

CSCI 3020U

Lab #3: Sudoku Solution Validator

Our implementation first opens a file **puzzle.txt** and loads the Sudoku into a 2D array defined within the structure:

```
typedef struct{
    int sudokuGrid[9][9];
    int Rselector;
    int Cselector;
    int Bselector;
} parameters;
```

In our solver starts by iterating through the first column and finds the first empty(zero) slot of the Sudoku. The main idea is that for the current row we are iterating over, we begin with the first 0 element and increment a counter variable i. We then create 3 different threads, 1 for validating the box, 1 for validating the column, and lastly 1 to validate the row.

```
pthread_t rowthread;
pthread_t colthread;
pthread_t boxthread;

data -> Bselector = (col/3) + (row/3)*3;
pthread_create(&boxthread, 0, validate_box, (void *) data);

data -> Cselector = col;
pthread_create(&colthread, 0, validate_col, (void *) data);

data -> Rselector = row;
pthread_create(&rowthread, 0, validate_row, (void *) data);
```

Once we join the validating threads we check that the box, column, and row validators return true, and if they do, we recursively call solve with the next row,column iteration. The function eventually terminates when the row reaches 9 and it returns 1. In the even that we have iterated through all the possibilities then we must set the current row,col to 0 and return false:

```
data -> sudokuGrid[row][col] = 0;
```