Project 2

Your task is to develop a Lisp or Java program that can solve Sudoku 9x9 puzzles by mapping this problem to propositional satisfiability (SAT) tests. You are required to use your solution from project 1 as basis to develop the Sudoku 9x9 puzzle solver. Here is a specification of the Sudoku problem for this project: A Sudoku puzzle is represented by a 9×9 board, which comprises nine 3×3 sub-boards (an element of a 3x3 board is also called a square). Some of the entries (squares) of the board are initialized with numbers from 1 to 9, whereas other entries are left blank. A Sudoku puzzle is solved by assigning numbers from 1 to 9 to the blank entries such that every row, every column, and every 3×3 sub-board contains each of the nine possible numbers.

Please observe the following requirements:

- Your program must be able to read in a given Sudoku configuration in a row-like manner and the number 0 (zero) is used to indicate an empty square. Each row is represented as one line of 9 numbers between 0 and 9. All numbers are separated by one space.
- If a solution exists your program must be able to print the solution and also generate a result file in the same format as used for the input file containing a Sudoku configuration.
- The program must be
  - Electronically submitted (including source code, executables, documentation, and test puzzles and their solutions).
  - It must be well documented and demonstrated with at least 20 Sudoku examples of varying difficulty.
  - Your program must be accompanied by a report explaining your chosen approach, especially how you map the Sudoku problem to a SAT problem.
- There will be a competition for the fastest Sudoku 9x9 puzzle solver. This will be judged with a set of Sudoku configurations. The winning group (best average runtime) will get extra marks for their project.
- Your program will be judged with reference to soundness, completeness, termination, and efficiency (speed).

Tips for the design and implementation

- Develop a mapping of the Sudoku rules into a propositional logic CNF sentence.
- Devise a scheme how to represent the state of a Sudoku puzzle and how to disallow a state that violates any of the Sudoku rules.
- Determine the role of propositional variables and how to encode the different values of a square (as element of the puzzle board).
- A solution of a given Sudoku configuration should correspond to a model of the propositional logic CNF sentence you generated.

Deadline

Project 2 is due on **Wednesday, December 1, 2010, midnight**. This deadline will **not** be extended. Only one member of a group (team leader) must submit the project work properly zipped as one archive.