(1) Write a Sudoku solver using resolution proofs. If you don’t know how Sudoku is played yet, check out 
Your solver should be able to solve problems marked as “easy” there within reasonable time, say, a few 
hours. (Mine, implemented in Java, takes about 20 seconds on easy puzzles, and about 10 minutes to 
a few hours on medium difficulty ones.)

Here are a few pieces of advice:

1. Try to make the Resolution engine generic. That way, you can apply it to other logic puzzles 
easily later. I would recommend defining a class Literal or so, which encapsulates all the 
information about the meaning of variables. Based on that, you can then talk about clauses and 
sets of clauses. It should be pretty easy to separate out the Sudoku specific parts, which is exactly 
what clauses you start with.

2. To make it run reasonably fast, you need to be careful about which data structures you use. 
You want to be able to quickly look up whether a new clause you generated is already among 
the clauses you have. You also presumably want to be able to keep track of which clauses have 
already been combined, so you don’t do the same work again. And you probably want a way to 
combine short clauses first, because they are more likely to generate (useful) short clauses. All 
the while, keep in mind that the number of potential clauses is extremely large. You are explicitly 
allowed to use STL/Java data structures such as Trees, Hashtables, etc., if they are useful.

(2) Describe at least two other natural problems that can be conveniently described in propositional logic 
(and presumably solved using resolution, or other techniques we discuss). List the variables and the 
formulas necessary. If appropriate, distinguish between the “interesting” formulas, and those that one 
might take for granted (such as, in our example from class “Each person is at least one of evil, good, 
or human, but not two”). Notice that the requirement of natural and convenient are to be taken 
seriously: in principle, every problem can be written in propositional logic.