Test report for Sudoku Testing, Assignment Week 5 of Course Software Testing:

By Group D2_1, Cigdem Aytekin 10463135, Jorryt Jan Dijkstra 10462015,
Zarina Efendijeva 10628185, Lukasz Harezlak, 10630171

Our test function (testSudoku) tests 10 randomly generated sudoku's. It performs testing on two different sudoku states: firstly on the initial generated sudoku, which means that it is incomplete and ready to be solved (we call it incomplete sudoku). Secondly on the solved sudoku, which is the finished, complete sudoku (we call this solved sudoku).

Seven different tests are performed in total on each randomly generated sudoku:
1.) Is the incomplete sudoku consistent ? (By using the consistent function from Week5.hs)
2.) Is the solved sudoku consistent? (Again, by using the function above)
3.) We then pick a random grid from the solved sudoku and validate that it is solved by checking that it is free from zero's
4.) We also pick the same random grid, but this time from the incomplete sudoku and validate that it is indeed incomplete, that it has at least one zero in it.
5.) We then look if the incomplete sudoku grid is a sub-sudoku of solved sudoku grid, in other words, the values in the incomplete grid are maintained in the solved sudoku grid, in the same positions, of course, except for zero's. For example, if there was value 5, in the coordinate (3,2) in the incomplete sudoku, then the solved sudoku should also have 5 in the same coordinate. Furthermore, this should also be the case for each non zero value in the incomplete sudoku grid.
6.) We check if the incomplete sudoku is minimal, by using the uniqueSol function from RandomSudoku.hs, which we expect to be true,
7.) We then erase a value from the incomplete sudoku, and call this altered incomplete sudoku. We expect this not to have a unique solution anymore, because the original incomplete sudoku was already minimal. Therefore we test this to **not** have a unique solution.

The program behaves as we expected, we get 'True'from all these tests.