdot n + 1}^{(r^{sg} + 1) \cdot n} \sum_{j = c^{sg} \cdot n + 1}^{(c^{sg} + 1) \cdot n} (x_{i, j, k}) = 1,\ \forall r^{sg} \in R^{sg},\ \forall c^{sg} \in C^{sg},\ \forall k \in K$$

No empty squares (same as: one value per square)

No position of the grid is empty.

$$\sum_{k \in K} (x_{c^g, r^g, k}) = 1,\ \forall c^g \in C^g,\ \forall r^g \in R^g$$

This constraint is no needed, as it is redundant given the first 2 constraints:

- The first constraint states that each row must contain each number exactly once. This ensures that each row is filled with any of the values inside K.
- For the second constraint, same conclusion for columns.

Therefore, since every row and column are completely filled, there cannot be empty squares. Thus, the combination of row and column constraints guarantees that no position in the grid is empty.

If we add this constraint the model will delete it in the **presolve phase**.

Objective

There is no function to maximize or minimize, as one solution is no “better” than another. Each solution that fulfills all the constraints is equally valid and optimal.

0