James Davidson - V00812527 | Bridget Rassell - V00804180 | Wilfred Lynch - V00809085
CSC 320 – SAT based Sudoku solving Fall 2015
Our solver can be found at: https://github.com/jamesthomasdavidson/SudokuSolver
☺

Introduction:

We implemented our SAT based Sudoku solver using an encoding and decoding program sudokusolver.py and Minisat, a well-respected SAT solver that takes as input a file specifying a CNF expression in DIMACS format and outputs whether or not the expression is satisfiable, along with the solution if it exists. Our program implements the encoding based on the article "Sudoku as a SAT problem" from connex, then decodes the Minisat output into readable format for the solved Sudoku, which then prints to the console.

Usage:

To run on our test cases, simply run our bash script run_solver.sh (chmod u+x run_solver.sh). All files can be found on the github link above.

Our bash script does the following (for all files of unsolved Sudoku puzzles in the "unencoded" folder):

1) Encodes the puzzle into a CNF expression, then outputs the file in DIMACS format. It does this by running sudokosolver.py with "encode" as the command line argument
2) Pipes the output file into minisat, which provides the solution (if satisfiable) and produces another output file containing the solution
3) Runs sudokosolver.py with "decode" as the command line argument, which decodes Minisat's solution from a base 9 integer to variables i,j and d

Ouput:

Our SAT solver prints all solutions to the console, as well as the cpu execution time.

Testing:

To test our Sudoku solver, we built test files that differ in difficulty then timed how long the the program took to print the solution. We were interested in finding out if problems deemed to be "difficult" or "evil" to humans would reflect on the performance of the SAT-based Sudoku solver. The different levels of difficulty are: easy, medium, hard and evil. Our results are as follows:

	Recorded times:
Easy:	0.014, 0.016
Medium:	0.016, 0.017
Hard:	0.018, 0.015
Evil:	0.018, 0.016

Discussion:

The human difficulty categorization of the Sudoku puzzles did not appear to reflect on the performance of the Sudoku solver. We reasoned that this might be the case because SAT solvers would produce a similar number of clauses for each category, and then the time to solve them would be very close. Humans typically use a trial and error method when solving Sudoku's, and the difficulty reflects on the amount of time it takes to solve it, which is not the case with SAT solvers.

Notes:

- We realize more (many many more..) input files are necessary to draw more concrete conclusions
- We would also need to draw our conclusions from other influences, including number of clauses
- We need to build a more concrete definition of what classifies a Sudoku as "easy" vs "evil"