Logical reasoning and programming, lab session II
(October 9, 2017)

II.1 Derive the empty clause from \{\{\overline{a}, b\}, \{\overline{b}, c\}, \{a, \overline{c}\}, \{a, b, c\}, \{\overline{a}, \overline{b}, \overline{c}\}\} using resolution.

II.2 Formalize [Sudoku] as a SAT problem and hence produce a Sudoku solver.

Write a program that generates a problem specification in the [DIMACS format] in such a way that it is possible to specify an input (a partially completed grid) by appending clauses saying which variables are true. You can use [MiniSat] and some input is available from [here].

You can try various cardinality constraints, e.g., the one based on binary encoding that requires $O(n \log n)$ clauses and $O(\log n)$ new variables.

By the way, is it possible to obtain also a generator of Sudoku puzzles this way?  

---

1 Note that this changes the number of clauses, a parameter specified in the DIMACS format.