

Name: Kumar Gourav Songara
roll Number: 170353

Name: Saurav kumar
roll Number: 200906

how to use files and zip content

- sudoku_solver.py
 - from tests folder copy content to 'test.csv'
 - run sudoku_solver.py in python3
 - enter the value for k
 - the solution will be output in the terminal
- sudokuGenerator.py
 - run sudokuGenerator.py in python3
 - enter the value for k
 - the output will be generated in csv format in a new file 'output.csv'
- tests folder:
 - contains test cases csv files

Assumption

1. assume k is the order of the sudoku then $n=k^2$ variables will be required for each cell
2. assume number of rows and columns in each sudoku as $n=k*k$ a r = row's index and c = column index and v is the value assumed at that cell
3. hence $x_{r,c,v}$ is a literal for value v on r th row and c th column for first sudoku and $y_{r,c,v}$ similarly for the second sudoku.
4. these two sudoku grids has a solution if and only if the formula is satisfiable

Implementation

- 1) for sudoku solving we have fed these clauses to pySAT solver using CNF g that is generated by glucose3() command of pysat.solvers lib:

a) for each sudoku one by one

every cell has at least one

$$\text{value: } C_1 = \bigwedge_{1 \leq r \leq n, 1 \leq c \leq n} (x_{r,c,1} \vee x_{r,c,2} \vee \dots \vee x_{r,c,n})$$

every cell has at most one

$$\text{value: } C_2 = \bigwedge_{1 \leq r \leq n, 1 \leq c \leq n, 1 \leq v < v' \leq n} (\neg x_{r,c,v} \vee \neg x_{r,c,v'})$$

every row has all the

$$\text{numbers: } C_3 = \bigwedge_{1 \leq r \leq n, 1 \leq v \leq n} (x_{r,1,v} \vee x_{r,2,v} \vee \dots \vee x_{r,n,v})$$

every column has all the

$$\text{numbers: } C_4 = \bigwedge_{1 \leq c \leq n, 1 \leq v \leq n} (x_{1,c,v} \vee x_{2,c,v} \vee \dots \vee x_{n,c,v})$$

every block has all the

$$\text{numbers: } C_5 = \bigwedge_{1 \leq r' \leq n, 1 \leq c' \leq n, 1 \leq v \leq n} (\bigvee_{(r,c) \in B_n(r',c')} x_{r,c,v})$$

$$\text{where } B_n(r',c') = \{ (r' * k + i, c' * k + j) \mid 0 \leq i < k, 0 \leq j < k \}$$

$$\text{The solutions must accept Clues } C_6 = \bigwedge_{(r,c,v) \in H} (x_{r,c,v})$$

*these all clauses must be satisfiable for every $y_{r,c,v}$

b) for these two sudoku's to not match each other at each corresponding cell

Each corresponding cell should not match

$$C_6 = \bigwedge_{1 \leq r \leq n, 1 \leq c \leq n, 1 \leq v < v' \leq n} (\neg x_{r,c,v} \vee \neg y_{r,c,v'})$$

2) for sudoku generation part the following steps are used:

- use a completely empty sudoku pair
- solve the module using solver
- start removing values from cells and each time check satisfiability using solver and uniqueness using fast backtracking solution stop which stops when unique solution is not possible.
- return this list of list in form of 'output.csv' when gone through every cell.

Limitation

1. as the number of variables increases with k^6 so for cases with $k \geq 4$ it will take much longer due to limited computation power of systems being used.