CSCE A405 Programming Assignment 4

**Students: Rafael, Ethan J, Hunter J**

# Solution:

We were able to develop a simple sudoku solver by implementing the following constraint satisfactions into our program.

* Check if all cells in 9x9 grid are filled.
* On cell insertion, check for row and column validity.
* On cell insertion, check subset 3x3 grid of neighbors for duplicate value

## Check if all cells in 9x9 grid are filled:

Here we analyze if all the cells in our grid are filled, and if so we have completed the game.



## On inserting a new value into a cell:

Check for row validity. Here we check if the value we wish to insert at a certain location already exists on the same row:



Check for column validity Here we check if the value we wish to insert at a certain location already exists on the same column:



## On inserting a new into a cell (3x3 grid neighbors):

Check all the neighbors around our piece, and see if at one of them is equal to our piece.

## 

## Backtracking:

We found the backtracking algorithm to be really useful as it recursively iterated through each new domain and eliminated those values that did not satisfy our constraints. Our program was able to find a solution for each of the test problems given for this assignment.



## Results:

|  |  |
| --- | --- |
| **Sudoku Problem** | **Time to Solve (Seconds)** |
| 17-3.txt | 0.198598146 |
| 17-9.txt | 1.15465188 |
| 17-15.txt | 1.548193932 |
| 17-10.txt | 1.598210096 |
| 17-19.txt | 2.523869038 |
| 17-14.txt | 6.627482891 |
| 17-4.txt | 7.647003889 |
| hardestinworld.txt | 10.65021205 |
| 17-5.txt | 17.56332684 |
| 17-16.txt | 20.33461499 |
| 17-2.txt | 25.24723411 |
| 17-6.txt | 36.94928193 |
| 17-11.txt | 42.04283404 |
| 17-18.txt | 53.52485204 |
| 17-20.txt | 60.52272487 |
| 17-1.txt | 63.30623388 |
| 17-8.txt | 65.71353388 |
| 17-7.txt | 81.18738699 |
| 17-12.txt | 85.68242693 |
| 17-13.txt | 90.63420391 |
| 17-17.txt | 428.0444469 |
|  |  |