Assignment 2 – Part 3

Write a class called SudokuToSatReducer<yourName>. This is where the bulk of the work is done.

An instance of this class receives the input file (not just a file name) as a parameter containing a Sudoku board, and outputs the equivalent boolean expression to a file in the DIMACS format, the format in which the SAT solver expects its input. This is accomplished by the following methods.

write a method to

1. Create a file for output.
2. Call createBoard( …) to create a Sudoku board, from the input file which is provided as a parameter.
3. Call reduceBoard() to reduce the board to a SAT input file.
4. Calculate and output the time taken for step c.

You can use the Timer class from your Assignment 1.

You may have to write the following methods. Each of these methods write a clause of numbers to the output file. So, remember to put a 0 at the end of each clause to be consistent with the DIMACS style input file format that your SAT solver expects. Also, you need to output the number of variables and the number of clauses to the output file. These methods translate the constraints on the Sudoku board puzzle. For explanations on the “constraints” see the assignment specification.

1. Write atleastOneInRow (<parameters>) to translate the constraint 1.1.A.
2. Write atmostOneInRow (<parameters>) to translate the constraint 1.1.B.
3. Write atleastOneInCol (<parameters>) to translate the constraint 2.1.A.
4. Write atmostOneInCol (<parameters>) to translate the constraint 2.1.B.
5. Write atleastOneInBox (<parameters>) to translate the constraint 3.1.A.
6. Write atmostOneInBox (<parameters>) to translate the constraint 3.1.B.
7. Similarly, two more methods for constraint 4.

Note that each of the above methods will be called several times (how many times and who calls?). furthermore, since the variables in a clause must be integers, each of the above functions must use an encoding scheme to convert the boolean variables to a single number. This encoding scheme should be written as a separate method.

1. If you need, you may have to write additional helper methods to complete the class.