LAB 4 : SUDOKU SOLVER (PYCOSAT)

Introduction to Artificial Intelligence (DS2020)

1. **Name: DHRUVADEEP MALAKAR**
   Roll : 142201026
2. **Name: DODDAPANENI UDITH**
   Roll : 142201012

1. **CONVERT TO MATRIX**

   The **conv_to_matrix** function in the provided code is responsible for converting a string representation of a Sudoku puzzle into a matrix format that can be processed by the solver. It takes the input which inturn converts it to sqrt of the length and makes a **n*n** matrix filled with all 0's, after which we iterate through the string and get the row and column for the pre included numbers from the string and update the value there.

2. **We use 'PYCOSAT' to solve Sudoku using Boolean satisfiability problem (SAT solver). These are the components that we create and use to solve the problem.**

   Note: C_ijd represents cell at i-th row, j-th column, and d digit (1 - n)

   a. **Preconditioning**

      The precond function prepares the initial state of the Sudoku board as a set of clauses in conjunction normal form (CNF), which is standard format for Boolean expressions.
      It iterates throught each clause of each cell of the Sudoku board, If the cell s already filled (ie. not equal to 0), it adds a clause indicating that the corresponding literal should be **True**.

   b. **Single Position Check**

      It checks that every position aka. Cell should have a number and each positon must not have multiple numbers. This can be given as the following.

      1. (C_ij1 **V** C_ij2 **V** C_ij3 …….)
      2. For all (~C_ijd **V ~**C_ijd') for **d != d'**

c. Single Row Check

The "single row check" function in the Sudoku solver is responsible for generating constraints to ensure that each row in the Sudoku grid contains exactly one occurrence of each number from 1 to nn, where nn is the size of the Sudoku grid (typically 9 for a standard 9x9 Sudoku puzzle)

1. (C_i1d **V** C_i2d **V** C_i3d ……..)
2. (~C_ixk **V** ~C_iyk)

d. Single Column Check

The "single column check" function in the Sudoku solver is responsible for generating constraints to ensure that each column in the Sudoku grid contains exactly one occurrence of each number from 1 to nn, where nn is the size of the Sudoku grid (typically 9 for a standard 9x9 Sudoku puzzle)

1. (C_1jd **V** C_2jd **V** C_3jd ……..)
2. (~C_xjk **V** ~C_yjk)

d. Block Check

The "Block check" function in the Sudoku solver is responsible for generating constraints to ensure that each block in the Sudoku grid contains exactly one occurrence of each number from 1 to n*n, where nn is the size of the Sudoku grid (typically 9 for a standard 9x9 Sudoku puzzle)

Here in case of going from **1 to n** , we go from 1 to sqrt(n) from both row and column and also check for where there is no repeating numbers in the block. It checks for all the possible n blocks in the grid.

3. TOTAL CHECK

It will include all the possible and necessary conditions into a singular statement, and allow the **pycosat** to solve using the **solve(statement)** function. If its correct it will return the all literals and their boolean values (positive means true and negative means false)

4. CONVERT TO OUTPUT

The output is converted into a single line of string and appended to the file named **output.txt**