

CS202 Assignment 1 Report

1) K Sudoku Pair Solver

Implementation:

Variable: $a(i, j, d)$

$a(i, j, d)$ denotes that i th row and j th column contains digit d . These are numbered from 1 to $2 \cdot k^6$ (Each cell can have k^2 values and there are total k^4 cells in each sudoku)

Various clauses to be considered for Valid Sudoku pair:

- i) At least 1 digit is assigned to each cell
- ii) A cell cannot contain more than 1 digit.
- iii) No 2 cells in a row have the same digit.
- iv) No 2 cells in a column have the same digit.
- v) No 2 cells of a box ($k \cdot k$) can have the same value.
- vi) There is at least 1 digit in each row/column.
- vii) There is at least 1 digit in each block($k \cdot k$).
- viii) Corresponding cells of 2 sudokus cannot have the same value.
- ix) Initial non-zero values in the sudoku pair must hold true.

To implement all these clauses properly we have used a function called **variable()** which takes as input as row, column, digit, sudoku number and gives a single variable corresponding to it in the range $[1, 2 \cdot k^6]$.

Then we add all clauses to the solver. If there exists a sudoku pair, we print one of the solutions and export it to "output_sudoku_pair.csv" or else print "No pair possible" and exit.

2) K Sudoku Pair Generator

Implementation:

- I. Using the implementation of 1st part we generate a random valid Sudoku pair.
- II. Then we make a list of size $2 \cdot k^4$ denoting each cell of the sudoku pair and shuffle it.
- III. We pick a cell from the list, make it 0 and check whether this sudoku has a **unique solution** -

If yes, then proceed to the next cell in the list.

Else keep that cell as before and proceed to the next block in the list.

- IV. Continue step 3 till all the cells in the list are covered.
- V. Check for **uniqueness**:

We know that given Sudoku will always have one solution that is generated at the start.

So, we check if there is any other solution using the clauses generated in part 1 of the assignment. If it has any other solution then it is not unique, otherwise, it is.

We can argue that this process will generate unique sudoku by points III and V. And since we do it for each cell of the sudoku pair, the generated sudoku will be maximal.

#Test Cases for part 1:

There are a total of 6 CSV test files with different sizes of sudoku.

